Modeling and Simulating GNSS Signal Structures and Receivers

Jón Ólafur Winkel

Vorsitzender:	Prof. Dr.-Ing. M. Becker
1. Berichterstatter:	Prof. Dr.-Ing. B. Eissfeller
2. Berichterstatter:	Prof. Dr.-Ing. G. Hein
3. Berichterstatter:	em. Prof. Dr.-Ing. Ph. Hartl

Vollständiger Abdruck der an der Fakultät für Bauingenieur- und Vermessungswesen der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines Doktors der Ingenieurswissenschaften (Dr.-Ing.) eingereichten Dissertation.

Contents

List of Tables

List of Figures

1. Introduction

1.1. Motivation–Why Model and Simulate a GNSS?

An operating Global Navigation Satellite System (GNSS) (the navigation system in its environment) is a very complicated system. In order to estimate the influence of a change in one component of the GNSS (e.g. a different carrier frequency or the impact of narrow correlating on the quality of the positioning measured by a user) it is necessary to model the whole system (satellite constellation, signal transmission, receiver, environment etc.), not just parts of it.

In view of future GNSS (e.g. Galileo), their design and development, and maybe more importantly their basic parameters such as carrier frequency, number of frequency channels or orbit parameters, it is very important to estimate their performance, *before* they are built.

When designing a receiver for an existing GNSS (GPS and/or GLONASS), it may also be helpful to "try it out" in a simulation before any hardware is implemented. Thus, an end-to-end simulation of a generic GNSS is of immense importance for existing and future GNSS.

In this work we would like to formulate the relevant theory around GNSS in a general framework, which encompasses all (or at least most) of the ideas and concepts currently circulating in the literature. Only in such a framework can the different ideas be compared and tested in a simulation. In developing the theory the common features and the differences of the various concepts become evident and can be analyzed in detail. The philosophy is to go as far as possible with the theory and from there algorithms are designed and implemented. Staying with the theory for as long as possible has two main advantages: the more important one is that a deeper understanding of the system is achieved. The second is that the numerics are usually reduced. But at some point we cannot carry the theory further and there the simulations step in.

When deciding what to include in the simulation, one needs to keep the goal in mind that one wants to achieve by the simulation. Implementing a simulation takes a lot of effort and thus phenomena should only be simulated that need to be simulated. Numeric simulation should also not be overestimated and should be regarded as a last resort when analysis will not carry us further. Proper analysis can usually be relied on to yield a more profound understanding of the physical phenomena than numeric simulations.

Another philosophy that is often encountered is to simulate everything that can possible be simulated. While this may often be the sensible thing to do, it at least as often does not make sense. For example using high precision ephemeris with an absolute orbit accuracy of 1 m to plan a measurement campaign where only the approximate azimuth and elevation angles are required is killing mosquitoes with a shotgun; it is too expensive, it takes too long time and demands too much computational power. Apart from that it contains too many potential sources of error.

When the simulator described in this thesis was designed, an effort was made to maintain a certain balance between the models and algorithms, concerning the level of detail and accuracy. Models and input data, which depend on each other should apply a similar level of accuracy; it does not make sense to use a terrain model which has a resolution in the order of centimeters, when the assumptions for the propagation model of the signal only applies to a curvature larger than tens of centimeters (see section 2.3.4.1 on rough surfaces).

The models and algorithms must also be compatible to each other. For example if the oscillator error in the receiver is modeled such that the error on each channel is *identical*, it will cancel *exactly* when the position is calculated.

Basically, there are two main reasons for implementing a simulation; understanding the system and/or comparing the performance of more than one system. For the understanding of the system, analysis is the vehicle of choice. However, it has its limitations, in particular when many components interact.

In this work emphasis is laid on the actual performance of the system, provided it is technically available. Questions of integrity, continuity of service, availability and system reliability are not considered in this work unless they concern the performance in some way (as is the case for multipath environments).

1.2. Structure of the Thesis

The thesis is structured as follows. In chapter two the models are introduced and/or derived. First a model for the constellation is introduced. As the emphasis in this thesis is not on the orbit propagation, but more on the receiver and signal side, the constellation is simply a Kepler model. In that section there is, however, some analysis regarding the dynamics of the line-of-sight (Doppler effect and derivatives of higer order).

In this thesis there are basically two models for the signal: a signal propagation model, which is a physical model that approximates the electromagnetic properties of the signal and its interaction with the environment. The second model describes the signal during signal processing. This model reflects the situation after reception through the antenna and conversion to a simple voltage. The next two sections deal with the signal model and the signal propagation model. The last section in chapter two concerns the receiver model. The signal processing model and the receiver model form the core of the thesis.

Chapter 3 contains a description of the implementation of the simulator called Simulated Navigation Satellite System (SNSS), which was developed and implemented as a part of this thesis.

In chapter 4 results from numerous simulations are presented. The simulations aim to analyse the performance of the various signal structures and compare their performance under various conditions. In particular multipath and non-optimal signal-to-noise situations are of interest as well as the impact of signal dynamics.

Chapter 5 contains analysis and theoretical results, which are not directly related to the analysis and implementation of the simulator. These are interference analysis and calculations related to the compensation of the ionosphere and its impact on the noise performance of the code and phase measurements. There is also some analysis on the spatial correlation of multipath errors.

The last chapter contains a summary of the thesis and conclusions drawn from the results.

2. Models

In this chapter we introduce and develop the theoretical models, which were implemented in the simulation software. Analysis not directly relevant to the implementation of the simulator is presented in chapter 5

In the first section the model for the orbits is introduced. The orbits are simply calculated using the same procedure as in GPS, i.e. the GPS data format and the corresponding algorithms are used to propagate the satellite orbits. Stationary satellites are also used to represent geostationary orbits and pseudolites. The second section contains a definition of the signal model and then some base-band properties, like spectrum, power spectrum and autocorrelation functions are derived.

In the third section the signal propagation model is developed. It consists of a geometric and electromagnetic part[1]. The receiver model is developed in the fourth section.

2.1. Constellation

For the purposes of a simulation like this it is considered sufficient to describe the orbits in a relatively crude way. Basically, there are two kinds of satellites: moving and stationary (relative to WGS-84). The stationary ones are simply described with a Cartesian coordinate and a zero velocity and acceleration[2] (in WGS-84). The moving orbits are described in the same way that GPS satellites are, using almanac data.

Although this section is called orbit modeling, the relative motion between the receiver and the signal source (satellite or pseudolite) will also be considered, i.e. the motion of the line-of-sight. From the receiver point of view this is the relevant geometry, as it relates to the Doppler shift and corresponding effects of higher order. The case for stationary receivers and satellites on GPS-like orbits was treated in-depth in [Eis97]. For the receiver model the dynamics of the line-of-sight are important and are required as an input. The line-of-sight dictates the transient behavior of the tracing loops.

[1] Theoretically, there is no need to make this distinction and theoretically it doesn't make sense. Here this distinction is made for algorithmic and implementation reasons.

[2] Coriolis forces etc. are not relevant in this context.

2.1.1. Orbit Modeling

2.1.1.1. Kepler Orbits

The orbits of the moving satellites are described by Kepler orbits, which are obtained from the almanac data. The almanac data is identical to that used for the GPS orbit calculation (without clock corrections).

For a Kepler orbit the motion is restricted to the plane (see e.g. [Gol80]). The position vector and velocity vector of a satellite on a Eulerian orbit in an inertial co-ordinate system where the orbit is assumed to lie in the xy-plane can be described by:

$$\vec{r} = a \begin{pmatrix} \cos(E) - e \\ \sqrt{1-e^2}\sin(E) \end{pmatrix} \quad \dot{\vec{r}} = \frac{na^2}{|\vec{r}|} \begin{pmatrix} -\sin(E) \\ \sqrt{1-e^2}\cos(E) \end{pmatrix} \tag{2.1}$$

where $E = E(t)$ is the eccentric anomaly, e is the eccentricity of the ellipse, a is the semi major axis and $n := \sqrt{\mu/a^3}$ is the mean angular velocity or mean motion. When the satellites are orbiting the earth, the value of the gravitational constant is μ = 3.986004415· 10^{14} m^3/s^2. The length of the vector $\vec{r}$ is $|\vec{r}| = a(1 - e\cos(E))$. The relations between the symbols are explained in figure (2.1)

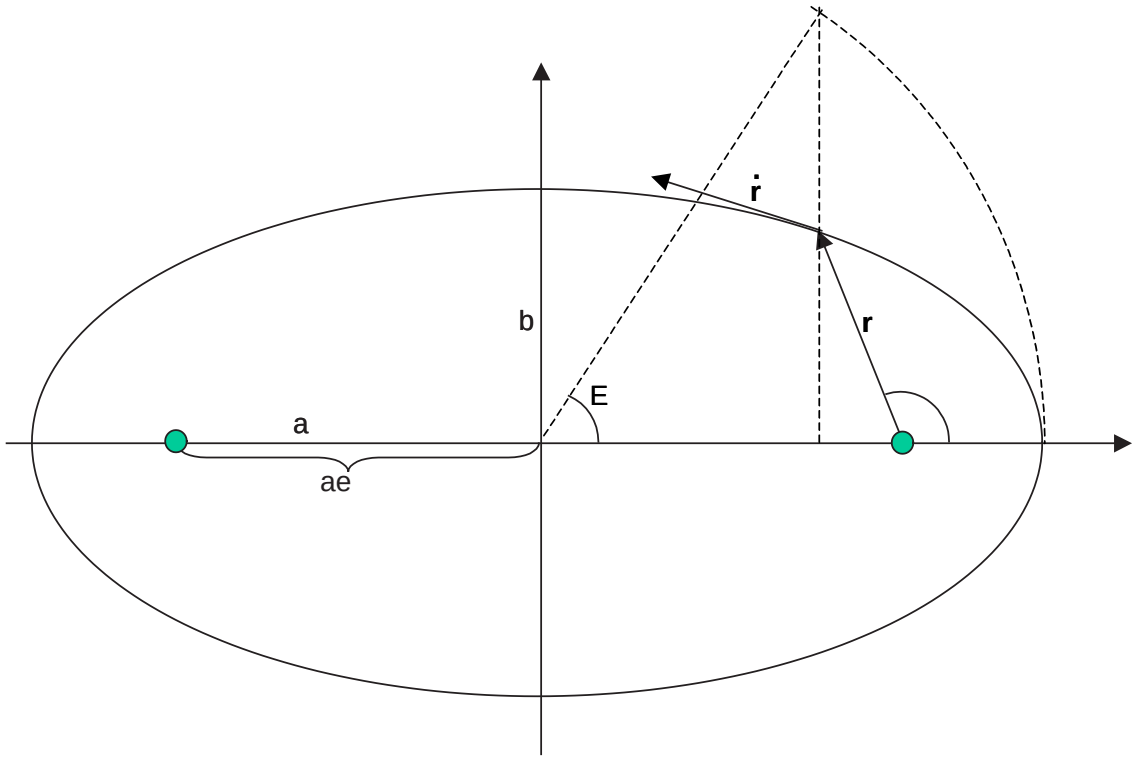

Figure 2.1.: Explanation of the parameters involved in the description of the Kepler orbit

For the purposes of this work it is necessary to obtain the velocity and acceleration of the line-of-sight. The acceleration vector is easily obtained, by differentiating eq. (2.1) with respect to time:

$$\ddot{\vec{r}} = \frac{-n^2a^3}{|\vec{r}|^2} \begin{pmatrix} \cos(E) - \frac{ea}{|\vec{r}|}\sin^2(E) \\ \sqrt{1-e^2}\sin(E)[1+\frac{ea}{|\vec{r}|}] \end{pmatrix} \tag{2.2}$$

here it was used that

$$E(t) = n(t-T_0) + e\sin(E(t)) \qquad \text{and thus} \qquad \dot{E} = \frac{n}{1-e\cos(E)} = \frac{na}{|\vec{r}|} \tag{2.3}$$

The quantities above are all in an inertial reference system. They now have to be transformed into WGS-84. Denoting the inclination of the orbit by θ_i and by Ω the corrected longitude of ascending node (Earth's rotation, to a first approximation), the results from above can be rotated into WGS-84 with the following transformation:

$$\begin{aligned} \vec{R} &:= \mathbb{R}\vec{r} \\ \dot{\vec{R}} &= \frac{d}{dt}(\mathbb{R}\vec{r}) = \dot{\mathbb{R}}\vec{r} + \mathbb{R}\dot{\vec{r}} \\ \ddot{\vec{R}} &= \frac{d}{dt}\left(\dot{\mathbb{R}}\vec{r} + \mathbb{R}\dot{\vec{r}}\right) = \ddot{\mathbb{R}}\vec{r} + 2\dot{\mathbb{R}}\dot{\vec{r}} + \mathbb{R}\ddot{\vec{r}} \end{aligned} \tag{2.4}$$

where $\mathbb{R}$ is defined as

$$\mathbb{R} := \begin{pmatrix} \cos(\Omega) & -\cos(\theta_i)\sin(\Omega) & 0 \\ \sin(\Omega) & \cos(\theta_i)\sin(\Omega) & 0 \\ 0 & \sin(\theta_i) & 0 \end{pmatrix} \tag{2.5}$$

The angle θ_i is a constant, obtained from the almanac data and Ω is given by

$$\Omega = \Omega(t) := \Omega_0 + (\Delta\dot{\Omega} - \dot{\Omega}_{\text{earth}})t \tag{2.6}$$

The parameters Ω_0, $\Delta\dot{\Omega}$ and $\dot{\Omega}_{\text{earth}}$ are constant values. Ω_0 is the angle of the ascending node in radians, $\Delta\dot{\Omega}$ is a correction value to account for the drift of the ascending node and $\dot{\Omega}_{\text{earth}}$ is the rotation rate of the earth.

The velocity of the line-of-sight is defined as $d_t|\vec{r}|$ and can be shown to be

$$d_t|\vec{r}| = \frac{ena^2}{|\vec{r}|}\sin(E) \tag{2.7}$$

2.1.2. Geostationary Orbits

In SNSS the geostationary orbits are simply defined by a time-independent co-ordinate in WGS-84. Such orbits are useful for specifying

- Geo-stationary orbits

- Pseudolites

Note, that as the geostationary orbit in SNSS can have any radius, it is not always a solution to the Kepler equations.

The velocity and acceleration can safely be set to $\vec{0}$, because the user is moving in WGS-84 and the speed and acceleration of the line-of-sight only arises from the motion of the user.

2.1.3. Motion of the Line-of-Sight

In this section we take a closer look at the behavior of the line-of-sight. The motion of the line-of-sight for stationary receivers and GPS-like orbits can be shown to be well-behaved. There is some steady Doppler (corresponding to the first derivative of the line-of-sight) in the order of 1 kHz, which changes from a positive to a negative value during the satellite pass. The duration of the pass is in the order of 10^4-10^5 seconds. Thus, higher order derivatives of the line-of-sight are very small.

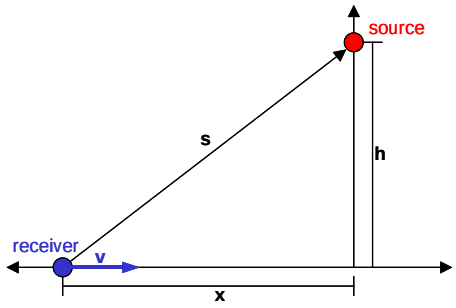

Figure 2.2.: A simple scenario to demonstrate the behavior of the dynamics of the line-of-sight for a receiver passing a pseudolite. The receiver moves with constant velocity

For the case of a terrestrial transmitter (i.e. pseudolite) the situation turns out to be less trivial and it turns out that even for a receiver moving at constant velocity and a stationary transmitter the dynamics of the line-of-sight can, in fact, become quite violent. To see this, let us assume the user to have the position $\vec{u}$ and the satellite position vector in WGS-84 to be $\vec{R}$, then the line-of-sight is the vector defined by $\vec{s} := \vec{R} - \vec{u}$. The relative velocity vector is defined by $\vec{v} := \vec{V}_R - \vec{V}_u$, where the vectors $\vec{V}_R$ and $\vec{V}_u$ are the velocity vectors of the satellite and user receiver respectively.

Using the identities $\frac{d}{dt}s^2 = 2s\dot{s}$ and $s^2 = \vec{v} \cdot \vec{v}$, we can calculate the first two derivatives of s:

$$\begin{aligned} \dot{s} &= \frac{\vec{s} \cdot \vec{v}}{s} \\ \ddot{s} &= \frac{1}{s}\left(\vec{v}^2 + \vec{a} \cdot \vec{s} - \dot{s}^2\right) \end{aligned} \tag{2.8}$$

where $s := |\vec{s}|$ and $\vec{a} := \dot{\vec{v}}$. The position, velocity and acceleration of the user are unknown; they are needed as input from the simulation. However, the satellite kinematics are known and can be inserted into eq. (2.8).

If the signal is coming from a satellite, higher derivatives of the line-of-sight will be small. But if the signal comes from a pseudolite, the situation can be entirely different. A typical

situation would be an aircraft flying over a pseudolite in final approach. The so-called „bubble concept" [CPC+93, CPL+93] foresees a pair of pseudolites at the end of the run-way. As the aircraft lands, it passes over these two pseudolites just before landing. Obviously, the plane will be traveling at velocities in the order of 100 m/s and the fly-by distance (i.e. the minimum distance that the receiver will have to the pseudolite) will be in the order of 100 m.

To see how this looks for a simplified configuration of a pseudolite and a receiver moving along a straight line with constant velocity, we can write down higher derivatives of s. Under these assumptions, we then get:

$$\dot{s} = \frac{\vec{s} \cdot \vec{v}}{s} \qquad \ddot{s} = \frac{\vec{v}^2 - \dot{s}^2}{s} \tag{2.9}$$

$$\dddot{s} = -3\frac{\dot{s}\ddot{s}}{s} \qquad \ddddot{s} = \dddot{s}\left(\frac{\ddot{s}}{\dot{s}} - 4\frac{\dot{s}}{s}\right) \tag{2.10}$$

Higher derivatives can also be calculated in a similar manner. The expressions in eq. (2.9) and eq. (2.10) are iterative, i.e. higher derivatives are expressed in terms of lower ones to keep the equations simple. Thus, if we were to write down eq. (2.9) and eq. (2.10) in terms of h and v the equations would become increasingly complex.

To demonstrate eq. (2.9) and eq. (2.10) on a simple example, consider the configuration in figure (2.2). The fly-by distance is h, the velocity of the receiver is v. The coordinate system is chosen such that at time t = 0 s, the receiver passes the pseudolite. The units chosen are SI, but the plots in figure (2.3) scale with v/h, so that we get the same plots for h = 200 m and v = 100 m/s as we would for say h = 20 m and v = 10 m/s. In this case the vertical scale would of course also be scaled down by a factor of 10.

The first five derivatives of the length of the line-of-sight are shown in figure (2.3). The fly-by distances were $h = 100$, 150, 200, 300, 400 and 500 m. The speed of the receiver was v= 100 m/s.

In the first two plots the amplitude of the higher derivatives are generally larger than for the first derivatives. So the higher the derivative, the larger the amplitude. As the fly-by distance is increased, the higher derivatives are reduced and as can be seen in the last plot (h = 500 m) the amplitude of the fifth derivative is the smallest one.

Apparently, there seems to be a „dynamic barrier" around the pseudolite, because if the fly-by distance is smaller than the velocity, the higher derivatives seem to diverge. If the derivatives are evaluated at t = 0 we see that the n-th derivative is[3]

$$\left.\frac{d^n}{dt^n}s(t)\right|_{t=0} = \begin{cases} \frac{v^n}{h^{n-1}}(-1)^{(n/2)-1}(n-1)\prod_{k=1}^{n/2-1}(2k-1)^2 & n \text{ even} \\ 0 & n \text{ odd} \end{cases} \tag{2.11}$$

[3]This statement is not proven, but it holds for up to n = 500, for which the product in eq. (2.11) is in the order of 10^{1000}. For all practical purposes the statement is considered valid.

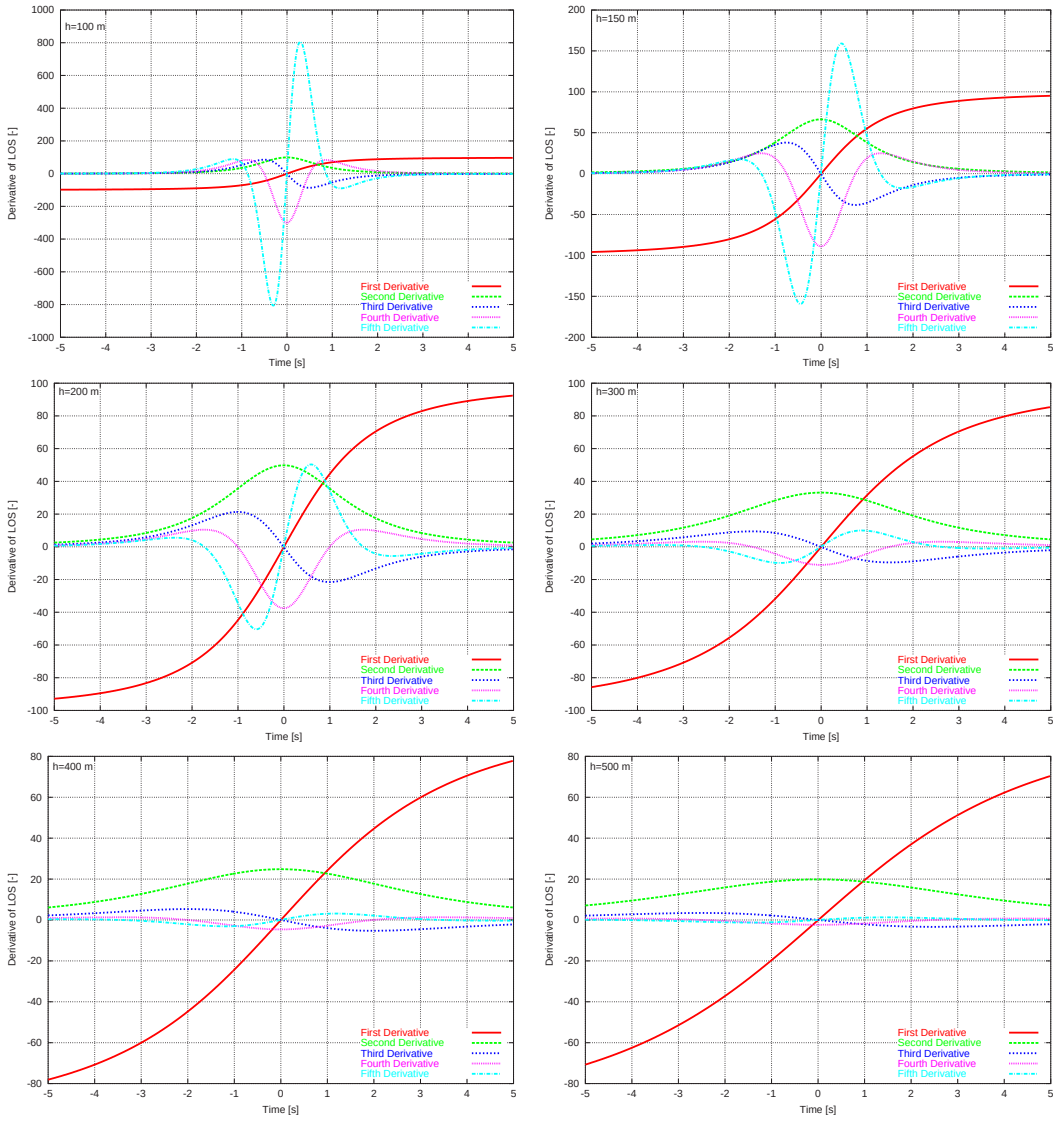

Figure 2.3.: The first five derivatives of the line-of-sight for the simple fly-by scenario described in figure (2.2). The velocity of the receiver was v = 100 m/s for all plots and the fly-by distance (h) was 150, 200, 300 and 400 m. The units on the vertical axis are SI, i.e. m/s^2, m/s^3 and so on. Note that the vertical scale differs between plots.

It is therefore important that the factor $v/h < 1$ Hz, else the derivatives in eq. (2.11) will diverge.

As can be seen from figure (2.3) the derivatives for odd n have their maxima to the left and right of t = 0 and their location depends on n. Therefore it is a bit harder to obtain an expression analogous to eq. (2.11) for odd n.

Usually, when signal dynamics are discussed in the context of pseudolites, people are concerned about the so-called near-far effect which relates to the variation in signal power with the line-of-sight. This has consequences for the dynamic range of the low-noise amplifiers, the automatic gain-control and the A/D converter in the front-end, which tend to become saturated or at least start operating in a non-linear way for too high signal-to-noise ratios.

The discussion here is more concerned with the higher derivatives of the line-of-sight and the impact on the tracking loops of the receiver. A tracking loop of order n will have a dynamic tracking error (also called transient tracking error) proportional to the n-th derivative of s. Tracking loops in GPS receivers are typically of order two or three and looking at the plots in figure (2.3) we see that the tracking error may thus become very large and may eventually cause the receiver to loose lock. This phenomenon is independent from the signal power in the sense that even if the dynamic range of receiver can handle the variations in the power and thus power control can successfully be applied, it may still loose lock because of the transient tracking error. However, the tracking loop has a certain temporal inertia, caused by the loop-bandwidth. If the fly-by is very fast, the tracking-loop may be able to sustain the sudden changes in the line-of-sight if the time frame for which the receiver is in the critical region, where the Doppler changes sign, is short compared to the inverse loop bandwidth. In section 4.2.2.1 this issue will be addressed and results of simulations particularly designed to investigate this effect will be presented.

2.2. Signal Model

2.2.1. Introduction

In this thesis there are two models for the signal. A physical one for the signal propagation from the satellite, through the atmosphere, environment and through the antenna into the signal processing part of the receiver. From there the signal is modeled using a signal processing model appropriate for modeling the signal within the receiver (analog and digital).

The approach followed here is very similar to the one taken in [Eis97]. However, here the model in [Eis97] is re-formulated to include any type of signal and some refinements are made.

The signal is assumed to be modulated onto a carrier wave in a binary phase shift keyed (BPSK) manner. Thus, the modulation would be equivalent to a multiplication of ± 1 with the carrier wave every time the sign of the chip changes. This can be formulated as a multiplication of the carrier with a function that is piece-wise constant and takes on the values ± 1. It has transitions each time the code changes sign. A plot of such a function looks like a random sequence of rectangles with a width corresponding to the chip duration. This is illustrated in figure (2.4). The left part is a schematic representation of the time domain and to the right the situation in frequency domain is shown. This corresponds to what can be called un-shaped chips and approximates the signal structure of the GPS C/A code adequately. In the model introduced here the shape of these rectangles is not constrained to rectangles, but can have any shape. The shaping of the chip can have many reasons, discussed below. The more general chip shape is shown in figure (2.5).

The data bit is assumed to be much longer than the code bit. For the pre-detection model this translates into the assumption that the code has infinite length. For the modeling of the pre-detection process (i.e. correlation process) the infinite code length assumption is quite sufficient, as a code length in the order of 1.000 or even 10.000 has very little impact on the shape of the resulting correlation functions, which are the output of the correlation process. Other effects like white noise and band limiting effects are much more important. It should be noted that, although the code length is ignored in the pre-detection model, it will be introduced in the receiver model where the finite pre-detection bandwidth plays an important role.

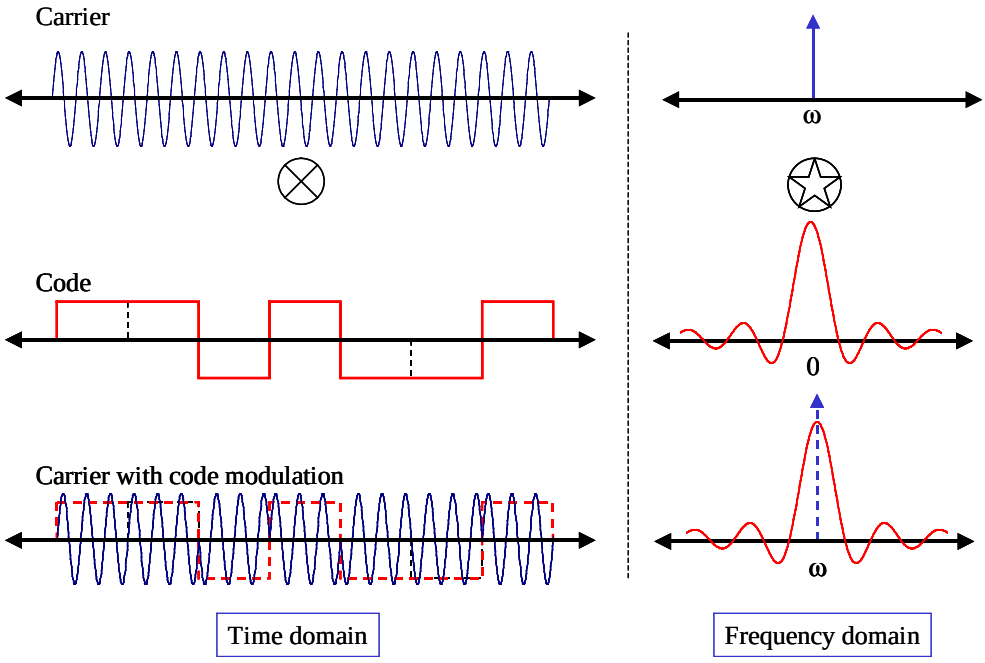

Figure 2.4.: Schematic representation of normal BPSK modulation. The left part shows the time domain and the right part the corresponding situation in the frequency domain

2.2.2. Mathematical Definition

For the signal processing the GNSS signal received at the receiver is modeled by the following equation

$$\begin{aligned} S(t;\tau(t),\phi(t)) &:= \sum_{k=1}^{N_c} b_k g_{T_c}(t - kT_c + \tau(t)) \sin(\omega t + \phi(t)) \\ &=: G^c_{T_c}(t + \tau(t)) \sin(\omega t + \phi(t)) \end{aligned} \tag{2.12}$$

The number of chips in the code is N_c and the bit-value is denoted by b_k. As already mentioned the data bit will only enter indirectly in the receiver model, as an upper limit on the integration time (pre-detection bandwidth). The carrier frequency is denoted by ω and T_c is the chip length (or duration, depending on the units used). The last equation just defines a convenient short-hand for the code. Note that in eq. (2.12) the code delay, $\tau(t)$ and phase shift, $\phi(t)$ are functions of time, t and thus Doppler-shifts, jerks, etc. are implied. In the simulations this general time-dependency is rigorously taken into account.

The function $g_{T_c}(t)$ contains information on the pulse-shape. By formulating the model using an explicit expression for the pulse-shape the flexibility is increased considerably. Not only is it possible to simulate the GPS and GLONASS signals with and without band-limitation, but any conceivable pulse-shape can be modeled and simulated like for example intentional pulse-shaping like the raised cosine pulse-shape.

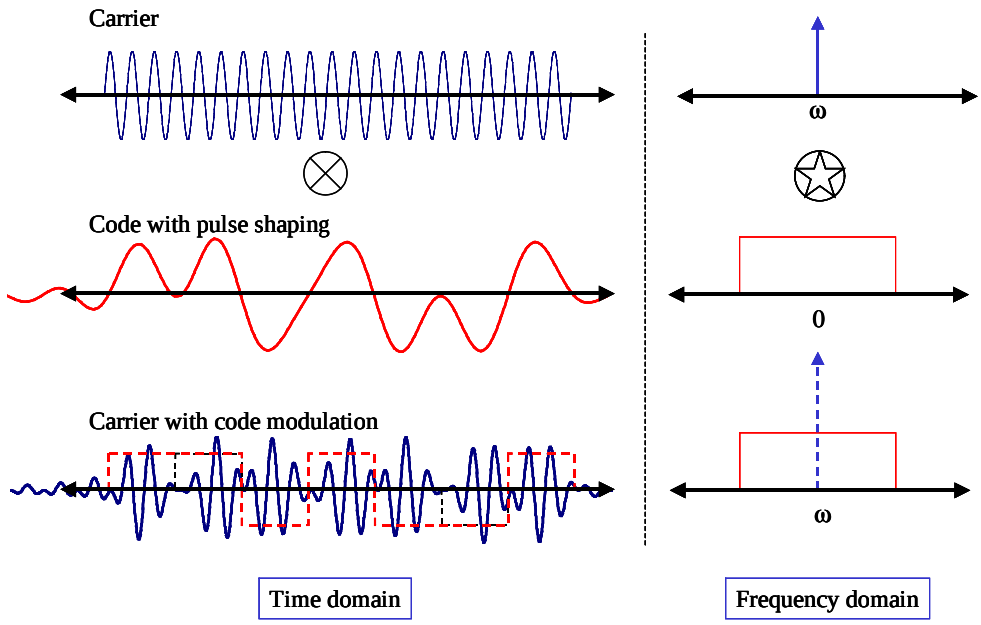

Figure 2.5.: Schematic representation of chip-shaped BPSK modulation. The left part shows the time domain and the right part the corresponding situation in the frequency domain

When considering coded bits, e.g. Manchester coding or binary offset carrier (BOC) coding (see section 2.2.8), it may be convenient to consider the coding of the bits as pulse shaping. Then the function $g_{T_c}(t)$ in eq. (2.12) contains the pulse-shaping as well as the bit-encoding.

The properties of the signal structure in this model are defined by the properties of the codes and the form of $g_{T_c}(t)$. The signal structures that are considered in this work are:

- Infinite bandwidth, rectangular chip-shape
- Band-limited rectangular chip shapes
- Infinite bandwidth rectangular chip-shape with binary offset carrier modulation (BOC) (see section 2.2.8)
- Band-limited rectangular chip-shape with binary offset carrier modulation (BOC) (see section 2.2.8)
- Infinite bandwidth rectangular chip-shape with sinusoidal offset carrier modulation (SOC) (see section 2.2.7)
- Raised cosine chip shapes (see section 2.2.6)

The signals above can be associated with any codes.

In this thesis the code will be assumed to be an infinite length random code. In practice that can of course not be realized but using long Gold codes or codes derived from them will in general approximate the infinite code reasonably well. In the case of the GPS C/A codes where the code length is only 1.023, this approximation is for most purposes very good. For other systems it is safe to assume that much longer codes[4] will be used and thus the approximation becomes even better than for the GPS C/A code case.

2.2.3. Base-Band Signal Structures; Infinite Code Length

In this section some properties of the signal structures mentioned above will be derived. A theory for the signal structures is developed under the assumption that the codes are infinite random sequences. Under this assumption it can be shown that the resulting spectrum is equivalent to the spectrum of the chip shape[LS73].

First we define the chip shapes (i.e. $g_{T_c}(t)$) and the encoding and then we proceed to establish the corresponding spectral densities. Finally, we write down the auto-correlation function or cross-correlation functions for the signals. When calculating the correlation functions, a reference signal of the form

$$S_{\text{ref}} = 2G^{\text{ref}}_{T_c}(t)\, \mathrm{e}^{i(\omega' t + \phi'(t))} = G^{\text{ref}}_{T_c}(t)\, [\cos(\omega' t + \phi'(t)) + i \sin(\omega' t + \phi'(t))] \qquad (2.13)$$

is used, where the reference frequency and phase are defined as $\omega' := \omega - \Delta\omega$ and $\phi' := \phi - \Delta\phi$, respectively. The parameters $\Delta\omega$ and $\Delta\phi$ represent the frequency and phase offset, respectively.

In this formulation the so-called down-mixing in the receiver is implied. The radio signal is modulated onto a high frequency, typically about 1.5 GHz. This frequency is too high to be treated directly in the digital part of the receiver. Thus, the signal is multiplied by a frequency close to the carrier frequency (ω' in the present analysis) in the analog section, before it is converted to a digital signal. The result is a signal having frequency components around $\Delta\omega$ and around $2\omega + \Delta\omega$. The high frequencies are filtered by a low-pass filter. The resulting low-frequency signal is called base-band signal and the process of transforming the radio signal down to base-band is called signal conditioning.

It is mathematically convenient to write the reference carrier signal in complex form, because the entire correlation process is linear[5] and thus the I and Q channels can be obtained by simply taking the real or imaginary part of the result. In the next section on the receiver modeling we shall see the reason for the definition of these two channels.

The correlation process as it is presented in this thesis is simply an integration of the product of the reference signal eq. (2.13) and the received signal eq. (2.12). The integration interval, T_p defines the pre-detection bandwidth.

[4] GPS L5, code length: 10,230, GPS L2, code length: c.a. 767,250

[5] Multiplying a real function with a complex one and integrating the result and then taking the real/imaginary part is equivalent to treating the real and imaginary part separately

2.2.4. General Comments on the Pre-Detection Integral

Before we proceed, we would like to establish a general expression for the correlation function of two codes modulated with arbitrary chip shapes.

Consider the two functions

$$s(t) := \sum_j c_j g_s(t + jT_c) f(t) \qquad \text{and} \qquad r(t) := \sum_k c_k g_r(t + kT_c) h(t) \tag{2.14}$$

where $\{c_j\}_{j=1}^{N_c}$ is a common bit sequence defining the code. g_s and g_r are the chip shapes of the signals. The functions f and h are arbitrary functions that vary slowly compared to the chip duration T_c.

Assume we want to perform a correlation between these two functions, as usual we then have (inserting eq. (2.14))

$$\begin{aligned} I(\tau) &= \int_{t-T_p}^{t} s(t')r(t'-\tau)dt' \\ &\approx \int_{t-T_p}^{t} \sum_j c_j^2 g_s(t' + jT_c) g_r(t' + jT_c - \tau) f(t') h(t'-\tau) dt' \end{aligned} \tag{2.15}$$

where T_p is the integration time and we assumed that the sum of the off-diagonal terms in the second equation are negligible. That assumption is only valid because it is assumed that the chip shapes (g_s and g_r) are substantial *only* in the region $[-T_c/2..T_c/2]$. Outside this region g_s and g_r fall off sufficiently fast to justify the assumption. For infinite bandwidth rectangular chips this is strictly true and the approximation is actually and equation. For the raised cosine the chips fall off faster than $1/t$.

If the functions f and h in eq. (2.15) vary slowly compared to the entire integration interval, T_p in eq. (2.15) they can be pulled in front of the integral and the integral is easily evaluated to

$$\boxed{I(t,\tau) \approx f(t)h(t-\tau) \int_{t-T_p}^{t} g_s(t')g_r(t'-\tau)dt' = f(t)h(t-\tau)R_{sr}(\tau)} \tag{2.16}$$

where $R_{sr}(\tau)$ is the cross-correlation function between $g_s(\cdot)$ and $g_r(\cdot)$. It was also used that the square of the code bit is 1. This is the assumption and approximation that is usually applied when calculating the auto-correlation functions for signals [Eis97, Die96].

If the functions f and h (or more accurately the factor $f(t')h(t'-\tau)$) do *not* vary slowly compared to the integration time T_p, but *do* vary slowly compared T_c all is not lost. The fact that the support of the chip shape is very small (i.e. in the order of the chip duration)

can be used in a more efficient manner. Starting with eq. (2.15) we can write

$$\begin{aligned} I(\tau) &\approx \sum_j \int_{t-T_p}^{t} g_s(t'+jT_c)g_r(t'+jT_c-\tau)f(t')h(t'-\tau)dt' \\ &= \sum_j \int_{t+jT_c-T_p}^{t+jT_c} g_s(u)g_r(u-\tau)f(u+jT_c)h(u+jT_c-\tau)du \\ &\approx \sum_j \int_{-3T_c/2}^{3T_c/2} g_s(u)g_r(u-\tau)f(u+jT_c)h(u+jT_c-\tau)du \end{aligned} \tag{2.17}$$

where a change of variable $u \leftarrow t - iT_c$ was made. In the last equation the assumption on the support of the chip shape was used to *shrink* the integration interval. The integrand is substantial only close to zero in this new variable u. As before the approximation is exact in the case of rectangular chips and for infinite random codes it also becomes exact.

Now, the integration interval in the last equation in eq. (2.17) is only 3 chip lengths wide instead of over the entire code. Thus we must only assume the factor $f(u+jT_c)h(u+jT_c-\tau)$ to be constant for a duration of $\approx 3T_c$ for each of the arguments $u+jT_c$. Using the weighted mean value theorem for integrals eq. (B.8) we can pull the functions $f(u+jT_c)h(u+jT_c-\tau)$ in front of the integral, where the functions are evaluated in the middle of the interval (i.e. for $u=0$). Thus eq. (2.17) becomes

$$\boxed{\begin{aligned} I(\tau) &= \sum_j \overline{f(jT_c)h(jT_c-\tau)} \int_{-3T_c/2}^{3T_c/2} g_s(u)g_r(u-\tau)du \\ &\approx \frac{1}{T_c}\sum_j f(jT_c)h(jT_c-\tau)T_c \int_{-\infty}^{\infty} g_s(u)g_r(u-\tau)du \\ &\approx R_{sr}(\tau)\int_{t-T_c}^{t} f(t')h(t'-\tau)dt' \end{aligned}} \tag{2.18}$$

In the first equation the bar over f and h symbolizes a mean value of the functions in the corresponding integration interval. Then the sum is approximated with an integral, where the integration interval corresponds to the integration interval of the code or more precisely the pre-detection bandwidth.

We see that the integral eq. (2.15) can be written as a factor of two integrals, one representing the correlation function of the two chip shapes and the other is simply a correlation between f and h. This result will be used extensively in this chapter.

2.2.5. Rectangular Chip Shape

The simplest chip-shape considered here is the rectangular pulse-shape, defined by

$$g_{rect}(t) := \mathrm{Rect}(t/T_c)\,. \tag{2.19}$$

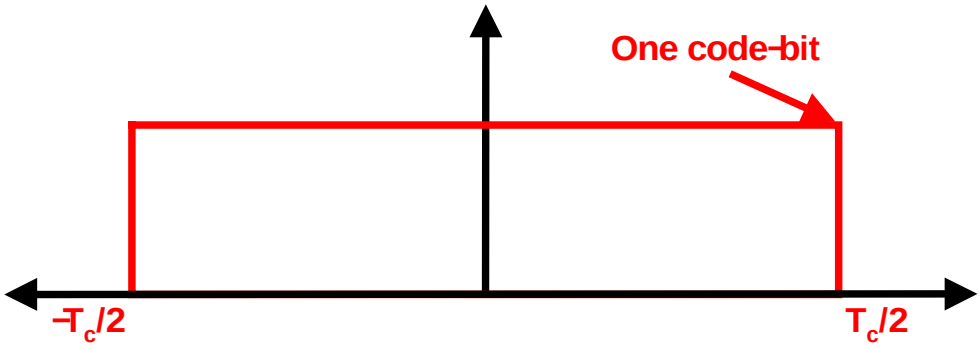

Figure 2.6.: Chip shape of a rectangular bit

Its shape is shown in figure (2.6) and the definition of $\mathrm{Rect}(\cdot)$ can be found in appendix A.

The spectrum for the rectangular chip is the well-known sinc-function (see any textbook on signals, e.g. [Spi73, LS73])

$$S_{\text{rect}}(\omega) = \int \mathrm{e}^{-i\omega t}\, g_{\text{rect}}(t)dt = T_c\,\mathrm{sinc}(\omega T_c/2)\,. \tag{2.20}$$

In base-band the autocorrelation function of the rectangular bit is simply the triangular function (see eq. (A.4) in appendix A):

$$R_{\text{Rect}}(\tau) = \mathrm{Tri}(\tau/T_c) \tag{2.21}$$

$$S_{\text{tri}}(\omega) = \int \mathrm{e}^{-i\omega t}\,\mathrm{Tri}(t/T_c)dt = T_c^2\,\mathrm{sinc}^2(\omega T_c/2) \tag{2.22}$$

The spectra of both the $\mathrm{Rect}(t)$ and $\mathrm{Tri}(t)$ are shown in figure (2.7).

If the frequency of the correlating function is a bit off, i.e. if the oscillator in the receiver is not exactly tuned onto the carrier of the received signal, we can describe the correlation process in the following way:

$$\begin{aligned} R^*(\tau,\Delta\omega) &= \frac{1}{T_p}\int_{-T_p/2}^{T_p/2} g_{\text{rect}}(t-\tau)g_{\text{rect}}(t)\,\mathrm{e}^{i\omega' t+i\phi'}\sin(\omega t+\phi)dt \\ &\approx -\frac{1}{2iT_p}R_{\text{rect}}(\tau)\int_{-T_p/2}^{T_p/2}\mathrm{e}^{i\Delta\omega t+i\Delta\phi}\,dt \\ &\approx R_{\text{rect}}(\tau)\frac{i\,\mathrm{e}^{i\Delta\phi}}{2}\,\mathrm{sinc}\,[\Delta\omega T_p/2] \end{aligned} \tag{2.23}$$

where eq. (2.18) was used and the high frequency terms were dropped in the second equation. $\Delta\phi$ and $\Delta\omega$ are the phase and frequency error respectively. In eq. (2.23) we

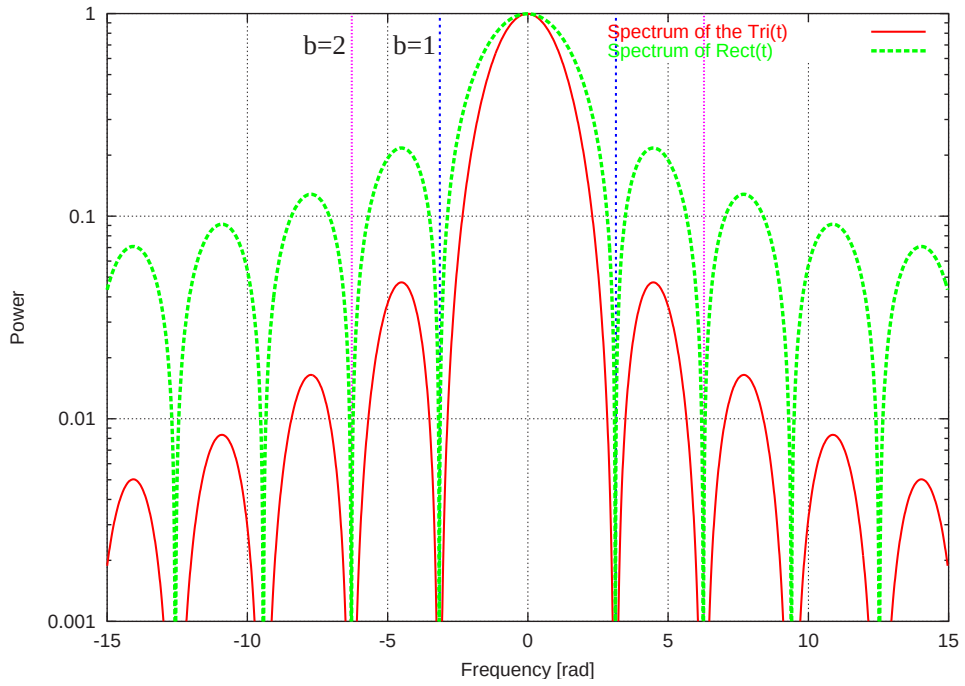

Figure 2.7.: Spectrum of a rectangular bit ($\mathrm{Rect}(t)$) and its autocorrelation function ($\mathrm{Tri}(t)$). For reference the band limiting parameter b is shown for the values 1 and 2.

see that the longer the integration time (T_p), the more sensitive to frequency errors will the ACF become. In the last approximation it was assumed that the phase error $\Delta\phi(t)$ and the frequency error $\Delta\omega$ were assumed constant over the integration interval. For the single path case, which we are currently investigating, this is a good approximation. For the multipath case, however, this is not necessarily so and the integral over the phase error can significantly reduce the multipath error. This will be investigated in detail in the chapter on receiver modeling.

There is a simple correspondence between the complex notation used here (in eq. (2.23) and throughout this chapter) and the in-phase and quadrature phase channel. The real part of the complex correlation function is equal to the quadrature-phase channel and the imaginary part is corresponds to the in-phase channel. Further this is related to the two most common type of tracking loops: the coherent and the non-coherent delay-lock loop. The discriminator of the coherent delay-lock loop is constructed from the quadrature-phase components of the correlation function, whereas the non-coherent type is constructed from the sum of the squares of the I and Q components ($I^2 + Q^2$), which corresponds to multiplying the correlation function with the complex conjugate of itself.

Assuming the phase error to be zero (i.e. $\Delta\phi = 0$) and taking the real part of eq. (2.23), we obtain the signal model for quadra-phase channel in the receiver. This operation cor-

responds to what is done in a coherent receiver.

$$R^{I}_{\text{rect}}(\tau, \Delta\omega, \Delta\phi = 0) = \frac{1}{2} R_{\text{rect}}(\tau)\,\text{sinc}\,[\Delta\omega T_p/2] \tag{2.24}$$

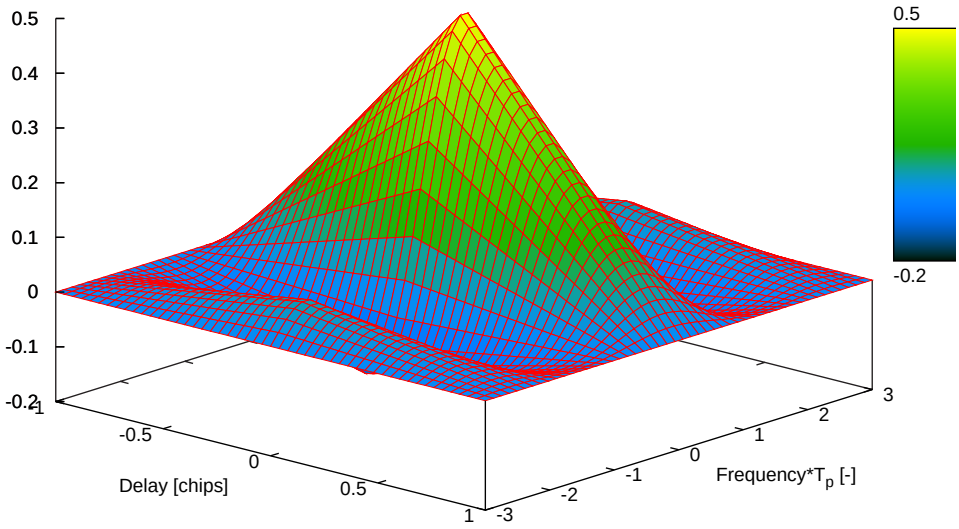

Figure 2.8.: ACF of the usual rectangular chip shape for the coherent case. The familiar triangle function is clearly seen for zero frequency error. Along $\tau = 0$ the amplitude of the ACF is proportional to the $\text{sinc}(\Delta\omega)$ function.

This, plotted as a function of τ and $\Delta\omega$, is shown in figure (2.8). In the figure the usual triangular form of the ACF for $\Delta\omega = 0$ is clearly recognized. In this limit (i.e. $\Delta\omega \to 0$) the sinc function approaches unity and we obtain the usual triangular ACF. This is the same result obtained in [Die96].

For $\tau = 0$ we have

$$R(\tau = 0, \Delta\omega, \phi = 0) = \frac{1}{2}\,\text{sinc}\,[\Delta\omega T_p/2] \tag{2.25}$$

This means that there are maxima, other than those corresponding to the correct carrier frequency. If we consider the non-coherent receiver we are effectively looking at the $I^2 + Q^2$, which amounts to multiplying eq. (2.23) with the complex conjugate of itself. We then obtain:

$$R^*(\tau, \Delta\omega)\overline{R^*(\tau, \Delta\omega)} = \frac{1}{4} R^2_{\text{rect}}(\tau)\,\text{sinc}^2\,[\Delta\omega T_p/2]\ . \tag{2.26}$$

As expected the dependence on the phase error is gone. The impact of all this can be seen in figure (2.9). Comparing with figure (2.8) we see that as a consequence of the squaring,

the ripples in the $\Delta\omega$ direction are by far not as pronounced. We shall be referring to these results, when considering the SOC and BOC signals later in this section.

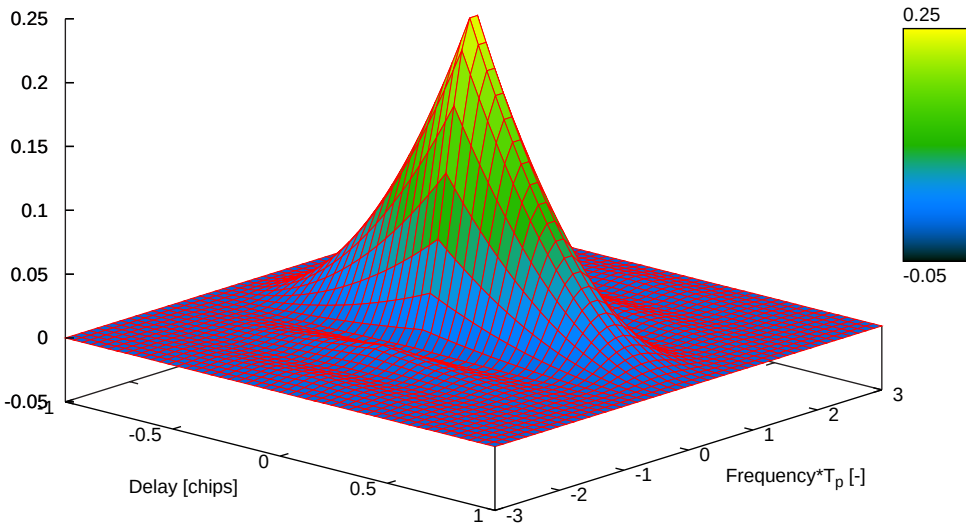

Figure 2.9.: ACF of the usual rectangular chip shape in the non-coherent case. The familiar triangle function is clearly seen for zero frequency error. Along $\tau = 0$ the amplitude of the ACF is proportional to the $\mathrm{sinc}^2(\Delta\omega)$ function.

Band-limited rectangular chip shape. Usually the signal cannot be emitted in infinite bandwidth. For example the entire spectrum of the GPS P(Y)-code is not emitted, or at least not received, but only the first main-lobe. Basically the C/A code is emitted through the same filter as the P(Y)-code, and as it only has $1/10$ the chip rate of the P(Y)-code, 10 side-lobes of its spectrum may be received and the chip-shape is practically rectangular.

How the spectrum is limited to a certain band of frequency depends on the filter used. Mostly the goal of a good band-limiting filter is to cut off the signal abruptly in frequency without distorting the phase. The most common filters types are the Butterworth-, Chebyshev-, Bessel- and Legendre-filters [LS73, Spi73]. In this thesis it will suffice to describe band limiting with an ideal band-limiting filter, which is simply described by the rectangular function in frequency. The band-limited rectangular signal is thus obtained by cutting off the spectrum symmetrically at $\pm 2\pi b/T_c$.

$$\begin{aligned} S_{\text{rect}}(\omega) &= \theta(|2\pi b/T_c| - \omega)\frac{1}{T_c}\left[\int \mathrm{e}^{-i\omega t}\, g_{\text{rect}}(t/T_c)dt\right] \\ &= \mathrm{sinc}(\omega T_c/2)\theta(|2\pi b/T_c| - \omega) \end{aligned} \tag{2.27}$$

The location of the cut-off is described by the parameter b, which is related to the bandwidth B_{BW} through the relation:

$$B_{BW} =: b\frac{2\pi}{T_c} \tag{2.28}$$

Thus for $b = 1$ exactly the first main-lobe of the spectrum is emitted and for $b \to \infty$, the entire spectrum is broadcasted and $c_{BL}(t)$ approaches the rectangular bit-shape. In figure (2.7) the values $b = 1$ and $b = 2$ are shown.

Transforming the band-limited spectrum eq. (2.27) back into the time domain yields the chip-shape for the band-limited rectangular signal (normalized to 1):

$$c_{BL}(t) = \frac{1}{\pi T_c}\left[\mathrm{Si}(2\pi b[t/T_c + 1/2]) - \mathrm{Si}(2\pi b[t/T_c - 1/2])\right] \tag{2.29}$$

where $\mathrm{Si}(\cdot)$ is the sine integral (see eq. (A.5)). The band limited chip shapes are shown in figure (2.10)

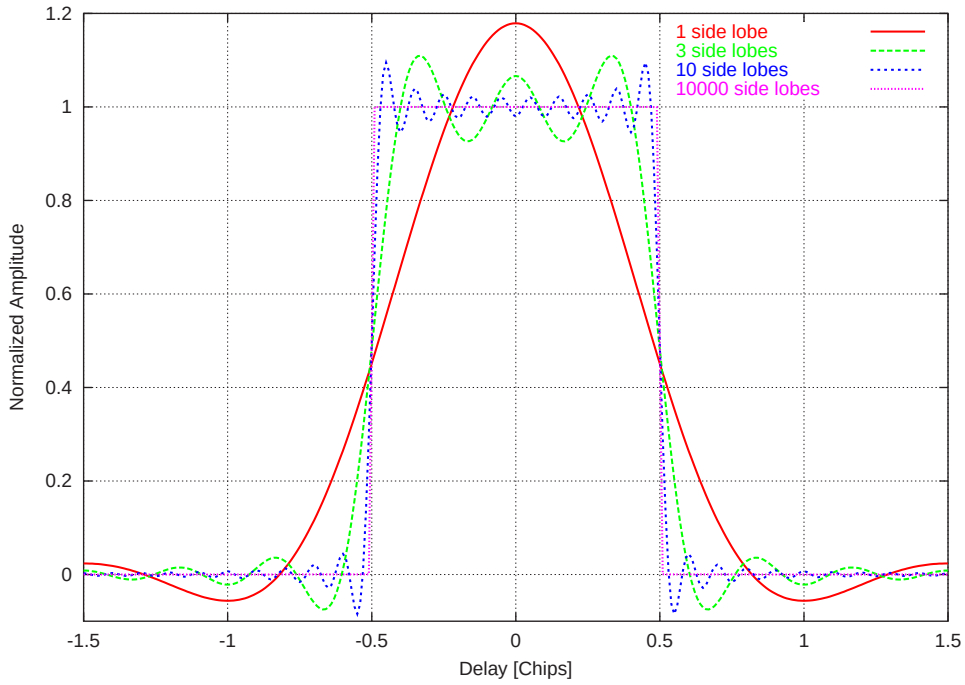

Figure 2.10.: Chip shape of a band limited rectangular chip for the values $b = 1, 3, 10$ and 10000 side-lobes

Assuming that the (P-code) receiver correlates the incoming signal with a code consisting of rectangular pulses, the correlation function can be approximated by:

$$R_{BL}(\tau) = \int c_{BL}(t)c(t-\tau)\,dt = \int_{x-1/2}^{x+1/2} c_{BL}(t)\,dt \tag{2.30}$$

Using a symbolic algebra tool (e.g. Mathematica or Maple V) this can be evaluated to:

$$\begin{aligned} R_{BL}(\tau) = &\frac{1}{\pi}(\tau+1)\,\mathrm{Si}[2\pi b(\tau+1)] + \frac{1}{2\pi^2 b}\cos[2\pi b(\tau+1)] \\ &+ \frac{1}{\pi}(\tau-1)\,\mathrm{Si}[2\pi b(\tau-1)] + \frac{1}{2\pi^2 b}\cos[2\pi b(\tau-1)] \\ &- \frac{2\tau}{\pi}\,\mathrm{Si}[2\pi b\tau] - \frac{1}{\pi^2 b}\cos[2\pi b\tau] \end{aligned} \tag{2.31}$$

As before the function $\mathrm{Si}(\cdot)$ is the sine integral. The first two derivatives of this function are generally of interest. The first derivative is needed for the implementation of the simulation and the second derivative is a dominating factor in the so-called Cramer–Rao bound.

Analogous to eq. (2.23) we now obtain the expression for the correlation function for the band-limited correlation process

$$R^*_{BL}(\tau, \Delta\omega) = R_{BL}(\tau)\frac{i\,\mathrm{e}^{i\phi}}{2}\,\mathrm{sinc}\,[\Delta\omega T_p/2] \tag{2.32}$$

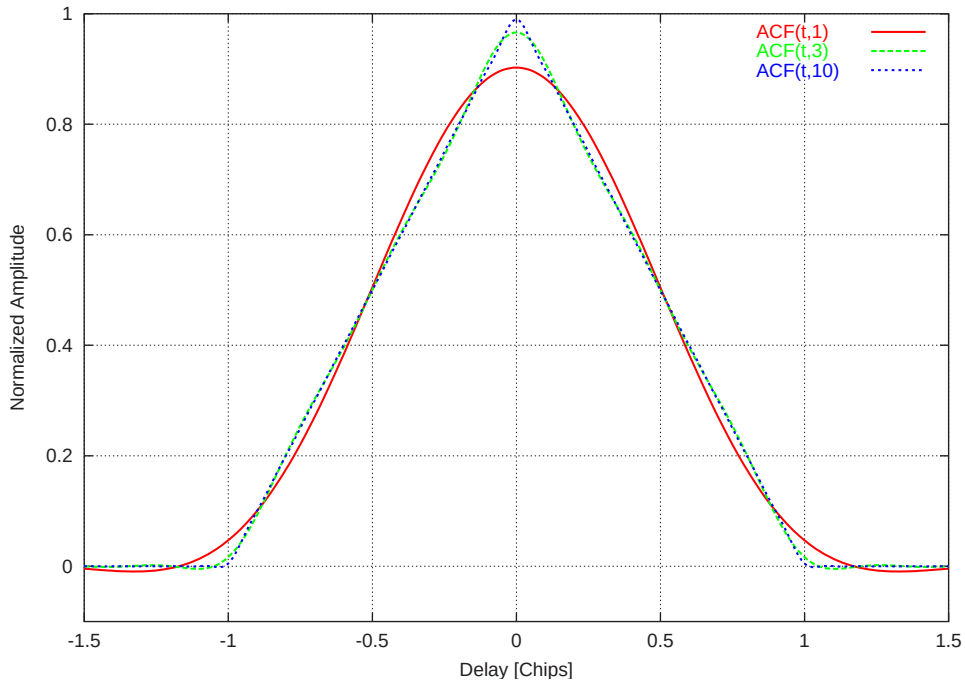

Figure 2.11.: The autocorrelation function of the band limited chip shape for values $b = 1, 3$ and 10.

The first derivative $R_{BL}(\tau)$ in eq. (2.31) is

$$R'_{BL}(\tau) = \frac{1}{\pi}\left[\mathrm{Si}[2\pi b(\tau+1)] + \mathrm{Si}[2\pi b(\tau-1)] - 2\,\mathrm{Si}[2\pi b\tau]\right] \tag{2.33}$$

and the second derivative is given by

$$R''_{BL}(\tau) = 2b\left[\mathrm{sinc}[2\pi b(\tau+1)] + \mathrm{sinc}[2\pi b(\tau-1)] - 2\,\mathrm{sinc}[2\pi b\tau]\right] \tag{2.34}$$

2.2.6. The Raised Cosine Chip Shape

Raised cosine pulse shaping, a well-known pulse shaping scheme in communications, is also considered here. The raised-cosine pulse-shaped bit stream in base-band can be written as [GLV93]

$$g_{RC}(t) := \frac{\sin(\pi t/T_c)\cos(\pi t/T_c\beta)}{\pi t/T_c(1-(2t/T_c\beta)^2)} \tag{2.35}$$

The limits corresponding to the roots of the denominator in eq. (2.35) exist.

The chip shape in eq. (2.35) is shown in figure (2.12). Looking at figure (2.12) we see that

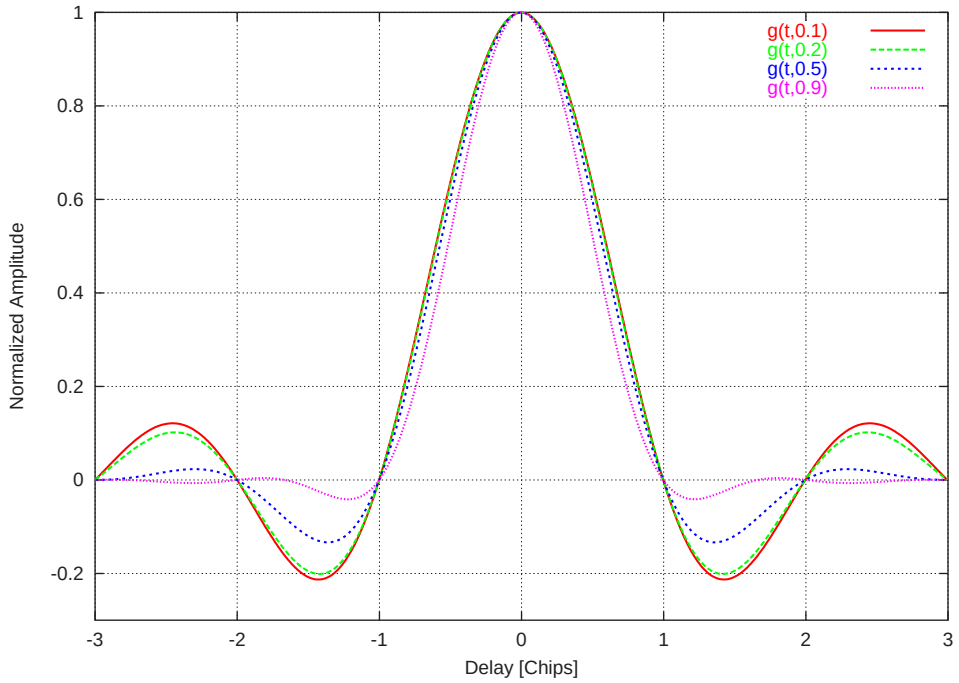

Figure 2.12.: The shapes of the raised cosine chips in time domain. The curves correspond to the following values for the roll-off factor $\beta = 0.1, 0.2, 0.5$ and 0.9.

the symbol is not zero for $\tau < -1$ and $\tau > 1$, as is the case for rectangular chips. This means that neighboring bits will interfere with the current bit. This effect becomes more pronounced for low values of the roll-off parameter β. For navigation this has serious consequences, in particular concerning multipath performance, as we shall see later.

The spectrum of the raised cosine chip shape is given by the following expression:

$$S_{RC}(\omega) = \begin{cases} \frac{1}{2\pi} & \omega \in [-\pi + \pi\beta, \pi - \pi\beta] \\ \frac{1-\cos[(w-\pi(1+\beta))/b]}{4\pi} & \omega \in [\pi - \pi\beta, \pi + \pi\beta] \\ \frac{1-\cos[(w+\pi(1+\beta))/b]}{4\pi} & \omega \in [-\pi - \pi\beta, -\pi + \pi\beta] \end{cases} \tag{2.36}$$

and this is shown in figure (2.13). In the limit $\beta \to 0$ the form of the spectrum approaches that of a rectangle. Consequently, the chip form approaches the $\mathrm{sinc}(t)$ function. In this sense the raised cosine is the „inverse" of the rectangular chip shape in this limit, i.e. the time and frequency domain representation have been interchanged.

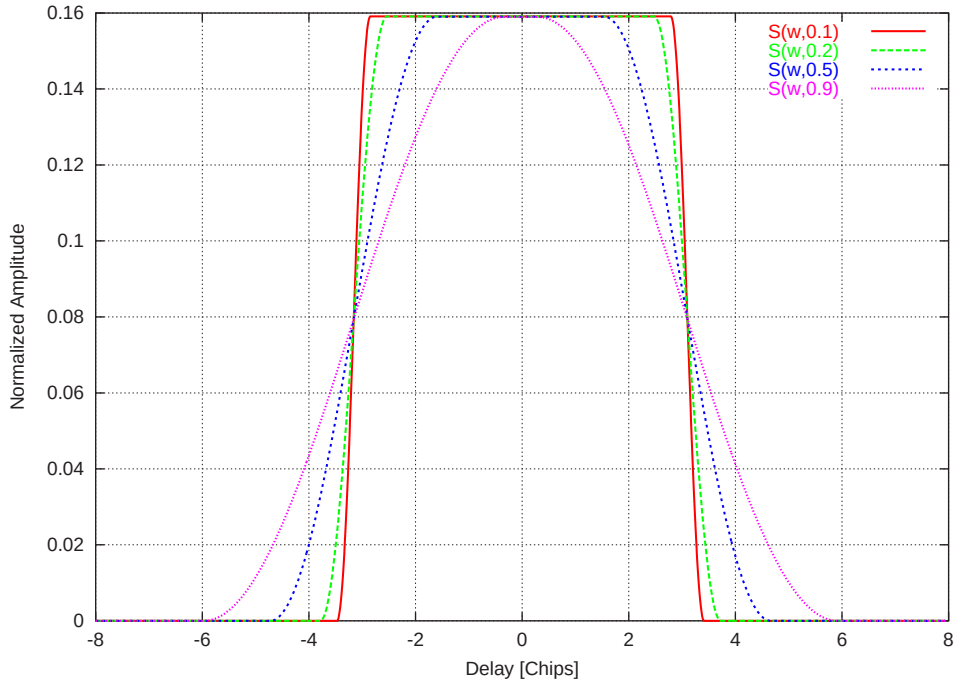

Figure 2.13.: The spectrum of the raised cosine chip shape. The curves correspond to the following values for the roll-off factor $\beta = 0.1, 0.2, 0.5$ and 0.9.

In earlier studies on the signal design of Galileo, a receiver design proposes that, in principle, the received raised cosine signal is sampled once per chip for each correlator. This means that a channel with three correlators, early, late and punctual sample the signal at $t = \tau - d/2T_c$, $t = \tau$ and $t = \tau + d/2T_c$, respectively.

Assuming that the received signal has been transformed down to base-band, i.e. the RF signal has been multiplied by $\mathrm{e}^{i\omega' t + i\phi'}$, then this process can be described mathematically

in the following way:

$$\begin{aligned}\bar{y}(t) &= \frac{1}{N_c}\left[\sum_{k=1}^{N_c} b_k \delta(t - kT_c - r)\right] \star \left[\sum_{k'=1}^{N_c} b_{k'} g_{T_c}(t - k'T_c + \tau(t)) \sin(\omega t + \phi)\, \mathrm{e}^{i(\omega' t + \phi')}\right] \\ &= \frac{1}{2iN_c} \sum_{k,k'} b_k b_{k'} g_{T_c}((k-k')T_c + \tau(t) + r)\, \mathrm{e}^{i\Delta\omega t + i\Delta\phi} \\ &= g_{T_c}(\tau(t) + r)\frac{1}{T_p}\int_{-T_p/2}^{T_p/2} \mathrm{e}^{i\Delta\omega t + i\Delta\phi}\, dt + \frac{1}{N_c}\sum_{\substack{k,k' \\ k\neq k'}} b_k b_{k'} g_{T_c}((k-k')T_c + \tau(t) + r) \\ &\xrightarrow[N_c\to\infty]{} g_{T_c}(\tau(t) + r)\frac{1}{T_p}\int_{-T_p/2}^{T_p/2} \mathrm{e}^{i\Delta\omega t + i\Delta\phi}\, dt\,,\end{aligned} \tag{2.37}$$

where r is a constant value ($-dT_c$, 0 or dT_c), depending on whether the early, punctual or late channel is meant. The star ($\star$) symbolizes convolution. In the last equation it was used that there are N_c identical terms for $k = k'$, which were separated from the main sum. Further we also have $b_k b_{k'} = 1$, for $k = k'$. The rest of the sum is bounded by an alternating sum of the form $\sum_k \frac{(-1)^k}{k}$, which converges. In the limit of infinite code length (i.e. $N_c \to \infty$) the last sum in eq. (2.37) vanishes.

2.2.7. Sinusoidal Offset Carrier Chips

In the previous section we have examined chip shapes due to filtering. However, it is also possible to shape the chip by modulation. In this and the next section such chip shapes are considered. In this section we examine a family of signals, which are generated by modulating each code-bit with a sinusoidal function. Such a signal was suggested in [Bet99] as an alternative to the BOC modulation (see section 2.2.8), but no analysis regarding it were found. In this thesis such a signal will be called a Sinusoidal Offset Carrier signal or SOC for short.

The base-band description of such a signal is given by:

$$g_{\mathrm{SOC}}(t) := \sqrt{2}\sin(\tfrac{2n\pi}{T_c}t)C(t) \tag{2.38}$$

where the sine modulates the code $C(t)$. The integer n corresponds to the number of periods of the sine wave that are contained in each code bit. The factor $\sqrt{2}$ was introduced in order to have the energy of the bit normalized to 1. In figure (2.14) the relation between the code bit and the modulating sub carrier is shown.

SOC Spectrum In deriving the spectrum of the signal in eq. (2.38) it is more convenient to use the convolution theorem than viewing the code chip $g_{\mathrm{SOC}}(t)$ as a single

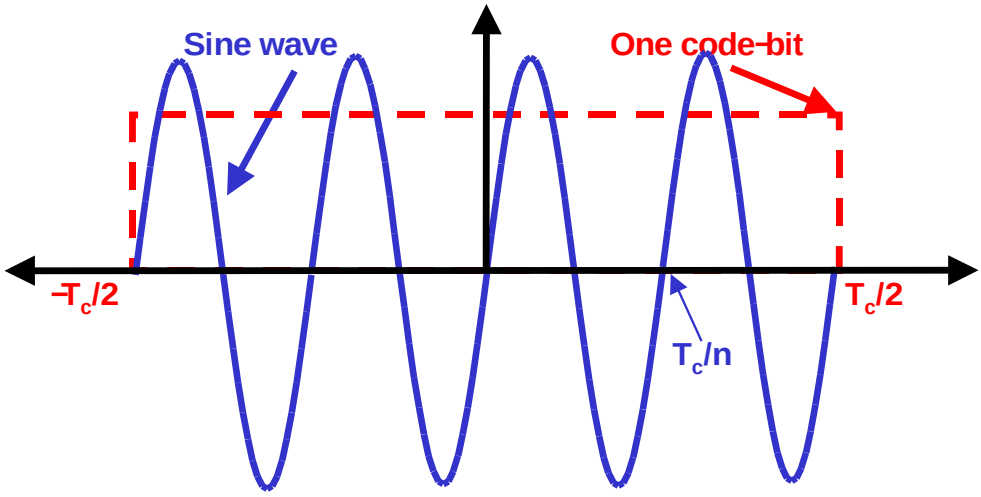

Figure 2.14.: Sinusoidal Offset Carrier. Each code bit is modulated with a sine wave with frequency $\frac{2n\pi}{T_c}$. The overshoot of the sine reflects the normalization factor $\sqrt{2}$

chip shape. According to the convolution theorem, the Fourier transform of the code chip can be expressed in terms of a convolution between the modulating carrier and the code bit and thus the problem is reduced to calculating the Fourier transforms for each signal separately, i.e. we have:

$$\widehat{S_{\text{SOC}}}(\omega) := \mathcal{F}\{g_{\text{SOC}}(t)\} = \mathcal{F}\{\sqrt{2}\sin(\tfrac{2n\pi}{T_c}t)\} \star \mathcal{F}\{C(t)\} \tag{2.39}$$

as before the star ($\star$) means convolution. We therefore have:

$$\begin{aligned}\mathcal{F}\{g_{\text{SOC}}(t)\} &= i\sqrt{2}\left[\delta(\omega + \tfrac{2n\pi}{T_c}) - \delta(\omega - \tfrac{2n\pi}{T_c})\right] \star T_c \operatorname{sinc}(\omega T_c/2) \\ &= i\sqrt{2}T_c\left[\operatorname{sinc}(\omega T_c/2 + n\pi) - \operatorname{sinc}(\omega T_c/2 - n\pi)\right]\end{aligned} \tag{2.40}$$

The role of n now becomes evident: The higher n is chosen, the further apart the peaks are separated.

Looking at eq. (2.40) we see that the modulation with a sine as a sub-carrier is equivalent to transmitting two absolutely synchronized signals on the carrier frequencies $\omega + \omega_\pm$ where $\omega_\pm := \pm n2\pi/T_c$. The fact that the two signals have different signs can be interpreted as a phase difference of π between the two carriers.

The spectrum of the SOC signal is shown in figure (2.15)

The power spectrum of this signal is simply obtained by:

$$\begin{aligned}P_{\text{SOC}}(\omega) &:= |S_{\text{SOC}}(\omega)|^2 = S^*_{\text{SOC}}(\omega)S_{\text{SOC}}(\omega) \\ &= 2T_c^2\left[\operatorname{sinc}(\omega T_c/2 + n\pi) - \operatorname{sinc}(\omega T_c/2 - n\pi)\right]^2 \\ &= 2T_c^2\Big[\operatorname{sinc}^2(\omega T_c/2 + n\pi) + \operatorname{sinc}^2(\omega T_c/2 - n\pi) \\ &\quad - 2\operatorname{sinc}(\omega T_c/2 - n\pi)\operatorname{sinc}(\omega T_c/2 + n\pi)\Big]\end{aligned} \tag{2.41}$$

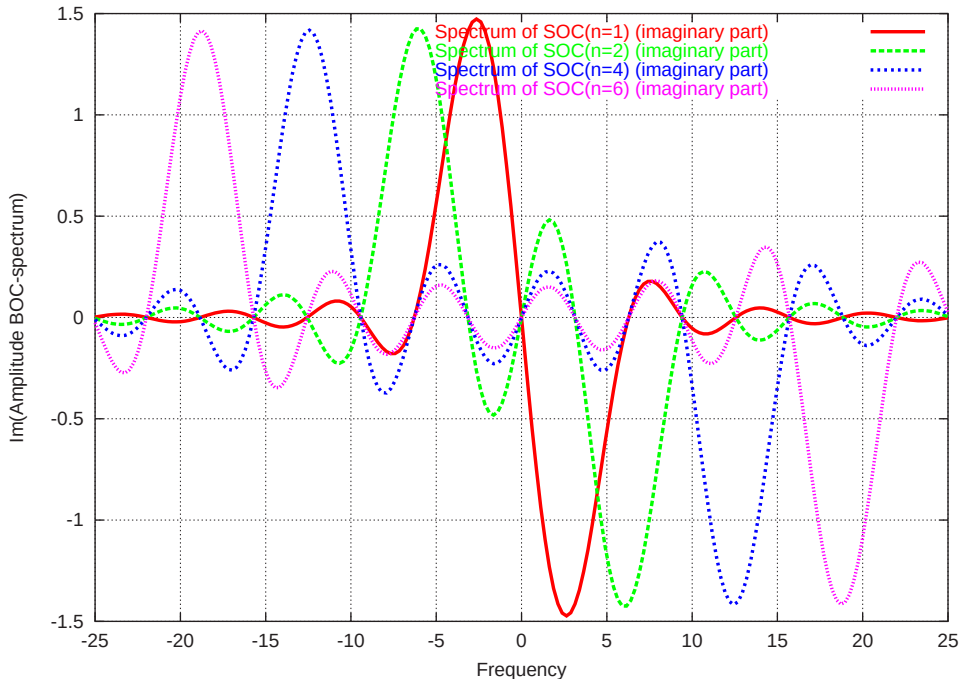

Figure 2.15.: The spectrum of a Sinusoidal Offset Carrier signal. The curves shown represent the values $n = 1, 2, 4$ and 6.

The power spectrum (logarithmic scale) is shown in figure (2.16)

SOC Spectrum Maxima As was mentioned earlier, the SOC signal can be considered as two coherent normal rectangular chips with a relative phase difference of π. This suggest that it may be possible to track those „virtual" signals individually. The first question that comes to mind is: Where are the maxima of the spectra of eq. (2.40)? This is in principle easily answered, but the answer cannot be given in a closed form. To find the maxima, eq. (2.40) is differentiated and the result is set equal to 0. Performing the analysis in base-band (i.e. $\omega = 0$), the result is:

$$\begin{aligned}\frac{\partial S_{\mathrm{SOC}}(\omega)}{\partial \omega} &= \frac{\cos(\omega T_c/2 - n\pi)}{\omega T_c/2 - n\pi} - \frac{\sin(\omega T_c/2 - n\pi)}{(\omega T_c/2 - n\pi)^2} \\ &\quad - \frac{\cos(\omega T_c/2 + n\pi)}{\omega T_c/2 + n\pi} + \frac{\sin(\omega T_c/2 + n\pi)}{(\omega T_c/2 + n\pi)^2} = 0 \\ &\Rightarrow \tfrac{1}{4T_c^2}\omega^2 - n^2\pi^2 - \omega T_c \tan(\omega T_c/2) = 0\,.\end{aligned} \tag{2.42}$$

Expanding the tangens to third order around[6] $\omega T_c/2 = n\pi$ we obtain the roots $\omega T_c = 2n\pi$ and $\omega T_c = n\pi \pm \sqrt{n^2\pi^2 - 6}$ of which only the second with the $+$ sign is usable. If we

[6]We expand the tangens here because we are interested in the maximum close to $\omega = 2n\pi T_c$. $\tan(x) \approx x + x^3/3$, because the $\tan(x)$ has a period of π this expansion is also valid around $x + n\pi$

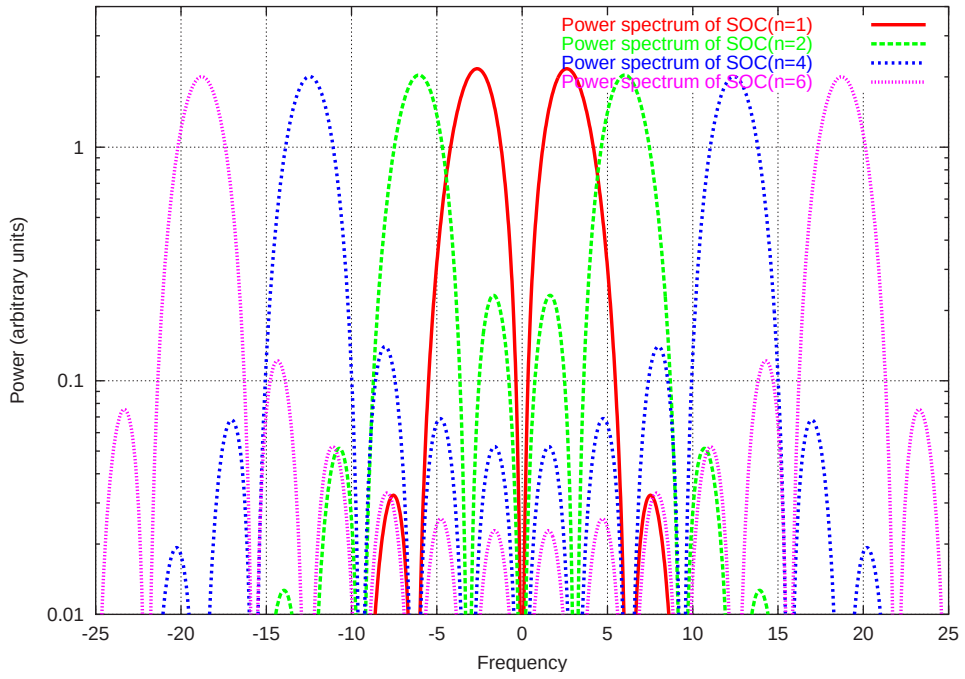

Figure 2.16.: The power spectrum of the SOC signal for $n = 1, 2, 4$ and 6.

now define the maximum frequency as

$$\left.\frac{\partial S_{\text{SOC}}(\omega)}{\partial \omega}\right|_{\omega =: \omega_{\max\pm}} = 0 \tag{2.43}$$

then using the third order approximation of the tangens we have:

$$\omega_{\max\pm} \approx \pm \frac{n\pi + \sqrt{n^2\pi^2 - 6}}{T_c} \tag{2.44}$$

So the maximum of the spectrum is *not* located at $\omega T_c = n2\pi$, but somewhere near the frequency given by eq. (2.43). Later in this section we will explore the possibility to track "half" the spectrum in eq. (2.40). We will see that the maximum of the cross correlation functions will be located at $\omega = n2\pi$ and not at $\omega_{\max\pm}$ as the form of eq. (2.40) might suggest.

The SOC autocorrelation function Assuming a frequency offset of $\Delta\omega$ we can calculate the ACF in a similar way as for the usual GPS-like signal in eq. (2.23) using eq. (2.18). We then get:

$$\begin{aligned} R^*_{\text{SOC}}(\tau, \Delta\omega) &= \frac{1}{T_p} \int_{-T_p/2}^{T_p/2} g_{\text{SOC}}(t-\tau) g_{\text{SOC}}(t)\, \mathrm{e}^{i(\omega+\Delta\omega)t + i\phi} \sin(\omega t) dt \\ &\approx R_{\text{SOC}}(\tau) \frac{i\, \mathrm{e}^{i\phi}}{2} \operatorname{sinc}\left[\Delta\omega T_p/2\right] \end{aligned} \tag{2.45}$$

The auto correlation function $R_{\mathrm{SOC}}(\tau)$ above can be obtained directly by a convolution in time domain: the ACF is simply a convolution of the signal with a time-reversed version of itself. It can also be calculated by an inverse Fourier transform of the power spectrum. We then get:

$$\begin{aligned} R_{\mathrm{SOC}}(\tau) &:= \int g_{\mathrm{SOC}}(t-\tau) g_{\mathrm{SOC}}(t) dt = \mathcal{F}^{-1}\mathcal{F}\{g_{\mathrm{SOC}}(t) \star g_{\mathrm{SOC}}(-t)\} \\ &= \mathcal{F}^{-1}\left\{\mathcal{F}\{g_{\mathrm{SOC}}(t)\}\mathcal{F}\{g_{\mathrm{SOC}}(-t)\}\right\} = \mathcal{F}^{-1}\{\widehat{g_{\mathrm{SOC}}}(\omega)\widehat{g_{\mathrm{SOC}}}^{*}(\omega)\} \end{aligned} \tag{2.46}$$

If we take a closer look at eq. (2.40), we can identify the terms and factors needed to perform the inverse Fourier transform implied in eq. (2.46). The first two terms in eq. (2.40) can be transformed by looking them up directly, but the last term needs some more consideration. The first two terms are of the form

$$\mathcal{F}^{-1}\hat{f}(\omega) = \mathcal{F}^{-1}\left\{\frac{\sin^2(a\omega + n\pi)}{(a\omega + n\pi)^2}\right\} = \frac{1}{a}\,\mathrm{Tri}(t/a)\,\mathrm{e}^{-in\pi t/a} \tag{2.47}$$

where $\mathrm{Tri}(x)$ is the triangular function with support in the interval $[-2, 2]$ and maximum amplitude $1/2$ (see eq. (A.4) and eq. (B.6)).

The last term in eq. (2.41) can be written as:

$$\begin{aligned} &\mathrm{sinc}(\omega T_c/2 - n\pi)\,\mathrm{sinc}(\omega T_c/2 + n\pi) \\ &\qquad = \frac{\sin(\omega T_c/2 - n\pi)}{\omega T_c/2 - n\pi}\frac{\sin(\omega T_c/2 + n\pi)}{\omega T_c/2 + n\pi} \\ &\qquad = \frac{\sin^2(\omega T_c/2)}{(\omega T_c/2 - n\pi)(\omega T_c/2 + n\pi)} \\ &\qquad = \frac{\sin^2(\omega T_c/2)}{2n\pi}\left(\frac{1}{\omega T_c/2 - n\pi} - \frac{1}{\omega T_c/2 + n\pi}\right). \end{aligned} \tag{2.48}$$

The second equation is valid because of $2\sin(x+n\pi)\sin(x-n\pi) = \cos(n2\pi) - \cos(2x) = 1 - \cos(2x) = 2\sin^2(x)$. Now the critical term in eq. (2.41) has been expanded using partial fractions and the resulting terms can now be evaluated:

$$\mathcal{F}^{-1}\left\{\frac{\sin^2(a\omega)}{a\omega \pm n\pi}\right\} = \frac{-i\,\mathrm{e}^{\mp in\pi t/a}}{4a}\left[\theta(t-2a) + \theta(t+2a) - 2\theta(t)\right] \tag{2.49}$$

where $\theta(\cdot)$ is the Heaviside step function. The construction in the parenthesis is simply a square wave. The entire term is then given by:

$$\mathcal{F}^{-1}\left\{\frac{\sin^2(a\omega)}{a\omega - n\pi} - \frac{\sin^2(a\omega)}{a\omega + n\pi}\right\} = \frac{\sin(n\pi t/a)}{2a}\left[\theta(t-2a) + \theta(t+2a) - 2\theta(t)\right] \tag{2.50}$$

Scaling the above equation correctly and putting it all together, we get:

$$\begin{aligned}\frac{\pi^2}{8T_c}R_{\text{SOC}}(\tau) &= \text{Tri}(2\tau/T_c)\left[\text{e}^{in\pi T_c/2}+\text{e}^{-in\pi T_c/2}\right] \\ &\quad - \frac{\sin(n2\pi\tau/T_c)}{n2\pi}\left[\theta(\tau-T_c)+\theta(\tau+T_c)-2\theta(\tau)\right] \\ &= \text{Tri}(2\tau/T_c)\cos(n\pi T_c/2) \\ &\quad - \frac{\sin(n2\pi\tau/T_c)}{n2\pi}\left[\theta(\tau-T_c)+\theta(\tau+T_c)-2\theta(\tau)\right]\end{aligned} \tag{2.51}$$

The ACF in eq. (2.46) is drawn in figure (2.17).

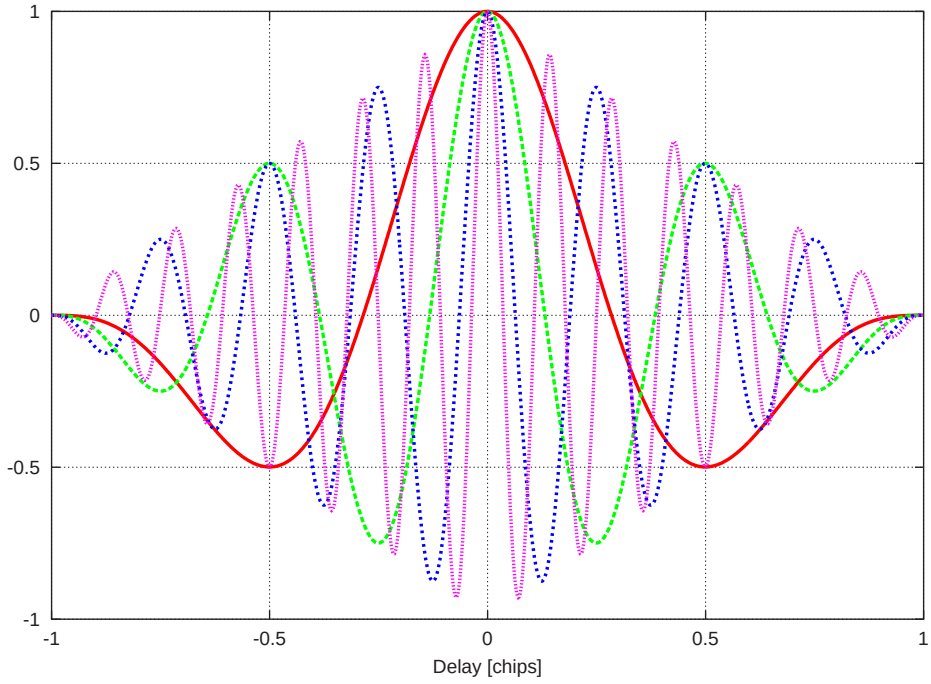

Figure 2.17.: ACF of the Sinusoidal Offset Carrier (SOC) for $n = 1, 2$ and 4. $n = 1$ corresponds to Manchester coding (if the modulating wave were square). The graph was done using eq. (2.46). As in the previous figure the frequency is measured in inverse chip lengths.

Correlation of the SOC Signal with the Code In this section the possibility is considered to track just one peak of the spectrum in eq. (2.40). This has been suggested in various places in the literature [FB00, BBC+00], but to my knowledge no analysis has been presented. This has been discussed as a possible means to acquire the BOC signal[7].

[7] Here we are considering the SOC signal, but the same arguments apply. In the next section we will concentrate on the BOC signal.

However, the procedure described in [FB00, BBC$^+$00] involves filtering the side-bands and then acquiring them. In this section a correlation function is derived that effectively correlates (mainly) with one of the side bands.

Looking at the form of eq. (2.40) we notice that the spectrum is basically a linear combination of the original code ($C(t)$) modulated onto two different carriers. If one is filtered out with a band-pass filter in the receiver, then the situation is very similar to the case of rectangular chips. However, the frequency response may be different with and without the side-band filter (i.e. the filter picking out the appropriate side-band.). Thus, after the acquisition when the side-band filter is turned off, the transition from the side-band acquisition (or tracking) may be more complicated due to the difference in frequency response. Therefore, we suggest to correlate the unfiltered, received (split spectrum) signal with a reference signal consisting only of the code. It is important to note that as the code alone and the code modulated with the splitting wave (square or sinusoidal) are perpendicular, this cannot happen on the central carrier frequency ω. However, the spectrum suggest that there may be the possibility to track the signal around $\omega \pm n\pi T_c$ or $\omega_{\text{max}\pm}$.

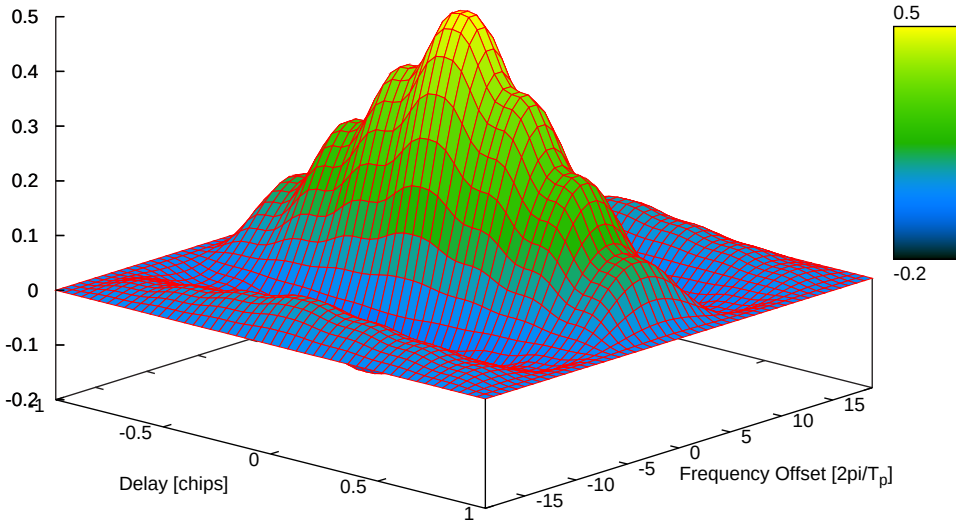

Figure 2.18.: 3-D plot of the coherent CF for $n = 2$. The delay is measured in chip durations, while the frequency offset is measured in pre-detection bandwidths. The plot is based on eq. (2.57). Note that for $\pm n$ we get different signs of the peaks of the CF. This reflects the phase difference of π between the side-bands mentioned in the text.

Unfortunately, we can't use the same base-band approach as we did for the other signals, because we are not correlating on the central carrier frequency. Furthermore it is not obvious what the carrier frequency is; is it $\omega T_c \pm n\pi$ or is it $\omega \pm \omega_{max,\pm}$? Therefore we choose to carry out the analysis around $\omega + \Delta\omega$ and start with the RF signal.

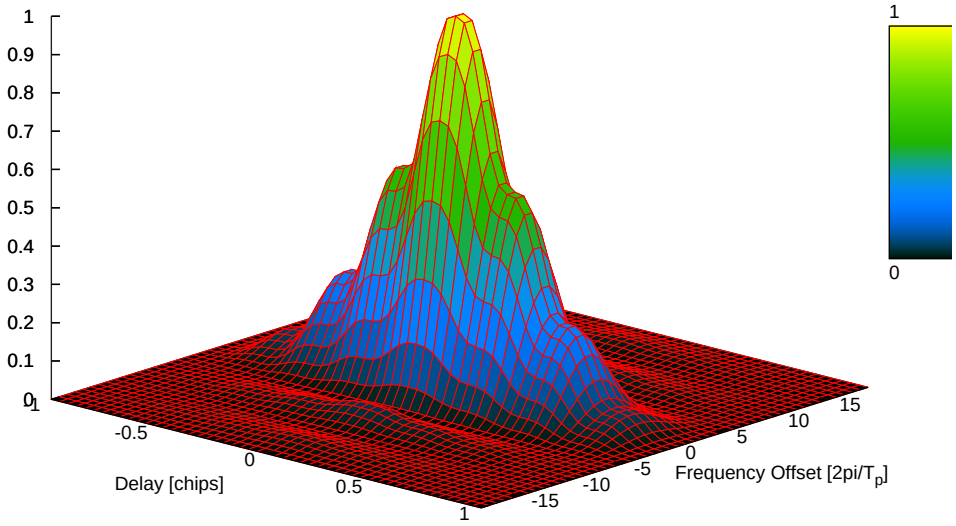

Figure 2.19.: 3-D plot of the non-coherent CF for $n = 2$. The delay is measured in chip durations, while the frequency offset is measured in pre-detection bandwidths. The plot is based on eq. (2.58). Note that for $\pm n$ we get different signs of the peaks of the CF. This reflects the phase difference of π between the side-bands mentioned in the text.

Let's now assume we modulate the code with a sine wave and that the RF signal is a sine wave as well. The RF signal could just as well be placed on a cosine and that would correspond to the channel *emitted* in quadrature. We choose a sine (without loss of generality) for mathematical convenience, the quadrature channel can be obtained by multiplying with a phase factor.

$$\begin{aligned} S^*(t) &:= \sqrt{2}\sin(n\omega_{T_c}t)C(t)\sin(\omega t+\phi) = \frac{-C(t)}{2\sqrt{2}}[\mathrm{e}^{in\omega_{T_c}t}-\mathrm{e}^{-in\omega_{T_c}t}][\mathrm{e}^{i\omega t+\phi}-\mathrm{e}^{-i\omega t-\phi}] \\ &= \frac{C(t)}{\sqrt{2}}\left\{\cos[(\omega-n\omega_{T_c})t+\phi]-\cos[(\omega+n\omega_{T_c})t+\phi]\right\} \\ &=: \frac{1}{\sqrt{2}}\left[S_n^-(t)-S_n^+(t)\right] \end{aligned} \tag{2.52}$$

where $\omega_{T_c} := \frac{2\pi}{T_c}$ and T_c is measured in time units. This simple math reveals an interesting result. We already knew that the „virtual" signals had a relative phase shift of π, but they are further $\pm\frac{\pi}{2}$[rad] out of phase relative to the composite signal.

Now let us take a look at how a correlation with a single sub-carrier can be realized. Assuming we want to track the signal on carrier $\omega' := \omega + \Delta\omega$. In the receiver we

generate a carrier with the frequency ω' and modulate it with the spreading code only. Then using eq. (2.52) we get:

$$r^*(t,\tau) := \frac{\mathrm{e}^{i\omega' t + i\phi'}}{\sqrt{2}} C(t-\tau)\left[S_n^-(t) - S_n^+(t)\right] \tag{2.53}$$

here the function $r^*(\cdot)$ is the integrand of the correlation function. The star means that both I- and Q-channels are contained in the expression. Dropping all high frequency terms, we have:

$$\begin{aligned} r^*(t,\tau,\Delta\omega,n) &= \frac{-\mathrm{e}^{i\Delta\phi}}{2\sqrt{2}} C(t)C(t-\tau)\left[\mathrm{e}^{i(\Delta\omega + n\omega_{T_c})t} - \mathrm{e}^{i(\Delta\omega - n\omega_{T_c})t}\right] \\ &= r^*_{\text{code}}(t,\tau)\left[\mathrm{e}^{i(\Delta\omega_+)t} - \mathrm{e}^{i(\Delta\omega_- t)}\right] , \end{aligned} \tag{2.54}$$

where $\Delta\omega_\pm := \Delta\omega \pm n\omega_{T_c}$ is a small frequency deviation from $\omega \pm n\omega_{T_c}$ rather than the carrier frequency itself. Before we can proceed to integrate eq. (2.54), the integrand must be rearranged because the $\mathrm{e}^{i(\Delta\omega_\pm)t}$ can not be assumed to vary slowly within an interval $[kT_c, (k+1)T_c]$. This was an assumption needed for the approximation in eq. (2.18). Therefore we start with the full expression of the integral[8]

$$\begin{aligned} &R(\tau,\Delta\omega_\pm,\phi) \\ &\quad= \int_{-T_p/2}^{T_p/2} \sum_k C(t-kT_c)C(t-kT_c-\tau)\sqrt{2}\sin(n\omega_{T_c}t)\sin(\omega t+\phi)\,\mathrm{e}^{i\omega' t + i\phi'}\,dt \\ &\quad= \frac{1}{\sqrt{2}}\int_{-T_p/2}^{T_p/2} \sum_k C(t-kT_c)C(t-kT_c-\tau)\sin(n\omega_{T_c}t)\,\mathrm{e}^{i\Delta\omega t + i\Delta\phi}\,dt \\ &\quad= \frac{1}{\sqrt{2}}\int_{-T_p/2}^{T_p/2} \sum_k C(t-kT_c)C(t-kT_c-\tau)\sin(n\omega_{T_c}t)\,\mathrm{e}^{in\omega_{T_c}(t-kT_c)}\,\mathrm{e}^{i\Delta\omega_\pm t + i\Delta\phi}\,dt \\ &\quad\approx \frac{1}{\sqrt{2}}\int C(u)C(u-\tau)\sin(n\omega_{T_c}u)e^{in\omega_{T_c}u}du \int_{-T_p/2}^{T_p/2} \mathrm{e}^{i\Delta\omega_\pm t + i\Delta\phi}\,dt \end{aligned} \tag{2.55}$$

In the second equation the high frequency terms were thrown away and in the third equation $\Delta\omega_\pm$ from above was substituted. Thus the variation of the term $\mathrm{e}^{i\Delta\omega_\pm t}$ can be considered slow compared to T_c. With this trick the rapid variations in the oscillating term is written as a factor periodic in T_c and a factor which now varies slowly compared to T_c. Further the integrand was multiplied with $1 = e^{\pm ink\omega_{T_c}T_c}$. In the last equation a change of variable was made and the same approximation applied as in eq. (2.18).

We see that there is a kind of „resonance frequency" where the side band can be tracked and this happens when the reference frequency is a multiple of ω_{T_c}, not just at $n\omega_{T_c}$.

[8] The off-diagonal terms in the sum are suppressed

In the standard situation for rectangular chips there are also similar kinds of „resonance frequencies" at $\omega + k2\pi/T_p$, corresponding to the maximums of the $\mathrm{sinc}(\cdot)$ factor. This effect is also present here in the second integral in the last equation in eq. (2.55). There is a fundamental difference between these two phenomena: one depends on the integration time, which is generated in the *receiver* and is thus not very stable and the second depends on the chip rate of the signal generated in the *satellite* and is therefore just as stable as the carrier frequency itself.

One might have expected that the „resonance frequency" would correspond to the maximum of the spectrum (i.e. $\omega_{\max\pm}$ in eq. (2.43)), but that is not the case.

Integrating the last equation in eq. (2.55) and assuming the phase error to be constant in the integration interval, we get:

$$\begin{aligned} R^{\text{code}}_{\text{SOC}}(\tau, \Delta\omega_\pm, n) &= \\ &= \frac{\pm T_p \, \mathrm{e}^{i\Delta\phi}}{\sqrt{2}} \, \mathrm{sinc}(\Delta\omega_\pm T_p/2) \left[\mathrm{Tri}(\tau/T_c) \mp \frac{\mathrm{e}^{\pm i n \omega_{T_c} \tau}}{n\omega_{T_c}} \sin[n\omega_{T_c}(T_c - |\tau|)] \right] \end{aligned} \tag{2.56}$$

Taking the real part, we get the expression for the coherent receiver.

$$\begin{aligned} R^{\text{code}}_{\text{SOC}}(\tau, \Delta\omega_\pm, n) &= \\ &= \frac{\pm T_p \, \mathrm{e}^{i\Delta\phi}}{\sqrt{2}} \, \mathrm{sinc}(\Delta\omega_\pm T_p/2) \left[\mathrm{Tri}(\tau/T_c) \mp \frac{\cos(n\omega_{T_c}\tau)}{n\omega_{T_c}} \sin[n\omega_{T_c}(T_c - |\tau|)] \right] \end{aligned} \tag{2.57}$$

Setting $\Delta\omega_\pm = 0$ results in a positive or negative triangular CF with a „disturbance" due to the other side-band. Furthermore, the absolute extrema of eq. (2.57) is located at $(\tau, \Delta\omega) = (0, n\pi\omega_{T_c})$ and not at the corresponding maximum frequency defined in eq. (2.43).

Comparing the equation above to eq. (2.23) we see similarities. However, it is not straight forward to view eq. (2.57) as a generalization of eq. (2.23); setting $n = 0$ in eq. (2.57) does *not* yield eq. (2.23). As the separation between the peaks in the spectrum increases, the mutual influence decreases, so the limit $n \to \infty$ would in this sense correspond to eq. (2.23).

As before the non-coherent expression is obtained by multiplying the correlation function by the complex conjugate of itself:

$$\begin{aligned} R^{\text{code}}_{\text{SOC}}(\tau, \Delta\omega_\pm, n) \overline{R^{\text{code}}_{\text{SOC}}(\tau, \Delta\omega_\pm, n)} &= \\ \frac{T_p^2}{2} \, \mathrm{sinc}^2(\Delta\omega_\pm T_p/2) \Big[\mathrm{Tri}^2(\tau/T_c) - \frac{2}{n\omega_{T_c}} \cos(n\omega_{T_c}\tau) \sin[n\omega_{T_c}(T_c - |\tau|)] & \\ + \frac{1}{n^2\omega_{T_c}^2} \sin^2[n\omega_{T_c}(T_c - |\tau|)] \Big] & \end{aligned} \tag{2.58}$$

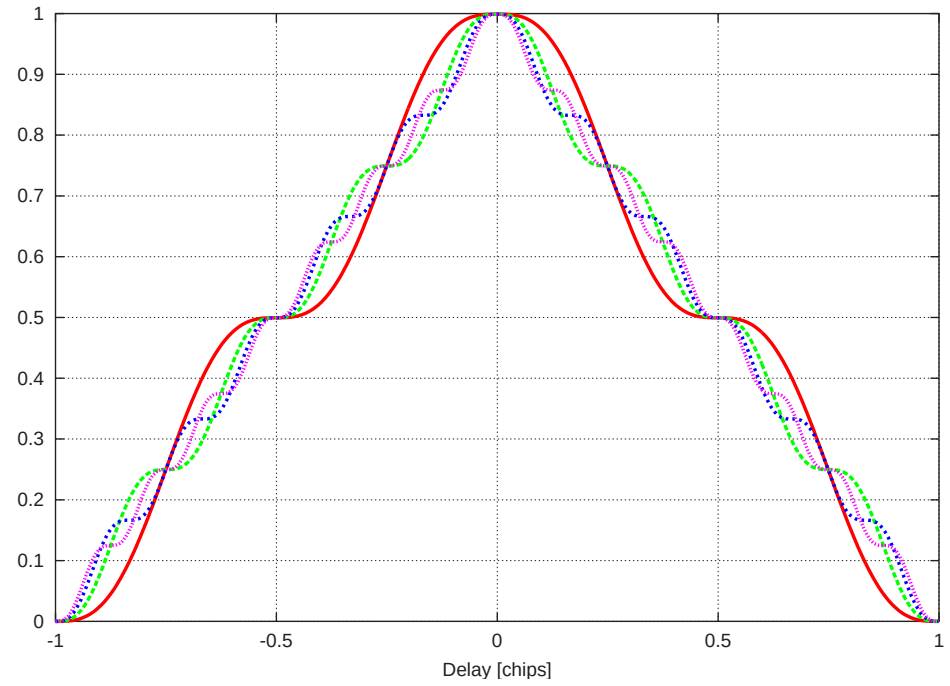

Figure 2.20.: Coherent correlation function of the sub-carrier for n =1, 2, 3 and 4. The strange wavy structure is due to the cross-talk from the other side-band. A look at the equations reveals that immediately.

As expected, the non-coherent expression of the correlation function is independent on the phase error.

In figure (2.20) the coherent correlation function eq. (2.57) is shown for the maximum carrier frequency for $\Delta\omega_{\pm} = 0$. In figure (2.21) the corresponding results for the non-coherent correlation function are shown.

We notice the wavy structure of the correlation functions in figure (2.21) and figure (2.20). These are due to the cross-talk from the coherent signal on the other side band.

When constructing a discriminator for the receiver S-curve, care must be taken when choosing the correlator spacing, because if chosen badly, the tracking point of the discriminator may be located on one of the flat regions of the correlation function. We will go further into that in the chapter on the receiver model.

2.2.8. Binary Offset Carrier Chips

The new military signals of the GPS system will feature a so-called split spectrum signal, also called binary offset carrier (BOC) signal. For the European Galileo system such modulations are also under consideration.

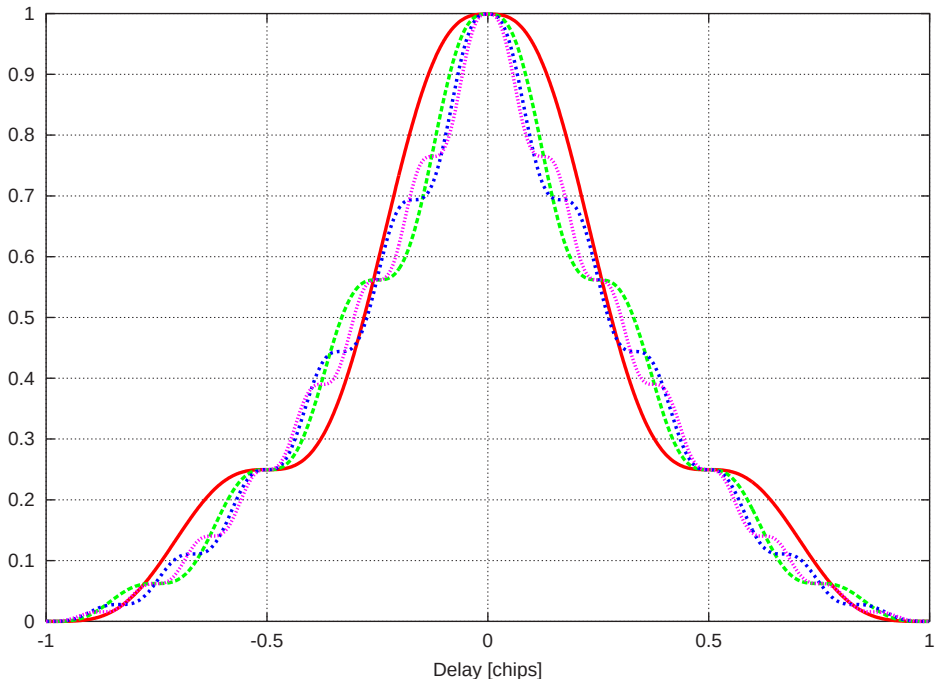

Figure 2.21.: Non-coherent correlation function of the sub-carrier for n =1, 2, 3 and 4. The strange wavy structure is due to the cross-talk from the other side-band. A look at the equations reveals that immediately.

This signal can be considered a generalization of the Manchester encoded signal. The splitting of the spectrum is implemented by modulating the spreading-code onto a square wave, replacing the sine wave of the SOC signal from the previous section. The square wave has a very short period, namely one chip length in the case of Manchester coding and if the spectrum is to be split farther apart, the square wave has an even shorter period. Due to the short period, the spectrum of the square wave is discrete and the first delta-peaks are located at $\pm 2\pi/T_c$. The other peaks themselves are outside of the spectrum, but in conjunction with the spectrum of the spreading code they will contribute to the over-all spectrum. The spectrum of the combined signal can now be calculated using the convolution theorem.

Another way of looking at it is simply to say that each symbol has the shape of a square-wave. The spectrum in the limit of infinite code-length is then simply the spectrum of one symbol.

Let us now consider a general BOC-signal, where each symbol is given by the square-wave:

$$\mathrm{Sqr}_n(t) := (-1)^n[\theta(x + T_c/2) + \theta(x - T_c/2)] + 2\sum_{k=-n+1}^{n-1} \theta(t + kT/2n)\,. \qquad (2.59)$$

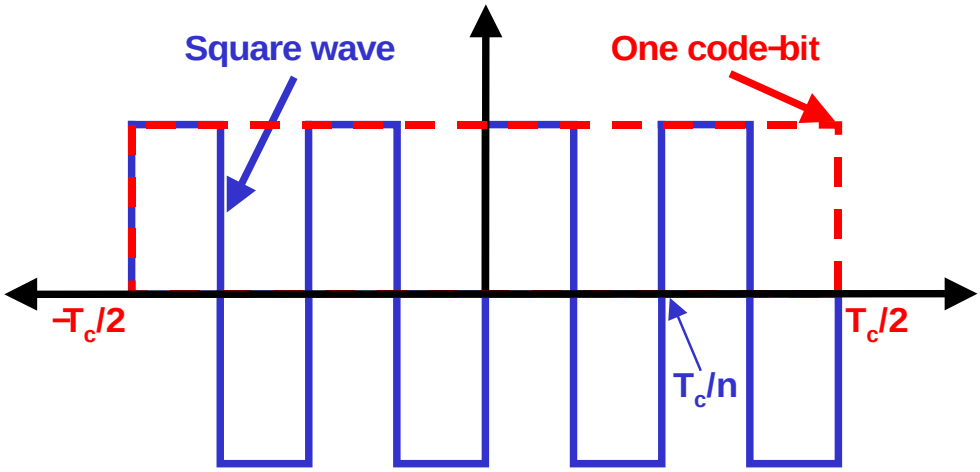

Figure 2.22.: Modulating square wave for splitting up the spectrum

For n even the square wave can be written as

$$\mathrm{Sqr}_n(t) = \begin{cases} 1 & t \in [(2k-n)t_c, (2k-n+1)t_c] \\ -1 & t \in [(2k-n+1)t_c, (2(k+1)-n)t_c] \end{cases} \quad \text{for} \quad k \in [0, .., n-1] \tag{2.60}$$

where $t_c := T_c/n$. The number n counts the number of periods of the square wave within one chip. The notation used here and the notation, which has established itself in the literature are then related by BOC($\frac{n}{T_c}$, $\frac{1}{T_c}$) (see for example [BBC+00, Bet00b]).

Then the Fourier transform corresponding to eq. (2.60) can be written as

$$\begin{aligned} S_{\mathrm{BOC}}(\omega) = \mathcal{F}\left\{\mathrm{Sqr}_n(t)\right\} &= \sum_{k=0}^{n-1} \left[\int_{(2k-n)t_c}^{(2k-n+1)t_c} \mathrm{e}^{-i\omega t}\, dt - \int_{(2k-n+1)t_c}^{(2(k+1)-n)t_c} \mathrm{e}^{-i\omega t}\, dt \right] \\ &= \frac{i\,\mathrm{e}^{i\omega(n-1)t_c}}{\omega} 4\sin^2(\omega t_c/2) \sum_{k=0}^{n-1} \mathrm{e}^{-i2k\omega t_c} \ . \end{aligned} \tag{2.61}$$

Identifying the sum in the eq. (2.61) as a geometric series (see eq. (B.2)) we have

$$\begin{aligned} S_{\mathrm{BOC}}(\omega) &= \frac{i\,\mathrm{e}^{i\omega(n-1)t_c}}{\omega} 4\sin^2(\omega t_c/2) \frac{1-\mathrm{e}^{-i2\omega n t_c}}{1-\mathrm{e}^{-i2\omega t_c}} \\ &= \frac{4i}{\omega}\sin^2(\omega t_c/2) \frac{\sin(n\omega t_c)}{\sin(\omega t_c)} \\ &= \frac{2i}{\omega}\sin(\omega t_c/2) \frac{\sin(n\omega t_c)}{\cos(\omega t_c/2)} \ , \end{aligned} \tag{2.62}$$

where the last equation is true because of $\sin(\omega t_c) = 2\sin(\omega t_c/2)\cos(\omega t_c/2)$. This is the form usually seen in the literature. The spectrum of the BOC signal is purely imaginary[9]

[9] One could of course shift the chip shape to the left or to the right which would result in a global phase factor. For the discussion to follow this global factor is irrelevant.

and an odd function of ω. In figure (2.23) the imaginary part of the spectrum is shown. The power spectrum, $P_{\text{BOC}}(\omega)$ for the BOC wave-form is obtained, as usual through

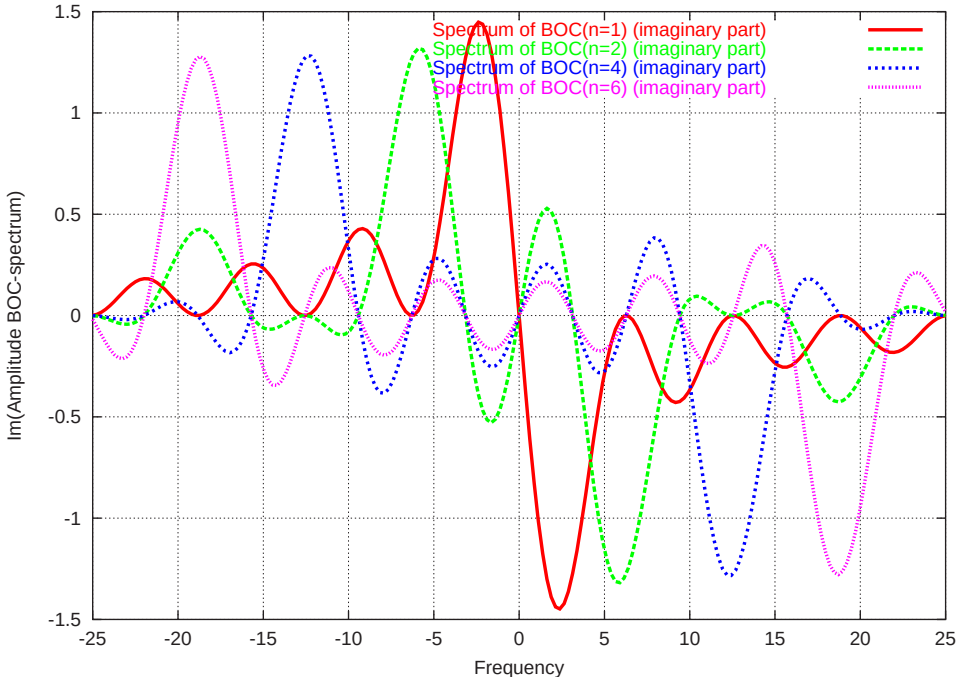

Figure 2.23.: Imaginary part of the BOC spectrum for $n = 2, 4$ and 6. The real part is zero (see eq. (2.62))

$$\begin{aligned} P_{\text{BOC}}(\omega) &:= |S_{\text{BOC}}(\omega)|^2 = S^*_{\text{BOC}}(\omega) S_{\text{BOC}}(\omega) \\ &= \frac{4}{\omega^2} \tan^2(\omega t_c/2) \sin^2(n\omega t_c) \,. \end{aligned} \tag{2.63}$$

The power spectrum (logarithmic scale) is shown in figure (2.24)

To obtain the auto-correlation function from the spectrum it is most convenient to start with the last equation in eq. (2.61):

$$\begin{aligned} P_{\text{BOC}}(\omega) &= \frac{16 \sin^4(\omega t_c/2)}{\omega^2} \left(\sum_{l=0}^{n-1} \mathrm{e}^{i2l\omega t_c} \right) \left(\sum_{k=0}^{n-1} \mathrm{e}^{-i2k\omega t_c} \right) \\ &= \frac{16 \sin^4(\omega t_c/2)}{\omega^2} \left[\sum_{k=-n+1}^{n-1} (n - |k|) \, \mathrm{e}^{i2k\omega t_c} \right] , \end{aligned} \tag{2.64}$$

where eq. (B.1) was used. The trick is to write $\sin^2(\omega t_c/2)$ in complex form and multiply it with the sum. The inverse Fourier transform of the remaining $\text{sinc}^2(\omega t_c/2)$ is then a triangular function. We then end up with a squared sinc-function multiplied with a sum

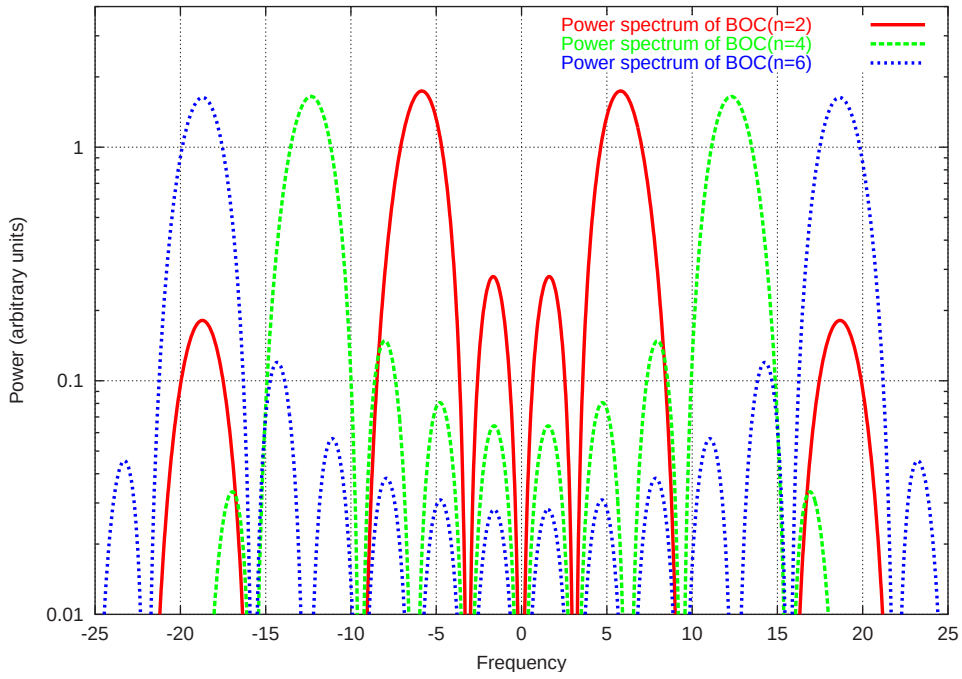

Figure 2.24.: The power spectrum of the BOC signal for $n = 2, 4$ and 6.

of complex phases. Transforming back into time domain, we use the convolution theorem and the result is a convolution of the triangle function with a sum of delta-distributions:

$$\begin{aligned} P_{\mathrm{BOC}}(\omega) &= 2t_c \,\mathrm{sinc}^2(\omega t_c/2)\left[1 - (\mathrm{e}^{i\omega t_c} + \mathrm{e}^{i\omega t_c})/2\right] \\ &\times \left[\sum_{k=-n+1}^{n-1} (n - |k|)\,\mathrm{e}^{i2k\omega t_c}\right] \\ &= 2t_c \,\mathrm{sinc}^2(\omega t_c/2) \\ &\times \sum_{k=-n+1}^{n-1} (n - |k|)\left[\mathrm{e}^{i2k\omega t_c} - (\mathrm{e}^{(2k+1)\omega t_c} + \mathrm{e}^{(2k-1)\omega t_c})/2\right] \end{aligned} \tag{2.65}$$

The inverse Fourier transform is then simply given by

$$\begin{aligned} ACF_{\text{BOC}}(t) &= \mathcal{F}^{-1}\left\{P_{\text{BOC}}(\omega)\right\} \\ &= \text{Tri}(t/t_c) \\ &\star \sum_{k=-n+1}^{n-1} (n-|k|)\left[2\delta(t-2kt_c) - \delta[t-(2k+1)t_c] - \delta[t-(2k-1)t_c]\right] \\ &= \sum_{k=-n+1}^{n-1} (n-|k|)\Big[2\,\text{Tri}(t/t_c-2k) \\ &\quad - \text{Tri}(t/t_c-2k-1) - \text{Tri}(t/t_c-2k+1)\Big] \end{aligned} \tag{2.66}$$

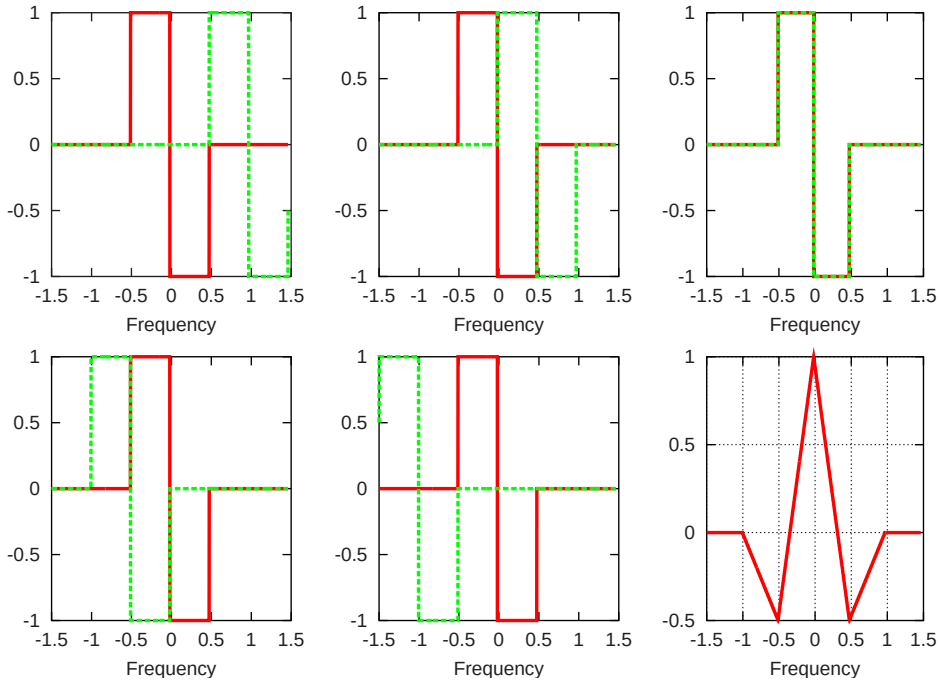

Figure 2.25.: A 'movie' of two correlating Manchester bits. The bottom right plot is the resulting ACF. (See text for explanations)

It is also possible to obtain the ACF directly by inspection. To do so we simply imagine shifting two copies of the ACF in time domain over each other. Let's first consider two Manchester coded bits that we shift into each other. In figure (2.25) this process is demonstrated. To completely construct the ACF we notice two things:

- We are integrating a function which is piece-wise constant and thus the result must be a piece-wise linear function

- As we shift the curves over each other we see that the extrema of the ACF are located where discontinuities of the two copies meet

Thus, the only thing we need to do is locate and calculate the extrema. The snapshots in figure (2.25) are exactly the extrema of the ACF. With that information we can construct the ACF using (figure (2.25))

The first plot in figure (2.25) shows the two Manchester encoded bits just before they overlap. At that moment the auto-correlation function is zero. The second plot shows the two bits as the overlap is exactly $-1/2$ and the second discontinuities meet. The third plot shows a perfect overlap of the two bits. The value of the auto-correlation function here is 1. The fourth and the fifth plot correspond to the second and first plot, respectively, as the auto-correlation function is symmetric.

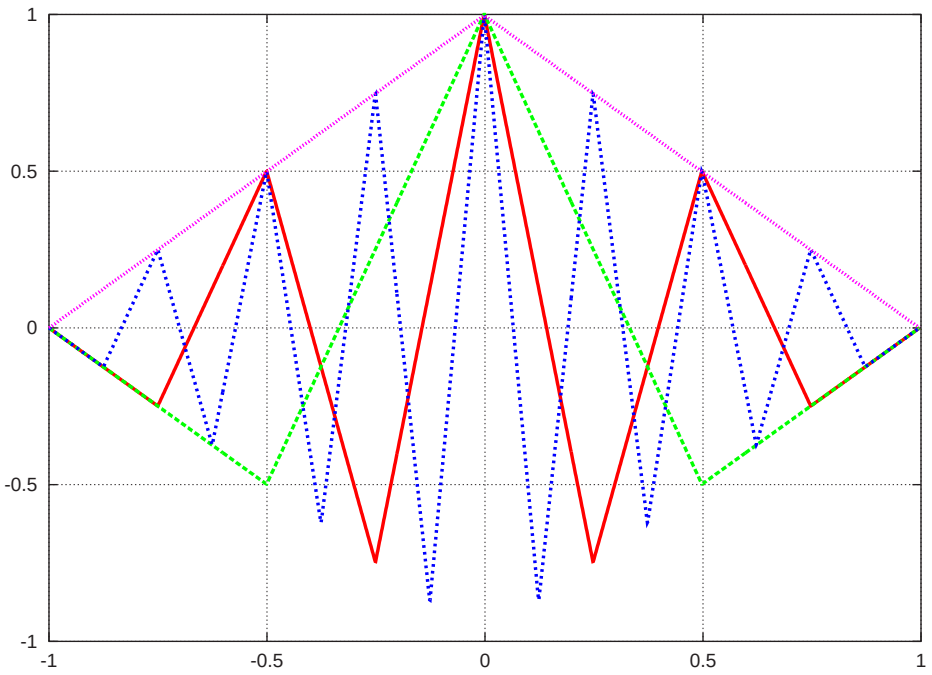

Figure 2.26.: The ACF of the BOC signal as a function of delay. The curves are plotted for $n = 1$, 2 and 4. $n = 2$ corresponds to the GPS M-code. For reference the ACF of the rectangular signal is shown.

For the general case of n we obtain analytical values for the extrema and simply say that they are connected with straight lines. The extrema of the ACF in the general case are given by:

$$R_k := R(\tfrac{|k|}{2n}) = (-1)^k(1 - \tfrac{k}{2n}) \quad \text{for} \quad k = 0, \ldots, 2n \tag{2.67}$$

and they are located at $x = \pm k/2n$, i.e. every time the modulating code flips, a new extreme has been reached. The units are in chip lengths and the amplitude of the bits is normalized to ± 1 so that the maximum of the ACF will be 1.

In figure (2.26) the ACF was plotted for infinite bandwidth for the cases $n = 1$, 2 and 4 using eq. (2.67).

Band-limited BOC signals It is relatively easy to obtain a band-limited version of the auto-correlation function of the BOC signal. Looking at eq. (2.65) we see that the factor in front of the sum is the inverse Fourier transform of the $\mathrm{Tri}(\cdot)$ function, which in turn happens to be the auto-correlation function of the rectangular chip in infinite bandwidth. In section 2.2.5 the band-limited version of that signal was derived. The eq. (2.66) was obtained using the convolution theorem, thus we simply have to replace the $\mathrm{Tri}(\cdot)$ in the first equation in eq. (2.66) with the corresponding expression for the band-limited version of $\mathrm{Tri}(\cdot)$. Thus we have

$$\begin{aligned} ACF^{\mathrm{BL}}_{\mathrm{BOC}}(t) = \sum_{k=-n+1}^{n-1} (n - |k|)\big[2R_{\mathrm{BL}}(t/t_c - 2k) \\ - R_{\mathrm{BL}}(t/t_c - 2k - 1) - R_{\mathrm{BL}}(t/t_c - 2k + 1)\big] \end{aligned} \tag{2.68}$$

where R_{BL} is given by eq. (2.31) and the bandwidth parameter b has the same interpretation as before. The band limiting parameter b defined in eq. (2.27) is normalized to the rate of the square wave, not the chip-rate. The parameter t_c refers to the period of the square-wave, but t_c replaces T_c in the equation above. Therefore the relation between the bandwidth limiting parameter b is:

$$B_{BW} = b\frac{4n}{T_c} \qquad \text{or} \qquad b = B_{BW}\frac{T_c}{4n} \tag{2.69}$$

where B_{BW} is expressed in Hz.

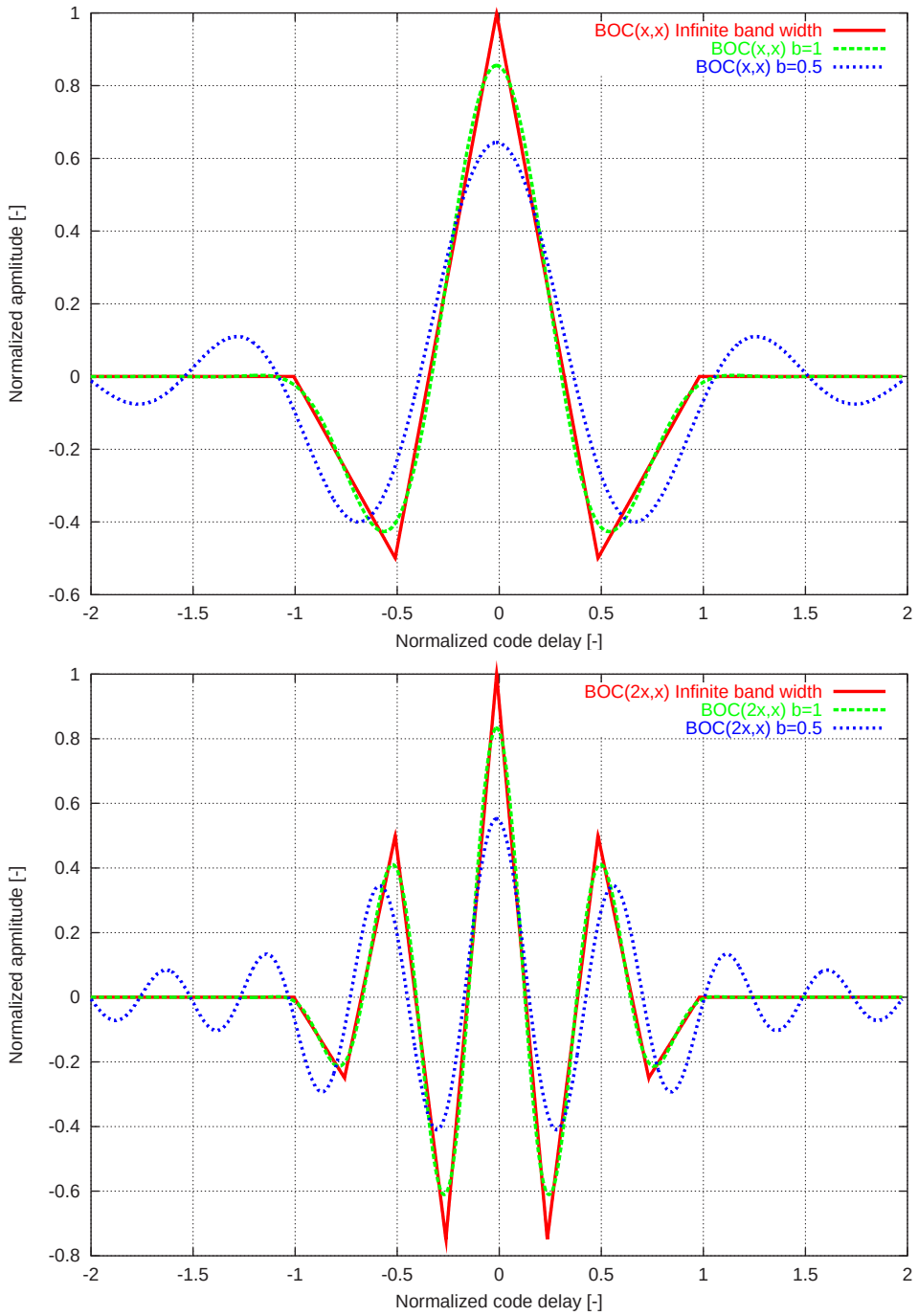

Figure 2.27.: Normalized auto correlation functions of the band-limited BOC signal. In the left plot $n = 1$, corresponding to Manchester encoding and the right plot has $n = 2$ corresponding to the GPS M-code signal structure. The values of the band limiting parameter were b = 0.5, 1 and ∞.

In figure (2.27) and figure (2.28) the auto correlation functions for band limited BOC

signals are shown. The left figure of figure (2.27) corresponds to a Manchester encoded signal and the right has the same structure as the GPS M-code. Note that although the ACF are termed as normalized the peak value is not equal to one. This reflects the fact that not all the power of the signal is available; some of the power is blocked by the band-limiting.

For b=1 the only effect of the band-limiting is the slight rounding off of the edges of the ACF. For a band-limiting of b=0.5, frequencies lower than the twice the square wave frequency are not let through by the filter. The result are the oscillations outside of the one chip length region. This would lead to undesirable side-lobe effects in case of multipath.

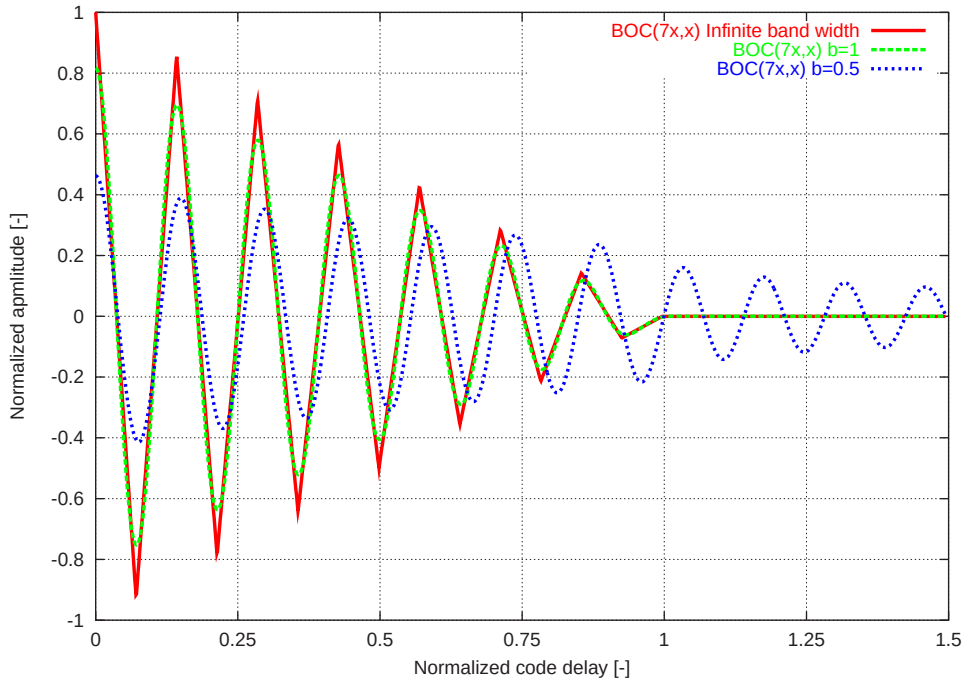

Figure 2.28.: Normalized auto correlation functions of the band-limited BOC signal for $n = 7$. This corresponds to a BOC(14,2). The values of the band limiting parameter were b = 0.5, 1 and ∞.

In figure (2.28) the band limited version of the BOC(7x,x) signal is shown. This is the same signal structure as the overlay on L1 (BOC(14,2)), which has been suggested for Galileo. For clarity, only the right side of the ACF is shown.

We also notice that with decreasing bandwidth, the side-lobes of the auto-correlation functions are shifted apart.

All the effects pointed out here coincide with the observations made in [Bet00b]. Except that here we have a closed form of the correlation process via eq. (2.68).

Correlation of the BOC signal with the Code For the SOC signal it was possible to correlate the signal with the code alone. This procedure resulted in a cross-correlation function with one maximum. The only difference between BOC and SOC is that the modulated sub-carrier is a square wave rather than a sine. Therefore one might think that a similar approach would be feasible for the BOC signal.

We proceed in the same manner as in the SOC case in eq. (2.55). The same arguments hold (the period of the square wave is the same as the modulating sine). We end up with an expression analog to the last equation in eq. (2.55) with the $\sin(\cdot)$ replaced by the square wave:

$$R(\tau, \Delta\omega_{\pm}, \phi) \approx \int C(u)C(u-\tau)\,\mathrm{Sqr}_n(u)e^{in\omega_{T_c}u}du \int_{-T_p/2}^{T_p/2} \mathrm{e}^{i\Delta\omega_{\pm}t + i\Delta\phi}\,dt \tag{2.70}$$

The cross correlation function for $\Delta\omega_{\pm} = 0$ and $\phi = 0$ is now given by the following integral:

$$R(\tau, n) = \int \mathrm{e}^{in\omega_{T_c}u}\,C(u-\tau)C(u)\,\mathrm{Sqr}_n(u)du = \int_{\Omega_\tau} \mathrm{e}^{in\omega_{T_c}u}\,\mathrm{Sqr}_n(u)du \tag{2.71}$$

where Ω_τ was defined as:

$$\Omega_\tau := \begin{cases} [\tau - T_c/2,\ T_c/2] & \text{for} \quad 0 \le \tau < T_c \\ [-T_c/2,\ \tau + T_c/2] & \text{for} \quad -T_c < \tau \le 0 \end{cases} \tag{2.72}$$

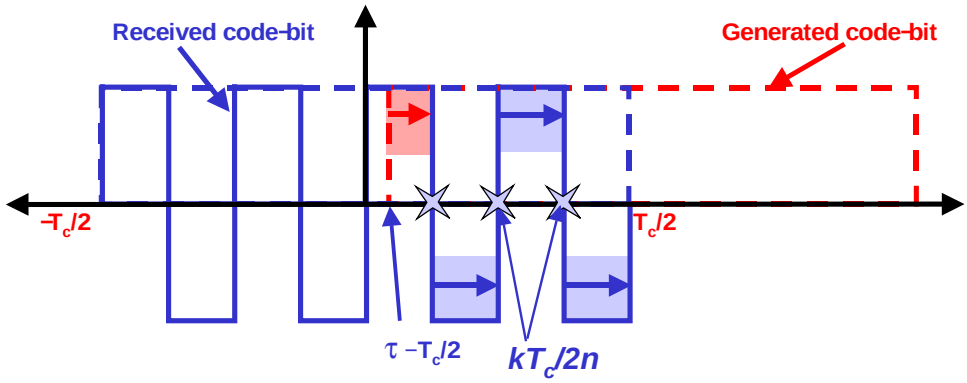

Figure 2.29.: Illustrating the integration in eq. (2.71).

When we were dealing with the SOC signal we had $\sin(\cdot)$ in the integral and the sine could easily be integrated over Ω_τ. Here on the other hand we have this square wave, which effectively divides the region Ω_τ into intervals with alternating signs. The situation for positive τ is sketched in figure (2.29). In order to simplify the following discussion, we would like to define the set of integers corresponding to the stars in figure (2.29):

$$\Omega_\tau^{int} := \{k | k \in \mathbb{Z} \wedge \frac{2n}{T_c}(\tau - T_c/2) \le k < n\} \tag{2.73}$$

Note that $k = n$ is *not* a member of the set. With these definitions we can easily calculate the integral in eq. (2.71):

$$\begin{aligned} R^*(\tau, n) &= \int_{\Omega_\tau} \mathrm{e}^{in\omega_{T_c} t}\, \mathrm{Sqr}_n(t) dt \\ &= \sum_{k \in \Omega_\tau^{\text{int}}} (-1)^k \int_{\frac{kT_c}{2n}}^{\frac{(k+1)T_c}{2n}} \mathrm{e}^{in\omega_{T_c} t}\, dt + \int_{\tau - T_c/2}^{k_{\min}} (-1)^{k_{\min}-1}\, \mathrm{e}^{in\omega_{T_c} t}\, dt \\ &= \frac{-i}{n\omega_{T_c}} \Bigg\{ \sum_{k \in \Omega_\tau^{\text{int}}} (-1)^k \left[\mathrm{e}^{in\omega_{T_c} \frac{(k+1)T_c}{2n}} - \mathrm{e}^{in\omega_{T_c} \frac{kT_c}{2n}} \right] \\ &\qquad + (-1)^{k_{\min}-1} \left[\mathrm{e}^{in\omega_{T_c} \frac{k_{\min} T_c}{2n}} - \mathrm{e}^{in\omega_{T_c}(\tau - T_c/2)} \right] \Bigg\} \end{aligned} \tag{2.74}$$

The coloring of the terms in the second equation refers to the colors in figure (2.29) and shows the integration intervals. The terms in the sum are constant in τ, except when the set Ω_τ^{int} changes. The variable $k_{\min}$ is defined as $k_{\min} := \min\{\Omega_\tau^{\min}\}$. The equation above is only valid for positive τ, but if we look at the first integral, we see that

$$\begin{aligned} R^*(-\tau, n) &= \int \mathrm{e}^{in\omega_{T_c} t}\, \mathrm{Sqr}_n(t) C(t+\tau) C(t) dt \\ &= -\int \mathrm{e}^{-in\omega_{T_c} t'}\, \mathrm{Sqr}_n(-t') C(-t'+\tau) C(-t') dt' \\ &= \int \mathrm{e}^{-in\omega_{T_c} t'}\, \mathrm{Sqr}_n(t') C(t'-\tau) C(t') dt' = \overline{R^*(\tau, n\omega_{T_c})} \end{aligned} \tag{2.75}$$

where the bar over R means complex conjugation. In the second equation the substitution $t \to -t'$ was made and it was used that the square wave is an odd function and that $C(t)$ is an even function.

Inserting the value for $\omega_{T_c} = \frac{2\pi}{T_c}$ in eq. (2.74) and using the fact that $\mathrm{e}^{i\pi} = -1$ we get:

$$\begin{aligned} R^*(\tau, n) &= \frac{-iT_c}{n2\pi} \left\{ \left(\sum_{k \in \Omega_\tau^{\text{int}}} -2 \right) + (-1)^{k_{\min}-1} \left[(-1)^{k_{\min}} - \mathrm{e}^{in2\pi(\tau/T_c - 1/2)} \right] \right\} \\ &= \frac{-iT_c}{n2\pi} \left\{ \left\lfloor \frac{2n(\tau - T_c)}{T_c} \right\rfloor + (-1)^{k_{\min}-1} \left[(-1)^{k_{\min}} - \mathrm{e}^{in2\pi(\tau/T_c - 1/2)} \right] \right\} \end{aligned} \tag{2.76}$$

The sum over Ω_τ^{int} basically counts the number of transitions of the square wave within the integration region (corresponds to the number of stars in figure (2.29)). This is formulated in the second equation of eq. (2.76) where $\lfloor \cdot \rfloor$ represents the integral part of the argument[10].

[10] I.e. $\lfloor 4.9 \rfloor = 4$ and $\lfloor 4.1 \rfloor = 4$

As usually the correlation function in time and frequency becomes:

$$R^*(\tau, \Delta\omega_\pm) = \frac{T_p T_c}{n2\pi}\left\{\left\lfloor\frac{2n(\tau - T_c)}{T_c}\right\rfloor + (-1)^{k_{\min}-1}\left[(-1)^{k_{\min}} - \mathrm{e}^{in2\pi(\tau/T_c - 1/2)}\right]\right\} \times \mathrm{e}^{i\Delta\phi}\,\mathrm{sinc}\left[\Delta\omega_\pm T_p/2\right] \tag{2.77}$$

under the assumption that the carrier frequency offset $\Delta\omega_\pm$ and the phase error ϕ does not vary considerably during the integration interval T_p.

Now we can look at coherent and non-coherent versions of eq. (2.76) by taking the real and imaginary part and the squared amplitude of the cross correlation function.

Taking the real part of eq. (2.76) we obtain the correlation function corresponding to the in-phase channel (I-channel) of the receiver:

$$\begin{aligned} R^{\mathrm{BOC}}_{\mathrm{c,I}}(\tau, \Delta\omega_\pm, n) &= \mathcal{R}\{R^*(\tau, \Delta\omega_\pm, n)\} \\ &= \frac{T_p T_c}{n2\pi}\left\{\left\lfloor\frac{2n(\tau - T_c)}{T_c}\right\rfloor \right. \\ &\qquad \left. + (-1)^{k_{\min}-1}\left[(-1)^{k_{\min}} - \cos(n2\pi(\tau/T_c - 1/2))\right]\right\} \\ &\qquad \times \cos(\Delta\phi)\,\mathrm{sinc}\left[\Delta\omega_\pm T_p/2\right] \end{aligned} \tag{2.78}$$

This is plotted in figure (2.30). Comparing to figure (2.18) we see that the surfaces are very similar as one would expect. Intuitively the influence from the "other" side-band should be minimal; it manifests itself in the wavy structure seen in the curves. Further, the only difference is that a sine is used for the modulation instead of a square wave. Therefore it is plausible that there is very little difference between the two.

Usually, the difference between the I- and the Q-channels is only the phase factor $\mathrm{e}^{\Delta\phi}$, but in eq. (2.78) the correlation function has an imaginary part. Thus, for zero phase error ($\Delta\phi = 0$) the I-channel is not zero and is given by

$$\begin{aligned} R^{\mathrm{BOC}}_{\mathrm{c,Q}}(\tau, \Delta\omega_\pm, n) &= \mathcal{I}\{R^*(\tau, \Delta\omega_\pm, n)\} \\ &= -\sin\left(n2\pi(\tau/T_c - 1/2)\right)\sin(\Delta\phi)\,\mathrm{sinc}\left(\Delta\omega_\pm T_p/2\right) \end{aligned} \tag{2.79}$$

In figure (2.32) the CF is plotted for four values of n. The similarity with figure (2.20) is evident. The maximum in frequency for both correlation functions (BOC and SOC) is $\Delta\omega_\pm = 0$, which means that the carrier frequency in the receiver will be $\omega' = \omega \pm n\omega_{T_c}$. This is a result of the frequency of the modulating wave.

One might think that the point on the correlation function where the slope is zero would cause problems when tracking the signal, because the correlator spacing can be chosen such that the slope of the S-curve at the tracking point is zero. This is exactly what

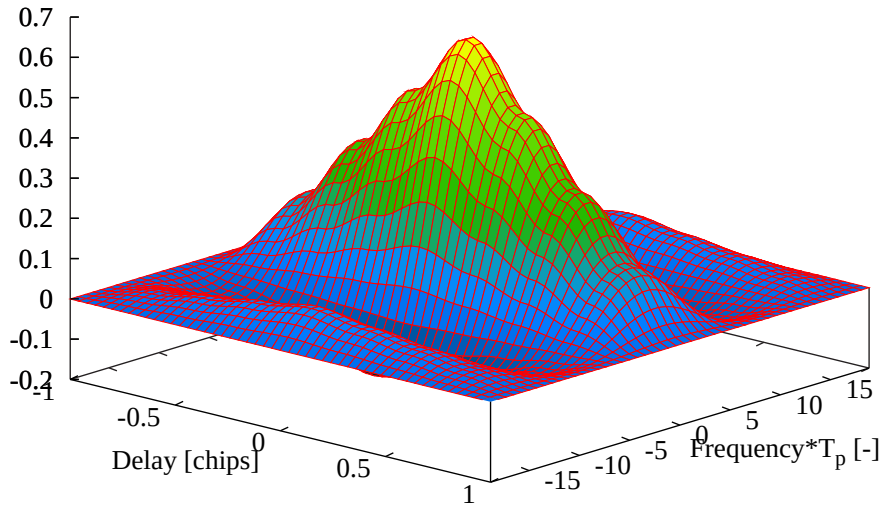

Figure 2.30.: 3-D plot of the coherent CF for a BOC signal, correlated with a reference signal consisting of the code alone for $n = 2$. The delay is measured in chip durations, while the frequency offset is measured in pre-detection bandwidths. The plot is based on eq. (2.77).

happens if a correlator spacing of T_c is chosen, which is the standard value for GPS C/A code receivers. By narrowing the correlator spacing such that the slope is maximized, a larger slope at the tracking point can be achieved than one would have obtained if there were only one side-band (i.e. only a normal rectangular chip were transmitted by the satellite).

In figure (2.33) this is shown. We see that for a correlator spacing of $2T_c/3$ the slope at the tracking point is increased, and in fact larger than for a rectangular chip shape. Provided the correlator spacing is chosen sensibly it is still theoretically possible that the tracking loop be “caught” on one of the plateaus. This, however, is not a stable state as the magnitude of the S-curve at that point is large and it is the magnitude of the S-curve that drives the tracking loops at this point.

2.2.9. Relation Between the SOC and BOC Signals

In the previous section it was implied that there is a relationship between the BOC and the SOC signal. To see exactly how they are related we expand the square wave in a Fourier series:

$$\mathrm{Sqr}_n(t) = \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{1}{2k-1} \sin[(2k-1)n\omega_{T_c} t] \tag{2.80}$$

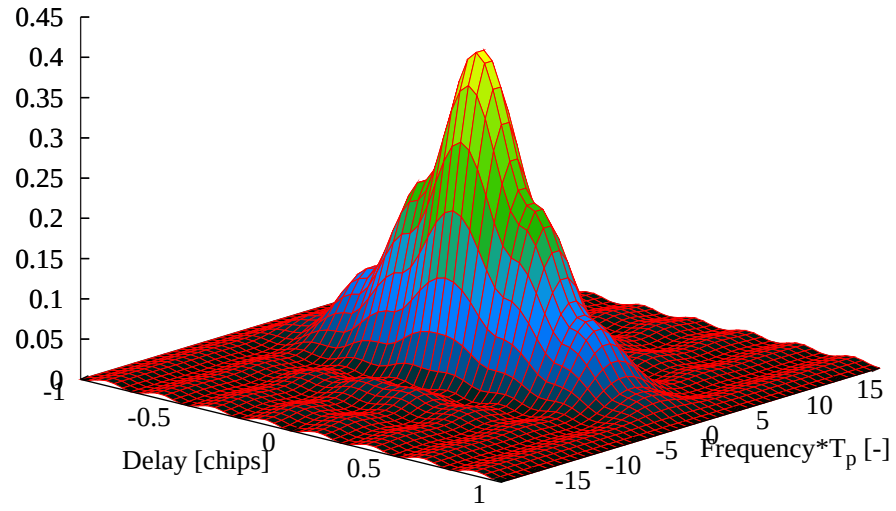

Figure 2.31.: 3-D plot of the coherent CF for a BOC signal, correlated with a reference signal consisting of the code alone for $n = 2$. The delay is measured in chip durations, while the frequency offset is measured in pre-detection bandwidths. The plot is based on eq. (2.77).

The first term in the expansion is proportional to $\sin(n\omega_{T_c}t)$, which is the sine in the SOC signal. We recall that both the SOC and BOC signals were normalized so that the power of each bit is 1[arbitrary units]. The power due to the first term in the expansion above is thus $\frac{(4/\pi)^2}{2} \approx 81\%$

In this sense the SOC signal can be viewed as a first order BOC signal. However, the SOC signal is *not* a band-limited version of the BOC signal; the bandwidth of the modulating carrier of the SOC signal is band limited (i.e. the sine wave), but the rectangular chip has an infinite bandwidth. So band-limiting a BOC signal will not result in a SOC signal.

Looking at eq. (2.80) we see that the square wave is a series of sine waves. These in turn are delta distributions in frequency domain. The spectrum of the rectangular pulse is a $\mathrm{sinc}(\cdot)$ function. So when the convolution of the two is formed (i.e. the $\mathrm{sinc}(\cdot)$ and the series of delta distributions), we have contributions from the higher order $\mathrm{sinc}(\cdot)$ functions *within* the frequency band of interest (i.e. close to the carrier frequency). Thus *each* of the higher order terms will contribute to the spectrum at the carrier frequency.

If the bandwidth limited versions of the BOC-signals in figure (2.27) and figure (2.28) are compared to the SOC-signals in figure (2.17) we see that the SOC signal has zero derivative at T_c and the bandwidth limited BOC doesn't. Outside of $[-T_c, T_c]$ the auto-correlation function of the SOC-signal is zero, whereas for the bandwidth limited BOC

signal there are side-lobes outside of this region.

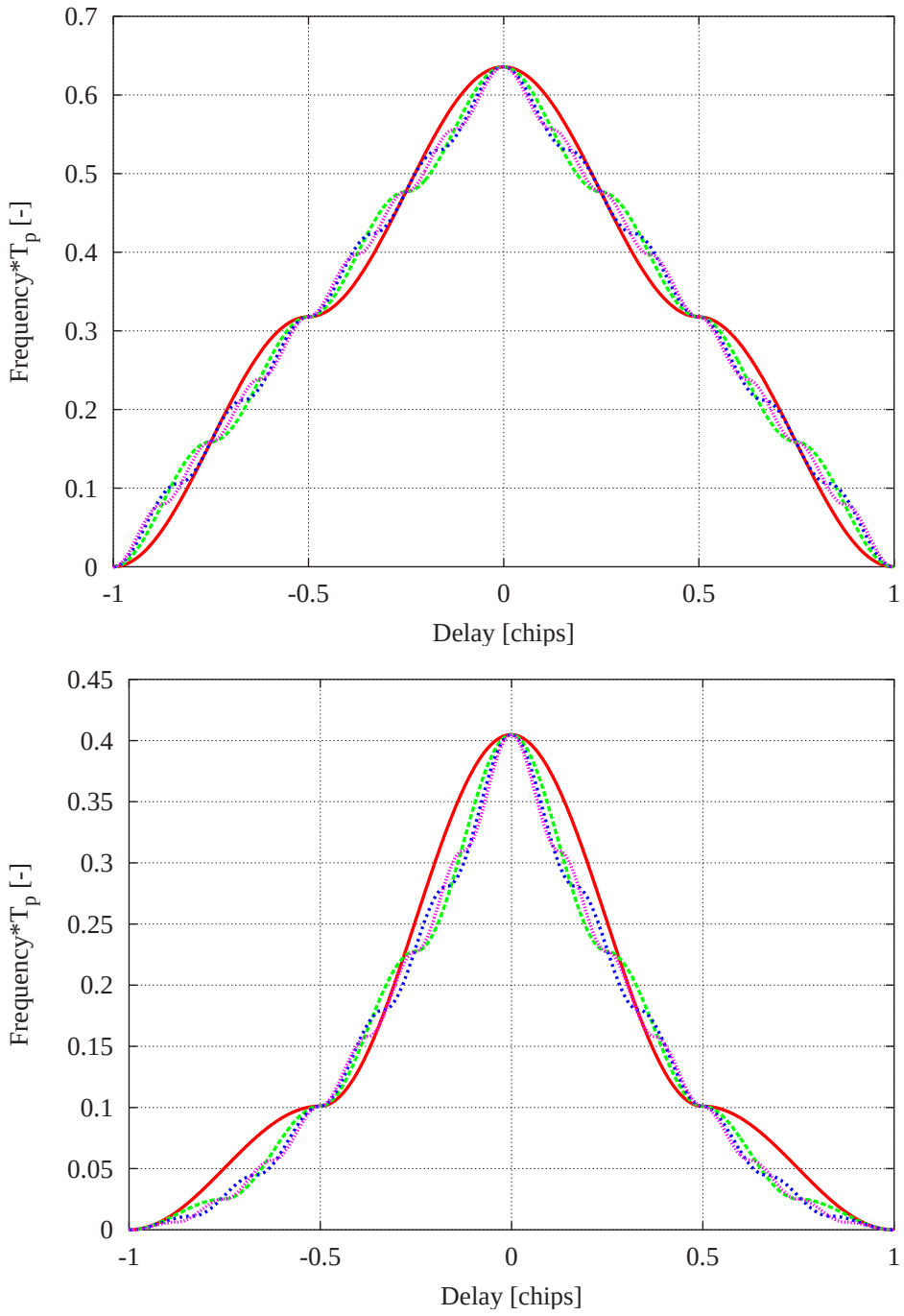

Figure 2.32.: The coherent CF on the left and the non-coherent CF on the right between the code and the BOC signal as a function of τ.

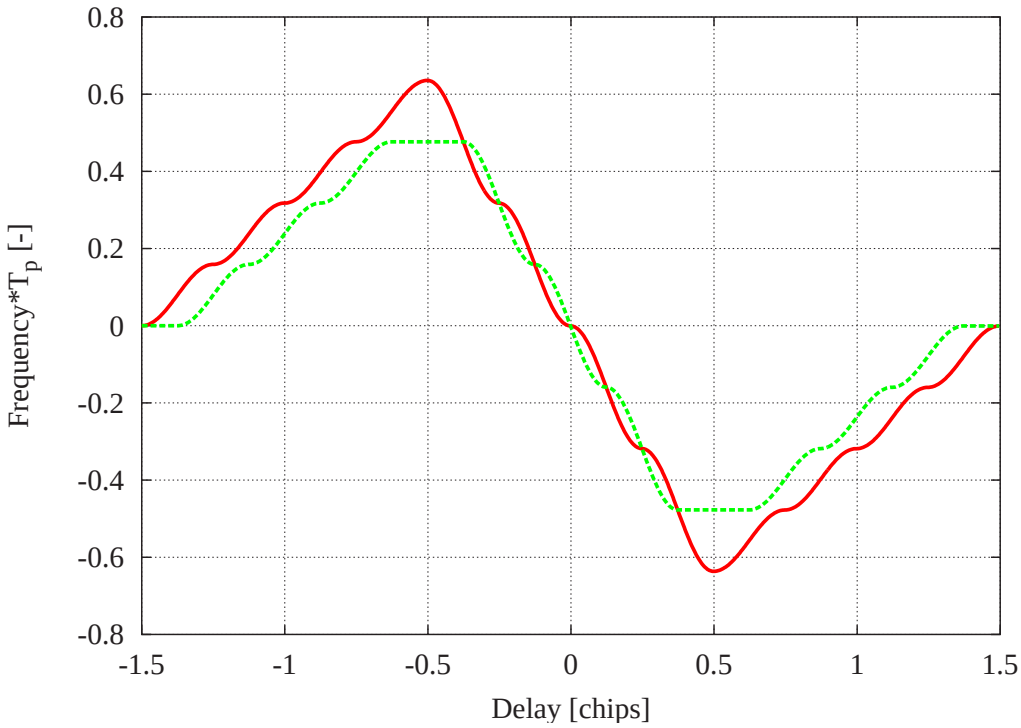

Figure 2.33.: The S-curve of the coherent correlation function for n=2. Correlator spacing was chosen T_c and $2T_c/3$. Note the slope at the tracking point ($\tau = 0$).

2.3. Modeling Signal Propagation

In this section a model for the propagation of the signal is developed. The modelling refers to the signal propagation from the satellite to the receiver taking reflections from physical objects into account. No dispersive effects are considered (ionosphere) and the medium is assumed to be vacuum (no additional delays or signal attenuation caused by the troposphere.)

The main requirement for this model is that it must yield a set of parameters, suitable to describe the signal in terms of the signal model in the receiver. Effectively, this means that the output of the signal propagation model must consist of

- Signal-to-noise ratio
- A set of geometric delays (on code and phase), representing the individual multipaths
- A set of damping parameter describing the relative signal-strength among the multipaths

All these parameters are functions of time.

To achieve this, a detailed physical model of the signal propagation must be analyzed and implemented, including a power budget model, geometric ray-tracing and an electromagnetic analysis of the signal paths.

2.3.1. Power Budget of the Propagation Channel

For reference the so-called communications equation (see for example [LS73]) will be written down and explained. Given a transmitter, emitting a signal with total power of P_t W, the signal-to-noise ratio (SNR) can be expressed by:

$$\frac{P_{rec}}{N_0} = P_t G_t \left(\frac{\lambda}{4\pi d}\right)^2 L_\theta L_p G_r \frac{1}{kT} \tag{2.81}$$

where G_t and G_r are the antenna gains of the transmitting and receiving antenna, respectively. The distance between receiver and transmitter is denoted by d and λ is the wave length of the signal. The factor in the parenthesis is called the space loss. Its definition is connected to the definition of the antenna gain parameters G_t and G_r. The antenna gain is defined by

$$G_i := \frac{4\pi A_i}{\lambda^2} \tag{2.82}$$

where the subscript i is either r or t and the antenna capture area is defined as A_i. Thus an antenna with gain 1 has a capture area of $A_i = \lambda^2/4\pi$. This reflects the physical fact

that the capture area of a particular type of antenna, say a dipole antenna, scales with λ^2. The power available at the receiving antenna emitted from an isotropic antenna is equal to the ratio between the capture area of the receiving antenna and a sphere with radius d times the power emitted:

$$\boxed{P_r = \frac{A_r}{4\pi d^2} P_{to} = G_r \left(\frac{\lambda}{4\pi d}\right)^2 P_{to} = G_r G_t \left(\frac{\lambda}{4\pi d}\right)^2 P_t} \tag{2.83}$$

The factor kT in eq. (2.81) is Boltzmann's constant, $k = 1.38 \cdot 10^{-23}$ J/°K times the noise temperature. The power, $P_{t_0} = G_t P_t$, is called the equivalent isotropic radiated power (EIRP) and corresponds to the power an isotropic antenna at the satellite would have to emit if the power P_{t_0} were to be emitted in all directions. This is of course not the case and the antennas on the satellites are designed such that the main part of the power is concentrated in a beam that covers the earth. Thus the gain, G_t depends on the orientation of the antenna.

The simulator will use eq. (2.81) to simulate the signal-to-noise ratio at the receiver.

2.3.2. Geometric Ray-Tracing

In this section the ideas for the ray-tracing in SNSS are discussed. In SNSS we want a general, exact 3-D ray-tracing algorithm.

Basic operations and tests needed for ray-tracing In the following a face means a piece of a plane which is defined by 3 or more points in 3-D. The face is also oriented and its orientation is defined by a normal vector pointing in the (so defined) positive direction. Further the face is required to be flat, i.e. all the vertexes of the face must lie in the same plane.

Definition 2.3.2.1. A face is a set of N vectors, which all lie in the same plane. Let $\vec{A}_i$, $i = 0, \ldots, N-1$ be vectors in the plane defining the vertexes of the face and let $\vec{q}_i := \vec{A}_i - \vec{A}_{i-1}$, ($i$ is cyclic in N). The normal vector of the face is

$$\hat{n} := \frac{\vec{q}_i \times \vec{q}_{i+1}}{|\vec{q}_i||\vec{q}_{i+1}|} \tag{2.84}$$

for all i.

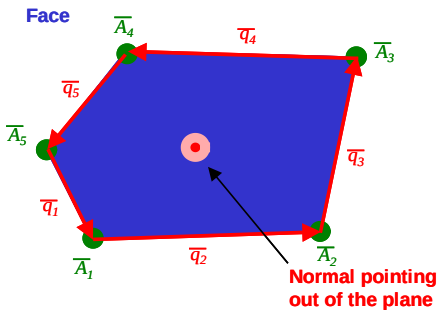

Figure 2.34.: Definition of a face

It is a consequence of the assumption that the vertexes of the face are all in the same plane that the equation above holds for all i. The region connecting the vectors $\vec{A}_i$ must be convex. The definition is illustrated in figure (2.34).

The algorithm is formulated in a coordinate-system independent way. The main ingredients for the ray-tracing engine are:

- Operations
  - Project a point onto a plane
  - Mirror a point in 3-D about a plane
  - Intersect a ray with a plane
- Tests
  - Determine if a point lies on a face
  - Determine if planes are facing each other
  - Determine if a point is in the positive half-space of a plane

Projecting a point onto a plane Let the plane P be defined by two non-parallel vectors $\vec{p}$ and $\vec{q}$ and a point $\vec{R}$ lying in that plane. The vectors $\vec{p}$ and $\vec{q}$ need not to be normalized or perpendicular. Consider a point $\vec{Q}$ that we want to project onto P. A vector in the plane can be expressed as a linear combination of $\vec{p}$ and $\vec{q}$:

$$\vec{R}_p := \vec{R} + t\vec{q} + s\vec{p}. \tag{2.85}$$

For $\vec{R}_p$ to be the projection point of $\vec{Q}$ onto P, We must have:

$$(\vec{Q} - \vec{R}_p) \cdot \vec{q} = 0 \quad \text{and} \quad (\vec{Q} - \vec{R}_p) \cdot \vec{p} = 0 \tag{2.86}$$

that is, the vector $(\vec{Q} - \vec{R}_p)$ must be perpendicular to both $\vec{q}$ and $\vec{p}$. These two equations can be solved for s and t:

$$t = \frac{(\vec{r} \cdot \vec{p})(\vec{p} \cdot \vec{q}) - p^2(\vec{r} \cdot \vec{q})}{(\vec{p} \cdot \vec{q})^2 - p^2 q^2}\,, \tag{2.87}$$

and

$$s = \frac{(\vec{r} \cdot \vec{q})(\vec{p} \cdot \vec{q}) - q^2(\vec{r} \cdot \vec{p})}{(\vec{p} \cdot \vec{q})^2 - p^2 q^2}\,, \tag{2.88}$$

where $\vec{r} := \vec{R} - \vec{Q}$. If $\vec{q}$ and $\vec{p}$ in eq. (2.87) and eq. (2.88) are perpendicular, s and t can be written as:

$$t = \frac{(\vec{r} \cdot \vec{q})}{q^2} \quad \text{and} \quad s = \frac{(\vec{r} \cdot \vec{p})}{p^2} \tag{2.89}$$

The factors s and t can be calculated using relatively little processing power. There are no sines or cosines involved, only some multiplications and one division (the denominator is the same in t and s). In an object-oriented implementation, where the vectors are defined as vectors in the mathematical sense, this method is independent of the coordinate-system (and could in a straight-forward manner be extended to N dimensions, although that is of no value in this context).

Mirroring a point about a plane An important part of the ray-tracing (as it is done here) is to mirror a point $\vec{Q}$ about a plane. This can easily be achieved after we have calculated the projection point $\vec{R}_p$ (see eq. (2.85)). If we denote by $\vec{Q}_I$ the mirrored point, we have:

$$\begin{aligned} \vec{Q}_I &:= \vec{Q} - 2(\vec{Q} - \vec{R}_p) \\ &= 2(\vec{R} + t\vec{q} + s\vec{p}) - \vec{Q}\,. \end{aligned} \tag{2.90}$$

Intersection point of a ray with a plane The reflection point of a ray going from $\vec{R}$ to $\vec{Q}$, reflected by a plane P, is the same as the intersection point of a ray going from $\vec{R}$ to $\vec{Q}_I$ with the same plane P. Therefore we must be able to find intersection points of rays with planes.

The question is: given two points $\vec{A}$ and $\vec{B}$, where is the intersection point of the line defined by these two vectors with the plane P?

A point $\vec{r}$ lying in a plane containing $\vec{R}_0$ with normal $\vec{n}$ must satisfy

$$(\vec{R}_0 - \vec{r}) \cdot \vec{n} = 0 \tag{2.91}$$

A line that goes through $\vec{A}$ and $\vec{B}$ is given by:

$$\vec{S} := \vec{A} + t(\vec{B} - \vec{A}) \tag{2.92}$$

Inserting into eq. (2.92) into eq. (2.91) and solving for t, we get:

$$t = \frac{(\vec{R}_0 - \vec{A}) \cdot \vec{n}}{(\vec{B} - \vec{A}) \cdot \vec{n}} \tag{2.93}$$

If $0 < t < 1$ the line-segment between $\vec{A}$ and $\vec{B}$ intersects the plane.

Determine if a point lies on a face In the previous section we calculated the intersection point of a ray with a plane. Now we want to know if that point $\vec{R}$ is on a given face, or more generally we want to know if a point $\vec{R}$ in the plane is also on the face.

Let $\vec{A}_i$, $i = 0, \ldots, N-1$ be the vectors in the plane defining the face and define $\vec{q}_i := \vec{A}_i - \vec{A}_{i-1}$, ($i$ is cyclic in N). First we construct a vector

$$\vec{b}_i := \vec{q}_i - t\vec{q}_{i-1}\,, \tag{2.94}$$

where t has to be determined such that $\vec{b}_i$ is perpendicular to $\vec{q}_i$, i.e. $\vec{b}_i \cdot \vec{q}_i = 0$. That yields:

$$t = \frac{q_i^2}{(\vec{q}_i \cdot \vec{q}_{i-1})} \,. \tag{2.95}$$

Now, because the face is convex, we always have $t > 0$ and thus $\vec{b}_i$ points into the face. Thus if $(\vec{R} - \vec{A}_i) \cdot \vec{b}_i < 0$ then $\vec{R}$ does *not* lie on the face.

With this test it can only be determined if the point is *not* on the face. That means it must be continued until it is known that the point is not on the face or until *all* $\vec{b}_i$'s have been tested.

If $\vec{q}_i$ and $\vec{q}_{i-1}$ are perpendicular, we simply set $\vec{b}_i := \vec{q}_{i-1}$.

Determine if a point is in the positive half-space of a plane The implementation of this test is straight-forward:

$$(\vec{r} - \vec{R}_0) \cdot \vec{n} > 0 \tag{2.96}$$

If eq. (2.96) is fulfilled, $\vec{r}$ is in the positive half-space.

Determine if faces are facing each other This is also straight-forward: if

$$\vec{n}_1 \cdot \vec{n}_2 > 0 \tag{2.97}$$

then the faces are facing each other. However, this does not mean that the faces are opposite and facing each other. It could happen that they are facing each other, but are not opposite (i.e. they don't "see" each other, like two people standing back-to-back, they would be facing each other in the sense of eq. (2.97).).

2.3.3. Electromagnetic Ray-Tracing

In the previous section the geometry of the signal propagation was discussed, but in order to describe the navigation signal adequately and its interaction with the environment, the electromagnetic properties of the signal must be considered. In this section the classical electrodynamics of reflection, refraction and diffraction needed for this work are summarized. For a detailed analysis the reader is refered to the excellent literature cited in the following sections. In the following discussion the spread spectrum character of the GNSS signal is ignored and it is treated as a monochromatic electromagnetic plane wave with frequency equal to the carrier frequency of the signal. However, the polarization of the signal will be treated in a rigorous way.

2.3.3.1. Specular Reflection

The propagation of a plane electromagnetic wave can be described in terms of two perpendicular electric vectors of the form [Jac82] and [KC73]:

$$\vec{E} := a_1 \, \mathrm{e}^{i(\omega t+\delta_1)} \, \hat{\mathbf{x}} + a_2 \, \mathrm{e}^{i(\omega t+\delta_2)} \, \hat{\mathbf{y}} \tag{2.98}$$

where a_i and δ_i are real, $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are orthogonal and parallel unit vectors to the plane of incidence respectively (see figure (2.35)). Referring to the figure (2.35) the electric field vector of the incident and reflected wave can be written as:

$$\begin{aligned} \vec{E}^i &:= a_1^i \, \mathrm{e}^{i(\omega t+\delta_1^i)} \, \hat{\mathbf{x}}^i + a_2^i \, \mathrm{e}^{i(\omega t+\delta_2^i)} \, \hat{\mathbf{y}}^i \\ &\text{and} \\ \vec{E}^r &:= a_1^r \, \mathrm{e}^{i(\omega t+\delta_1^r)} \, \hat{\mathbf{x}}^r + a_2^r \, \mathrm{e}^{i(\omega t+\delta_2^r)} \, \hat{\mathbf{y}}^r \end{aligned} \tag{2.99}$$

The polarization of the signal is accounted for by the magnitudes of a_1 and a_2 and the

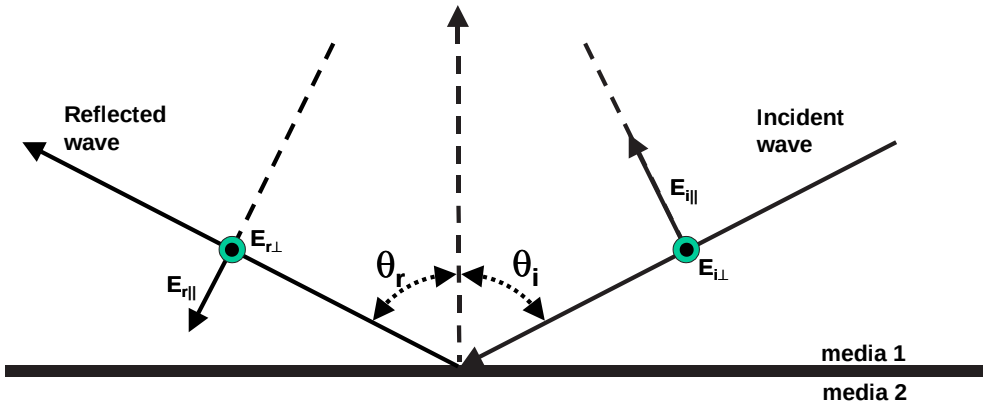

Figure 2.35.: Electric field components reflected off a more dense medium

phase shifts δ_1 and δ_2. For example when $a_1 = a_2$ and $\delta_1 = \delta_2$, then the wave is linearly polarized in the direction of $\hat{\mathbf{x}} + \hat{\mathbf{y}}$. When $a_1 = a_2$ and $\delta_1 = \delta_2 \pm \pi/2$, then the wave is left/right hand circularly polarized.

Fresnel Equations When a plane wave hits a boundary between two media (figure (2.35)), a part of the energy is reflected while the rest is refracted. In this work we are only concerned about the reflected part of the wave. The state of the wave after a reflection off a surface is completely described by the following two complex quantities:

$$\eta_1 := \frac{a_1^i \, \mathrm{e}^{i\delta_1^i}}{a_1^r \, \mathrm{e}^{i\delta_1^r}} \quad \eta_2 := \frac{a_2^i \, \mathrm{e}^{i\delta_2^i}}{a_2^r \, \mathrm{e}^{i\delta_2^r}} \tag{2.100}$$

It is clear that the η_i's will depend upon the properties of the medias. In particular the dielectric constant ϵ the magnetic permeability μ and the conductivity σ of the media

are of importance. By imposing the usual boundary conditions [KC73], [Jac82] on the electric and magnetic vector of the incident, reflected and refracted wave, the Fresnel equations are obtained:

$$\eta_1 = \frac{\cos\theta_i - \sqrt{Y^2 - \sin^2\theta_i}}{\cos\theta_i + \sqrt{Y^2 - \sin^2\theta_i}} \qquad \eta_2 = \frac{Y^2\cos\theta_i - \sqrt{Y^2 - \sin^2\theta_i}}{Y^2\cos\theta_i + \sqrt{Y^2 - \sin^2\theta_i}} \tag{2.101}$$

where θ_i is the angle of incidence (see figure (2.35)), ε_r is the relative dielectricity of the media, μ_r is the relative magnetic permeability, σ is the conductivity in [℧/m] and λ is the wave length. The normalized effective conductivity Y is defined as [BS87]:

$$Y := \sqrt{\frac{\varepsilon_r + i60\lambda\sigma}{\mu_r}} \tag{2.102}$$

For most substances the magnetic permeability can be assumed to be $\mu = 1$, i.e. the materials are assumed to be non-magnetic. Furthermore we note that the effective conductivity is a complex quantity and thus it affects not only the magnitude of the field, but also the phase.

2.3.3.2. Stokes Parameters

The polarization state of a propagating electromagnetic plane wave can be described with 4 parameters; the magnitude of both electric vectors as well as their phase. A suitable set of parameters are the so-called Stokes parameters. Defining the magnitude and phase of the electric vectors by:

$$E_1 := a_1\,\mathrm{e}^{i\delta_1} \quad \text{and} \quad E_2 := a_2\,\mathrm{e}^{i\delta_2} \tag{2.103}$$

then the four Stokes parameters are given by:

$$\begin{aligned} S_0 &= a_1^2 + a_2^2 \\ S_1 &= a_1^2 - a_2^2 \\ S_2 &= a_1 a_2 \cos(\delta_2 - \delta_1) \\ S_3 &= a_1 a_2 \sin(\delta_2 - \delta_1) \end{aligned} \tag{2.104}$$

With these definitions a linear polarized wave, polarized along the first axis is described by the parameters $\vec{S} = [1, 1, 0, 0]$. Similarly a right-hand circular polarized wave (negative helicity) has the stokes parameters $\vec{S} = [1, 0, 0, -1]$. The behavior of the antenna can also be described by the Stokes parameters. The coupling between the incident wave and the antenna is then given by [KC73]

$$S = \frac{1}{4}\vec{S}_{\text{antenna}} \cdot \vec{S}_{\text{signal}} \tag{2.105}$$

The parameters a_i and δ_i can be found from the Fresnel equation eq. (2.101).

All these things are relative to the orientation of the propagating vector of the wave. In the case of the antenna, the Stokes parameters are relative to some predefined orientation vector of the antenna.

2.3.4. Reflection of Rough Surfaces

The simplest case of reflections are the smooth reflections. Here Snell's law is valid and the reflection can easily be calculated using the Fresnel equations eq. (2.101) and the polarization is accounted for by the Stokes parameters. However, when the surfaces are not

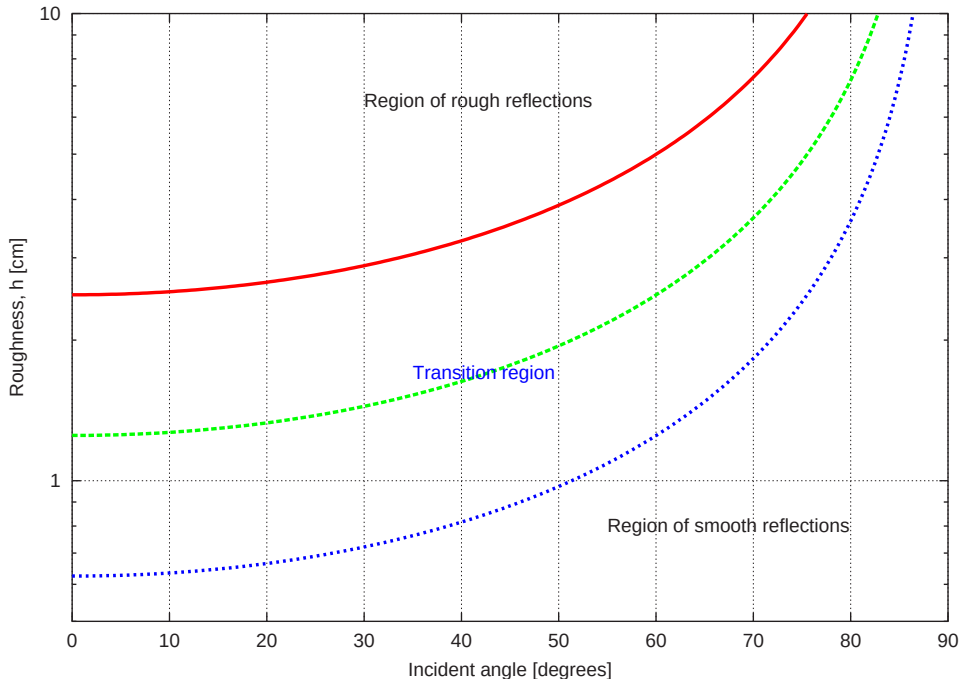

Figure 2.36.: The Rayleigh criterion for rough surfaces as a function incident angle for a wave length $\lambda = 20$ cm. On the y-axis the characteristic dimension of the roughness is shown. Indicated are the regions of rough/smooth reflections. The three curves correspond to $\Gamma = 8, 16$ and 32. See eq. (2.106) and discussion threafter for explanations.

smooth, then the analysis becomes increasingly more complex. In this work two cases of roughness will be considered. In the first case surfaces are considered that have a characteristic roughness comparable to the carrier wave length. The second case reflector with a roughness considerably larger than the carrier wave length are considered. Basically, these reflectors are composed of a number of smaller reflectors.

The classical measure of whether a surface is rough or not, is given by the Rayleigh criterion:

$$h \geq \frac{\lambda}{\Gamma \cos(\theta_1)}\,, \tag{2.106}$$

where λ is the wave length of the carrier, θ_1 the incident angle of the wave and h is a measure of the roughness of the reflector. The factor Γ is an arbitrary factor, which is conventionally chosen to be $\Gamma = 8$. This value corresponds to a phase difference of $\pi/2$ between a wave reflected at $\xi = 0$ and $\xi = h$. It turns out that this value is rather optimistic

(i.e. a surface is considered smooth, when it should in fact be considered rough [BS87]). More realistic values would be $\Gamma = 16$ or $\Gamma = 32$, corresponding to a phase difference of $\pi/4$ and $\pi/8$, respectively.

For L-band signals we have $\lambda \approx 20$ cm. The Rayleigh criterion is shown in figure (2.36).

As pointed out by various authors the Rayleigh criterion should not be considered a hard measure whether a surface is rough or not. It only gives the order of magnitude. A comparison with the specular scattering coefficient in the next section will show this (see figure (2.39)).

Looking at figure (2.36) we see that the region above the upper-most curve (red, solid one) can be considered as a rough surface and in the region below the lowest curve (the blue dotted one) the reflector can be considered smooth. Obviously, surfaces with a characteristic dimension of roughness more than $h = 2.5$ cm will *always*[11] qualify as rough for signals in L-band. On the other hand, it is reasonable to assume surfaces with a roughness less than $h = 0.62$ cm to be smooth.

Therefore most man-made surfaces (walls, roads etc.) are either in the transition region or can be considered smooth. For natural terrain, however, this does not necessarily hold.

In the following two sections models will be developed for the moderate roughness and the very rough case.

2.3.4.1. Roughness in the Order of the Carrier Wave Length

In this section a very brief summary of the scattering from rough, random surfaces is given. This re-cap is by no means meant to be complete and the reader is refered to the very excellent book by P. Beckmann and A. Spizzichino [BS87] for further reading. For the reader's convenience the corresponding pages and sections of this reference are sometimes given.

As the scattering theory of rough surfaces involves many subtle assumptions and approximations, it is useful to briefly outline the theory and thus point out the limitations of the theory in context.

The Kirchoff Solution The starting point for the well-known Kirchoff solution [BS87] is the Helmholtz equation:

$$E_r = \frac{1}{4\pi} \iint\limits_S \left(E \frac{\partial \psi}{\partial n} - \psi \frac{\partial E}{\partial n} \right) dS \tag{2.107}$$

with

$$\psi = \frac{\mathrm{e}^{ik_2 R'}}{R'} \tag{2.108}$$

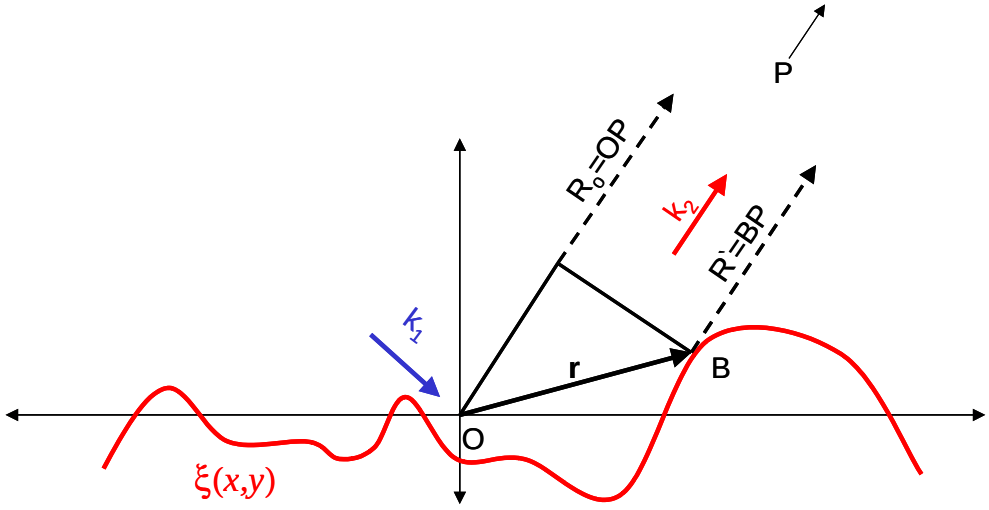

Figure 2.37.: Definition of the coordinate system used in eq. (2.109)

where E is the electric field, S is the surface described by the function $\xi(x, y)$, indicated in figure (2.37). The coordinate system has its origin at O and the point of observation is denoted by P. Points on the surface $\xi(x, y)$ are denoted by B. The parameter R' is defined as the distance between the point of observation P and a point on the surface B. The point of observation P is assumed to be „far away" from the reflecting surface, thus OP and BP can be assumed to be parallel. The wave vectors of the incident and reflected wave are $\vec{k}_1$ and $\vec{k}_2$, respectively. The position vector of a point on the surface, $\vec{r} = x\hat{e}_x + \xi(x)\hat{e}_z$ for a one-dimensional surface, where $\hat{e}_x$ and $\hat{e}_z$ are the basis vectors of the coordinate system indicated in figure (2.37).

The extension to a two-dimensional surface is straight forward and can be found in [BS87]. However, in order to keep the notation as simple as possible only the one-dimensional analysis is shown here. The corresponding expressions for two-dimensional surfaces will be written down in the end.

Inserting the electric fields and its surface derivatives and assuming the Fresnel equations eq. (2.101), we find[12]:

$$\begin{aligned} E_2 &= \frac{\mathrm{e}^{kR_0}}{4\pi R_0} \iint_S (R_\xi \vec{v} - \vec{p}) \cdot \hat{n}_\xi \, \mathrm{e}^{i\vec{v}\cdot\vec{r}_\xi} \, dS \\ &= \frac{\mathrm{e}^{kR_0}}{4\pi R_0} \int_{-L}^{L} (a\xi'(x) - b) \, \mathrm{e}^{iv_x x + iv_z \xi(x)} \, dx \end{aligned} \tag{2.109}$$

where the sub-script in the first equation indicates dependence on the shape of the surface. The factor R in eq. (2.109) is the reflection coefficient, obtained by the Fresnel equations

[11] i.e. for all incident angles
[12] This is done in [BS87] in section 3.1 on p. 20.

eq. (2.101). The vectors $\vec{v}$ and $\vec{p}$ are defined as

$$\vec{v} := \vec{k}_1 - \vec{k}_2 \quad \text{and} \quad \vec{p} := \vec{k}_1 + \vec{k}_2. \tag{2.110}$$

The normal vector $\hat{n}_\xi = \hat{n}_\xi(x, y)$ is the „local" normal vector of the surface $\xi(x, y)$. The factors a and b can be shown to be:

$$\begin{aligned} a &= (1 - R)\sin(\theta_1) + (1 + R)\sin(\theta_2) \\ b &= (1 + R)\cos(\theta_2) - (1 - R)\cos(\theta_1) \end{aligned} \tag{2.111}$$

where θ_1 and θ_2 are the incident and reflection angles, respectively. It is important to realize that at this point it has been assumed that the Fresnel equations in eq. (2.101) are valid at *each local point* of reflection on the surface. This means that the curvature of the surface must be small compared to the wave length. This is the most serious limitation of the Kirchoff solution for frequencies in L-band. The curvature is proportional to $\frac{h}{\tau}$ where τ is a measure of the „wave length" of the surface or the correlation length in the case of random surfaces (see discussion later in this section).

Intuitively we can identify the various terms of eq. (2.109). The phase factor in the integral actually describes the phase difference between the locally reflected wave and the one reflected at the origin (O). The vector $\vec{v}$ is perpendicular to the reference plane and $\vec{p}$ is parallel to it. Thus, in the case of a smooth surface the first term in the second equation in eq. (2.109) will vanish and the second term is finite. When the normal vector $\hat{n}_\xi$ is tilted relatively to the smooth surface, the second term yields a contribution to the electric field at the expense of the first one.

In the following it is convenient to define the so-called scattering coefficient ρ. Let the electric field scattered by a perfect smooth, infinitely large conductor be denoted by E_{20}, then the scattering coefficient is defined as

$$\begin{aligned} \rho &:= \frac{E_2}{E_{20}} \\ &= \frac{1}{4L\cos(\theta_1)} \int_{-L}^{L} (a\xi' - b)\, \mathrm{e}^{iv_x x + iv_z \xi}\, dx \end{aligned} \tag{2.112}$$

This is the general formula for the scattering off a one-dimensional rough surface, with finite conductivity. The integral, however, is not as trivial as it looks, because a and b depend on x in a complicated way. Therefore, it is necessary to make some assumptions and simplify the expression eq. (2.112). In the following the case of infinite conductivity (i.e. $Y \to \infty$ in eq. (2.102)) is considered. In this case the reflection coefficient R becomes 1 for horizontal polarization and -1 for the vertical polarization.

At first this may appear as a severely limiting assumption, but as we shall see later, in the case of random surfaces with additional approximations the resulting scattering coefficient can be generalized to the finite conductivity case in a simple manner. Setting[13]

[13] corresponding to the horizontal and vertical polarizations

$R = \pm 1$ and integrating by parts, ρ can be expressed as

$$\rho_{\pm} := \frac{E_2^{\pm}}{E_{20}} = \pm \frac{F_2}{2L} \int_{-L}^{L} \mathrm{e}^{i\vec{v}\cdot\vec{r}}\, dx \tag{2.113}$$

where $\pm$ stands for horizontal and vertical polarization, respectively. The factor F_2 is defined as:

$$F_2 := \frac{1 + \cos(\theta_1 + \theta_2)}{\cos(\theta_1)[\cos(\theta_1) + \cos(\theta_2)]} \tag{2.114}$$

The important thing about eq. (2.113) is that the only dependence on x is contained in the exponent.

Random Surfaces In principle the expression eq. (2.113) is valid for any surface described by $\xi(x, y)$. The surface can be periodical or random. In this thesis only random surfaces are considered. In figure (2.38) a random surface is shown schematically. Let us now assume the surface to be generated by a Gauß-ian, stationary process, i.e. the divergence from the planar smooth surface has a Gauß distribution with variance σ^2. Further assume the autocorrelation function of the process to be exponential with a correlation length τ. Both curves in figure (2.38) have the same variance, but the blue one (upper) has a much shorter correlation length than the red one (lower).

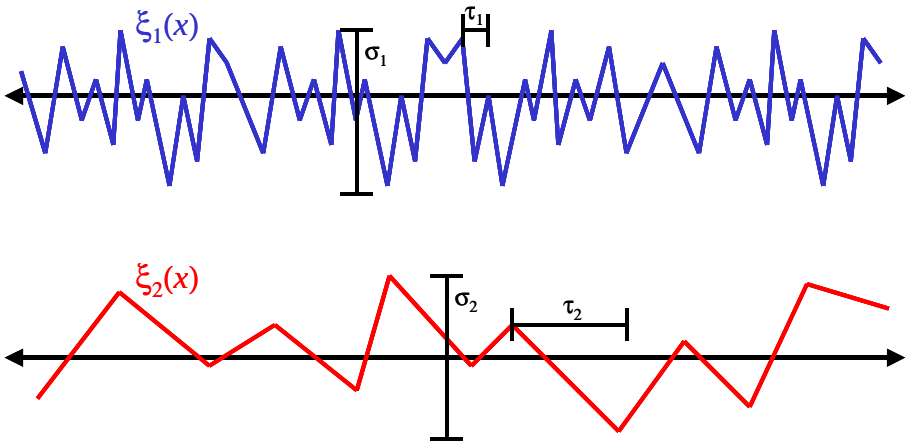

Figure 2.38.: The two characteristic parameters for Gauß-ian random surfaces. Both surfaces in the figure have the same variance (σ^2), but different correlation lengths (τ).

The quantity of interest here is the reflected *power* and not the electric field. Therefore we are interested in $\langle \rho\rho^* \rangle$ rather than $\langle \rho \rangle$. Inserting the Gauß-ian surface into eq. (2.113)

and performing the average, we get

$$\langle \rho\rho^* \rangle = \mathrm{e}^{-g} \left(\rho_0^2 + \frac{\sqrt{\pi} F_2^2 T}{2L} \sum_{k=1}^{\infty} \frac{g^k}{k!\sqrt{k}} \, \mathrm{e}^{-v_x^2 \tau^2 / 4k} \right) \quad \text{for one dimension} \tag{2.115}$$

$$\langle \rho\rho^* \rangle = \mathrm{e}^{-g} \left(\rho_0^2 + \frac{\pi F_3^2 \tau^2}{A} \sum_{k=1}^{\infty} \frac{g^k}{k!k} \, \mathrm{e}^{-v_{xy}^2 \tau^2 / 4k} \right) \quad \text{for two dimensions} \tag{2.116}$$

where the first equation is valid for a one-dimensional surface. In this section only 1-dimensional surfaces were considered, but the expression for the two-dimensional surface is obtained in an analogous manner. The interested reader is referred to [BS87], section 3.2 and section 5.3.

The parameter ρ_0 in eq. (2.115) and eq. (2.116) is the scattering coefficient for a finite, smooth, perfectly conducting reflector. It is given by

$$\rho_0 = \mathrm{sinc}(v_x L) \quad \text{one-dimension} \tag{2.117}$$

$$\rho_0 = \mathrm{sinc}(v_x L)\,\mathrm{sinc}(v_y L) \quad \text{two-dimensions} \tag{2.118}$$

where v_x and v_y are the x and y components of the vector $\vec{v}$ in eq. (2.110). The dimensions of the reflector is given L in one dimension and $L \times L = A$ in two dimensions. The factor g is short hand for

$$\sqrt{g} := 2\pi \frac{\sigma}{\lambda} \left[\cos(\theta_1) + \cos(\theta_2) \right] \tag{2.119}$$

The factor F_2 in eq. (2.115) is given by eq. (2.114) for one dimension and by

$$F_3 := \frac{1 + \cos(\theta_1)\cos(\theta_2) - \sin(\theta_1)\sin(\theta_2)\cos(\theta_3)}{\cos(\theta_1)[\cos(\theta_1) + \cos(\theta_2)]} \tag{2.120}$$

for two dimensions (eq. (2.116)).

In figure (2.39) the specular reflection coefficient is plotted as a function of incident angle. The values of σ are the same that were obtained using the Rayleigh criterion at normal incidence, i.e. 2.5, 1.25 and 0.625 cm. (see eq. (2.116) and figure (2.36)). We see that for normal incidence the scattered power in the specular direction is considerably reduced. For $\sigma = 2.5$ cm the scattered power is reduced by an order of magnitude for normal incidence! For oblique incidence the situation is better.

In figure (2.40) the factor g in eq. (2.119) is shown. We see that for a roughness of more than $\sigma = 1.25$ cm we find ourselves in the region of a moderately rough surface.

For the purposes of this work, we shall consider surfaces that have $\lambda \ll \tau \ll L$ where L is the linear dimension of the reflector. Then the dependence on the correlation length can be ignored and the main contribution of the scattered power is due to the specular term in eq. (2.116). Thus in this approximation we have

$$\boxed{\langle \rho\rho^* \rangle \approx \langle \rho\rho^* \rangle_{\text{spec}} \approx \mathrm{e}^{-g}\,|_{\theta_1=\theta_2} = \exp\left[-\left(\frac{4\pi\sigma\cos(\theta_1)}{\lambda} \right)^2 \right]} \tag{2.121}$$

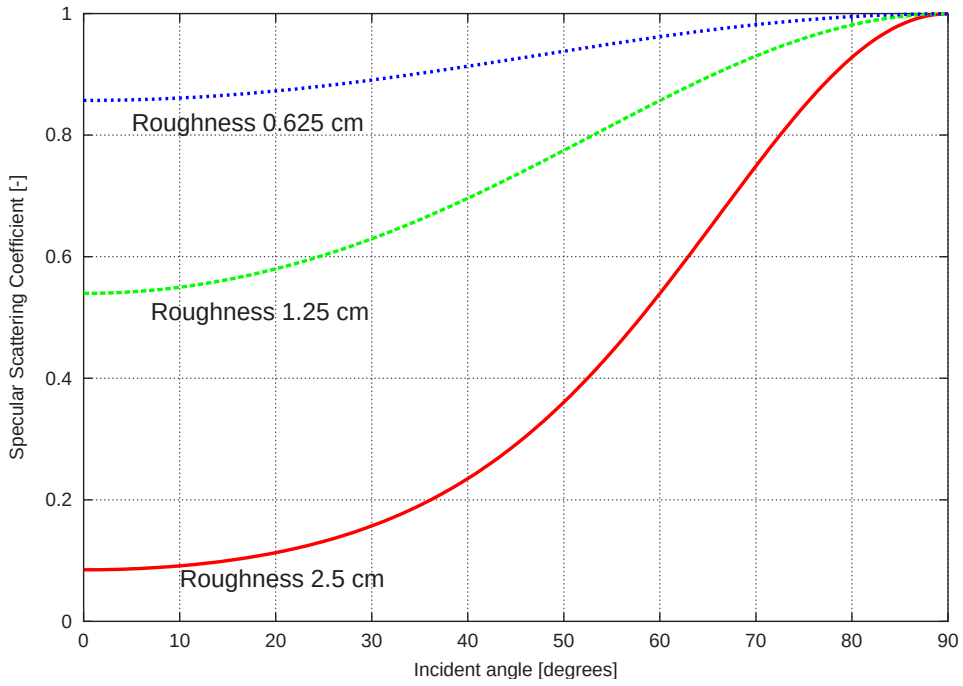

Figure 2.39.: The specular scattering coefficient for the scattering of an electromagnetic wave of a wave length $\lambda = 20$ cm as a function of incident angle. The three curves correspond to a roughness of $\sigma = 2.5$, 1.25 and 0.75 cm respectively. These are exactly the values of the parameter h in eq. (2.106) for normal incidence and $\Gamma = 8$, 16 and 32 (see figure (2.36) and eq. (2.106))

Considering the surfaces of most buildings, this approximation seems sufficient. To account for natural terrain, these approximations are probably not valid. For radio waves with wave length of c.a. 20 cm one has to ask oneself if the Kirchoff solution is valid in that case at all. Considering natural surfaces like grass, regions with gravel, were each stone has a typical diameter of, say, 1-5 cm. It is questionable whether the Fresnel equations can be applied locally on those surfaces for this wave length.

Finite Conductivity The analysis above assumes a perfectly conducting reflector. However the effect of the roughness is found to depend mainly upon the *shape* of the surface and not the electric properties.

Basically, the problem with finite conductivity involves the dependence of a and b on $\xi(x)$ in eq. (2.112). If it is assumed that

$$\langle a\xi' \rangle \approx \langle a \rangle \langle \xi' \rangle \tag{2.122}$$

then it can be shown ([BS87], section 5.4, p. 97) that the scattering coefficient for finite conductivity is given by

$$\langle \rho\rho^* \rangle_{\text{finite}} \approx \langle RR^* \rangle \langle \rho\rho^* \rangle_{\infty} \tag{2.123}$$

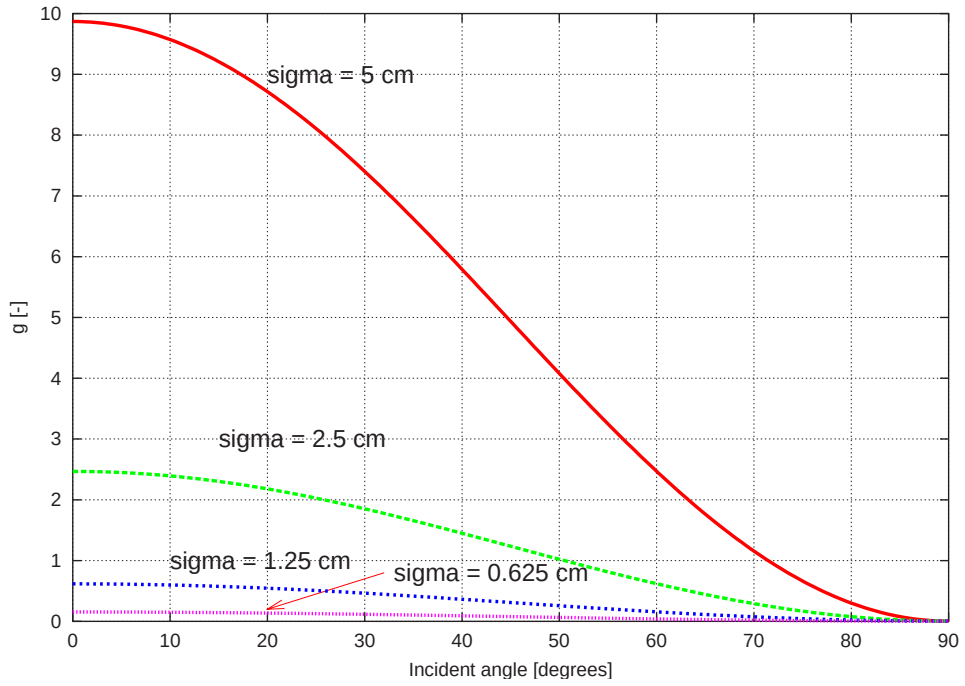

Figure 2.40.: The g factor in eq. (2.119) as a function of incident angle. The wave length of the scattered wave was $\lambda = 20$ cm. The four curves correspond to a roughness of $\sigma = 5$, 2.5, 1.25 and 0.625 cm respectively.

where the sub-scripts ∞ and finite denote the scattering coefficient for infinite and finite conductivity, respectively. It is therefore justified to calculate the scattering coefficient for an infinite conductor and then „correct" the result with the reflection coefficient evaluated for the reference plane (i.e. for $\xi(x) = 0$).

2.3.4.2. Roughness Larger Than the Carrier Wave Length

It would now be possible to use the equations of the previous section to obtain an expression for the scattering when the surface is very rough. Such an approach would, however, not fit into the overall scheme of things in the modeling and implementation of SNSS . The main reason is that the geometric ray-tracing only yields the specular reflections and the geometric information to sum up the fields from reflectors that „nearly" fulfill Snell's law simply isn't there.

A more straight-forward approach, from an implementation point of view, is to generate a random number of virtual reflectors around a real one. The target application here are oceanic surfaces. A similar idea is discussed in [BS87], section 6.4, where the surface is explicitly generated by a Markov process.

The idea used here is illustrated in figure (2.41)

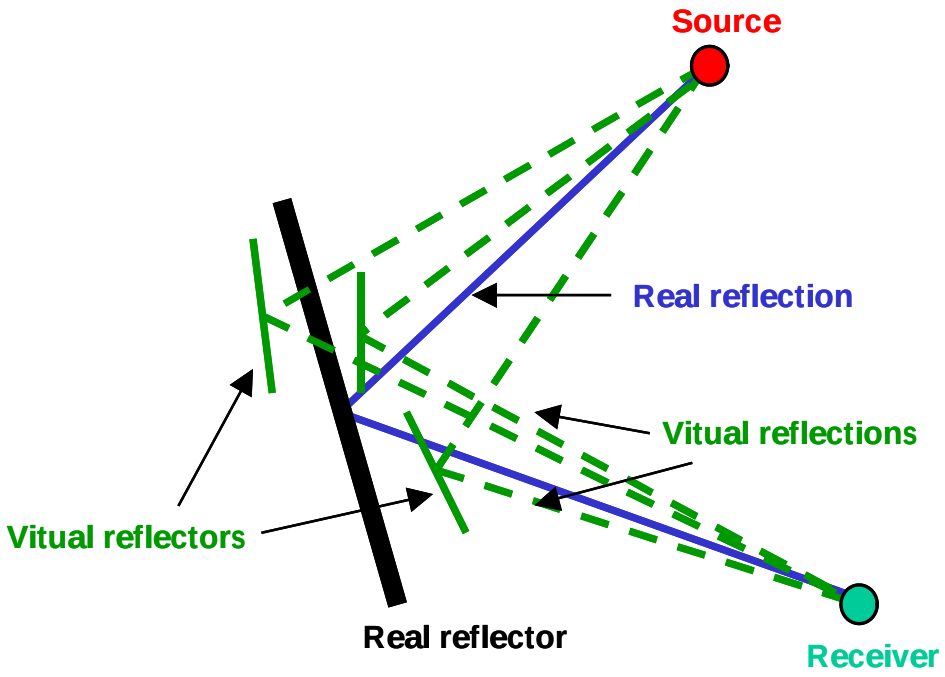

Figure 2.41.: After the geometric ray-tracing has successfully found all specular reflectors, virtual reflectors are generated as illustrated in the figure.

$$E_{\text{total}} = \frac{\hat{e}_r}{M} \sum_{j=0}^{N} E_j \, \mathrm{e}^{i\vec{k}\cdot(\vec{r}+\Delta\vec{r}_j)} = \hat{e}_r E_0 \, \mathrm{e}^{i\vec{k}\cdot\vec{r}} + \frac{\mu}{M'} \sum_{j=1}^{N} E_j \, \mathrm{e}^{i\vec{k}\cdot(\vec{r}+\Delta\vec{r}_j)} \tag{2.124}$$

where E_0 is the electric field as it is obtained from the procedures in the previous sections. E_j for $j \geq 1$ is the amplitude of the virtual field and $\Delta\vec{r}_k$ corresponds to the phase of the virtual signal. N is the number of virtual reflectors and M is a normalization. It is assumed that the reflectors are far away from the receiver, so that the unit vector $\hat{e}_r$, pointing from the specular reflection point to the receiver, can be considered the same for all virtual reflectors.

The normalization M' is chosen such that the power of the virtual scatterers can be adjusted with the parameter μ:

$$M'^2 := \frac{1}{E_0^2} \sum_{k=1}^{N} E_j^2 \tag{2.125}$$

With this normalization the power of the virtual scatterers is scaled relatively to the reflected power of the actual specular scatterer. The relative amplitudes, E_j are diced out through some statistical distribution, e.g. a Gauß-ian one with a certain variance σ_{E/E_0} and mean. The mean value of the distribution is irrelevant, because of the normalization.

As can be seen in eq. (2.124) the vectors $\Delta\vec{r}_j$ are not needed, only the scalar value $\phi_j := \vec{k} \cdot \Delta\vec{r}_j$. Some statistical distribution is also assumed for the phases.

Usually, when such approaches are made, one wants to let $N \to \infty$. Here that is not the case. This construction is very easy to implement in the framework of SNSS and it is *desired* to choose N some finite value, say $N = 5$, 10 or 100.

In the approach in the previous section the scattering coefficient basically yields an electric field which collectively represents the field at the antenna. The averaging constitutes the limit $N \to \infty$ and all the contributions from the local reflections are summed up into one single field with, perhaps, a shift in phase. This corresponds to the tap delay line approach used in communications.

This limitation is exactly what we want to avoid in this model. An important aspect of this model is that the receiver 'sees' a finite number of different phases. In the receiver model the main quantities of the signal is the SNR and the phase. When there is multipath, i.e. reflections are also received, a linear combination of fields with different phases is received. When this combination contains a finite number of contributions, the relative phases between the received signals is relevant and it is *not* desired to have that averaged out. If the phases are averaged out before entering the tracking loops we don't know how the *tracking loops* react to the multiple phases. If an average has to be taken, then it cannot be done independent on the structure of the discriminator and the tracking loop behavior. Therefore this process is implemented explicitly in the receiver model.

2.4. Modeling the Receiver

The receiver modeling is one of the most important part of this work. It starts at the antenna and finishes at the calculated position. The components of the generic receiver are:

- Antenna
- Down conversion
- Correlation
- Discriminator
- Tracking
- Positioning

These modules all have their counterparts in theory as well as in reality. In this chapter we will establish theoretical models for each of these components and discuss their strengths and limitations.

2.4.1. Overview of the Generic GNSS Receiver Structure

In this section an overview of the receiver structure is given. The various components of the receiver hardware are described in general terms.

2.4.1.1. Structure of a Modern GNSS Receiver

Modern GNSS receivers are mostly digital, i.e. the analog-digital conversion is carried out as early as possible. Usually the first step in the down-conversion[14] is done in the radio component, after that the signal is digitized. The tracking of code and carrier is of course also fully digital and is mostly implemented in digital signal processors and/or RISC processors.

The hardware responsible for correlating and tracking it is called a channel. For each satellite that is being monitored, a channel is needed. Most earlier receiver had only one channel and therefore the channel was switched between the satellites in view for monitoring. This switching is called multiplexing. Modern receivers have an array of channels, which can be operated in parallel, independent on each other.

A GNSS receiver consists of four sub-systems:

[14] The transition from high-frequency to base-band.

- Antenna
- Radio frequency component (RF-component)
- Signal processing component
- Data processing component

RF-component The RF-component is composed of an antenna, usually in L-band, a low noise amplifier (LNA) and the RF component itself. In the RF-component the conversion to an intermediate frequency is performed. Depending on the implementation this is done in one or more steps. There exists receiver where the down conversion is done in one step by ultra fast A/D conversion[FFF+91].

Signal Processing Component The analog radio signal is fed into the A/D converter and the output is a high-frequency data stream. The sampling rate for current commercial receivers are 6 MHz and 60 MHz for commercial C/A receivers and P(Y) code receivers respectively[War94]. However, some high-end C/A code receivers may have sampling-rates significantly exceeding 6 MHz. Commercial receivers mostly use a 1-bit converter as opposed to multiple bit converters[Die94]. After the sampling process, further down-conversion steps may follow. An effective down-conversion can also be achieved through an appropriate choice of sampling frequency. The digital data stream is now inserted into the channels. The signal is then multiplied with the reference code in each channel and accumulated. This is the correlation process. The instantaneous values in the accumulation buffers can be read out by the tracking loops for code and carrier tracking.

Data Processing Component The data processing unit controls the signal processors, reads out the tracking loops for code and carrier as well as demodulating the navigation messages. Further it calculates the position and synchronizes to the system time.

The reference oscillator plays a fundamental role in the receiver; it delivers the necessary reference frequency for the down-conversion, the A/D conversion and code-generation. Further information on GNSS receivers can be found in [War94] and [Blo92]

2.4.2. Receiver Hardware Technology – State-of-the-Art

The principles of the technologies used in GNSS receivers can be found in e.g. [Sen91]. In this section important technologies are summarized.

GNSS Antenna Generally there are two categories of antennas: ones with a constant antenna diagram and ones with a variable antenna diagram. The gain of an antenna depends on the azimuth and elevation of the incoming signal. For an antenna of the second type the gain can be changed dynamically. This is implemented using an array of antennas, so-called phased arrays, whose relative delay can be changed. Thus constructive and destructive interference can be induced, depending on the direction the satellite signal is coming from. A high gain in the direction of the satellites and a low gain elsewhere can be achieved. With this technique jamming and multipath can effectively be mitigated. Compared to the usual constant gain pattern antennas, these are very sophisticated and expensive devices.

As the GNSS signals are modulated onto an electromagnetic wave with helicity -1 (right hand circularly polarized) the antenna should have a high gain for a RHC wave. The three most popular antenna types are: Patch antennas, helix antennas and crossed dipole antennas. A simple means of multipath mitigation is the so-called choke ring. Basically this is a ground plate or ring structure mounted at the bottom of the antenna in order to mask out multipaths incident from negative elevation angles.

RF-Unit The radio front end is often implemented with ceramic filters and SAW (Surface Acoustic Wave) filter. The low noise amplifier (LNA) is built with GaAS technology and bi-polar transistors. For the frequency reference VCOs (Voltage Controlled Oscillators) with a PLL (Phase Lock Loop) is used. In recent years the trend goes towards using high quality RF - ASICs (Application Specific Integrated Circuits) and MMIC (Microwave Miniaturized Integrated Circuit) technologies[Eis97].

Oscillator The reference oscillator is almost exclusively a quartz oscillator (XO). They are of varying quality and some are computer compensated, temperature compensated or temperature stabilized (i.e. kept at a constant temperature).

Signal Processing An important requirement for building a digital GNSS receiver is a powerful digital signal processor. The currently applied DSP are constructed as ASICs using CMOS technology. The ASIC consists of an array of channels, each one having its own code generator, correlator and accumulator. The input is the down-converted and digitized signal and the output is the raw data (code, carrier and navigation messages). On current ASICs more than 24 C/A channels can be integrated onto one chip. As an example of the performance of a DSP in a GPS receiver we take a look at the Texas Instruments TMS 320 VC33:

- 32 Bit RISC ("Reduced Instruction Set") Architecture
- 150 MFLOPS ("Million Floating Point Operations per Second")
- 75 MIPS ("Million Instructions per Second")

- 77 MHz processing frequency

The TMS 320 VC33-150 is from the product line of Texas Instruments in 2002-2003.

Data Processing The computation of the position once the raw data is available is not very difficult and does not require special purpose DSPs. For the navigation data processing a normal PC-type processor is adequate. The Motorola 68020 CPU is an example of a microprocessor used in GPS receivers:

- 32 Bit CISC ("Complete Instruction Set") Architecture
- 0.14 MFLOPS
- 25 MHz processing frequency

These figures are from 2002-2003. The performance of the special purpose DSP is thus roughly 500 times higher than the navigation processor.

2.4.3. GNSS Receiver Signal Flow

2.4.3.1. Introduction

In this section the signal flow in a generic GNSS receiver is described. The block diagram serves as a reference for the mathematical description of the generic GNSS receiver. Such block diagrams can be found in various references, e.g. [Spi80],[DFF92], [ICA92], [War94] and [Bec94].

One of the main difficulties in constructing a mathematical model of a GNSS receiver is to keep the model as general as possible. For some sub-systems this is possible, while others must be divided into cases. For the code tracking loops the following cases are considered:

- Coherent DLL
- Non-coherent DLL

The phase locked loop (PLL), which tracks the carrier phase, is often implemented with the so-called Costas-loop. The two cases considered here are:

- Product (Costas) detector
- Arctan or Tan detector

The coherent DLL requires the phase locked loop to be in lock, i.e. it is tracking the carrier phase and that the navigation bit has been de-modulated, i.e. removed from the signal. On the other hand the non-coherent DLL is able to track the code even with the data bit on the signal and the PLL is not required to be in lock. As the DLL is much more robust than the PLL under non-optimal signal condition (high dynamics, low signal-to-noise ratio), the non-coherent design is the preferred one.

The difference between the Costas and the tan/arctan detector is not as important as between the coherent and non-coherent DLLs. The tan/arctan detector removes the correlation function dependency by a division of two signals, while the Costas detector depends on the correlation function.

2.4.3.2. Receiver Block Diagram

In reference to the above mentioned literature the receiver signal model is based on the block diagram in figure (2.42).

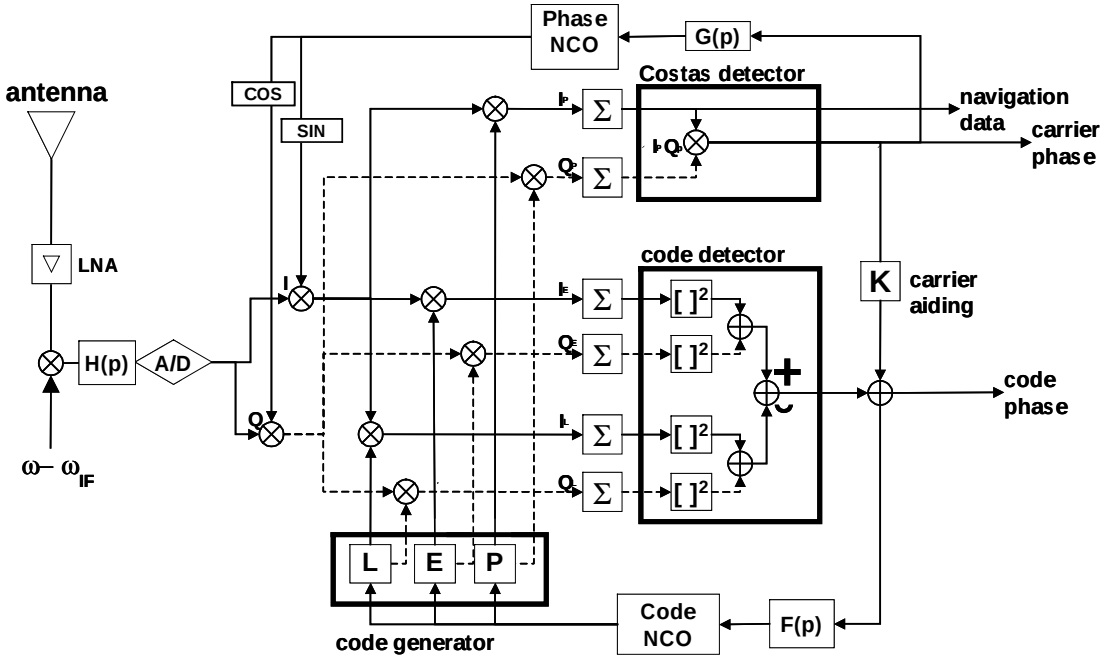

Figure 2.42.: Receiver block diagram. The dashed lines show the Q-channel.

In figure (2.42) the following abbreviation were used:

- LNA: Low Noise Amplifier
- ω, ω_{IF}: GNSS carrier and intermediate frequency respectively
- $H(p)$: Transfer function of the IF-

filter

- A/D: Analog - Digital converter
- I: In-phase component
- Q: Quadra-phase component
- L: Late channel
- E: Early channel
- P: Punctual channel
- $\otimes$: Signal multiplication
- $\oplus$: Signal addition
- $\sum$: Integration (accumulation)
- NCO: Numerical Controlled Oscillator
- $G(p)$: Transfer function of the phase locked loop
- $F(p)$: Transfer function of the delay lock loop
- K: Gain factor for carrier aiding

In the following the signal flow in the diagram in figure (2.42) will be described.

IF-Filtering The signal received at the antenna is first amplified in the LNA (low-noise amplifier). After that the signal is down-converted by multiplying it with a sine of frequency $\omega - \omega_{IF}$. The high-frequency ($2\omega - \omega_{IF}$) component in each channel are simply filtered out in a band-pass filter (The filter with the transfer function $H(p)$ in figure (2.42)). The bandwidth of the filter must be large enough to allow the signal with the code to pass, taking into account possible Doppler shifts and drifts in the reference oscillator. If the bandwidth is too small, the entire spectrum of the signal will not enter the channel and thus information is lost. A smaller bandwidth, on the other hand, will effectively block out noise, which would otherwise enter the channel. When choosing the bandwidth for the pre-correlation filter (i.e. $H(p)$) a compromise between these two factors must be found.

A/D Conversion After the pre-correlation filtering the analog signal is sampled and converted to a digital data stream. The digital signal now consists of an effective carrier with frequency ω_{IF} modulated with the spreading code and navigation data. It is then multiplied with the reference spreading code, modulated onto a carrier sine and a cosine, to produce the I and Q channels respectively. Two orthogonal data streams result from this operation, the in-phase (I) and quadra-phase (Q) channels. In the bock diagram in figure (2.42) this is done in two steps. The actual intermediate frequency (the nominal IF-frequency with Doppler shift and phase shift) is achieved with the phase locked loop (Costas loop). The I and Q channels are then fed into the correlators.

Correlation The correlation is simply an integration or more precisely an accumulation of the received signal multiplied with the reference code in the I and Q channels. The time and phase offset of the reference code relative to the signal can be adjusted. In most receivers the three copies of the reference code are produced: an early, a punctual

and a late version. These copies have a fixed, constant offset relative to each other of half a chip duration. With this configuration a local maximum of the symmetric part of the correlation function can be tracked. If more information on the shape of the correlation function is desired, more correlators are needed. Under ideal conditions there is no need for such information, but when reflections of the direct signal are also received, the correlation function is not symmetric any more and the correct signal delay is not tracked. The multipath estimating DLL (MEDLL) uses an *ansatz* like that.

The accumulation time that the correlators can be set to depends on the signal structure and what kind of discriminator is used. For the non-coherent discriminator, tracking a C/A GPS signal the integration time T is usually set to the duration of the navigation bit, i.e. 0.02s.

Code Discriminator The code discriminator, sometimes called the S-curve [Nee94] is constructed from the output of the correlators to have a zero at the correct code delay and a well-defined slope at the root. The explicit form of the S-curve depends on the signal structure received and defines the type of receiver.

In the coherent case it is assumed that the unknown navigation bit has been de-modulated in the phase tracking loop and removed from the signal. The discriminator is then simply the difference between the early and late correlator signals.

In the non-coherent case the results of the correlators is squared and thus the data bit is removed. This of course assumes that the data bit did not change during the integration in the correlators. This explains why the integration times in non-coherent receivers has to be smaller or equal to the duration of the data bit. After squaring the signals the early and late correlator signals are subtracted from each other. The results from the I and Q channels are then added.

These two implementations have their pros and cons. The non-coherent discriminator has a higher noise, caused by the squaring of the signal (squaring loss). On the other hand, as will be shown later, the non-coherent discriminator does *not* depend on the tracking error of the phase locked loop. This is a desirable feature, because the DLL is usually more stable than the PLL.

Code Tracking Loop The code tracking loop or the delay lock loop (DLL) has a transfer function ($F(p)$) and controls the numerically controlled oscillator (NCO) of the code generator, where the reference signals are generated. The frequency of the NCO controlling the code generator is shifted such that an optimal alignment between the received signal and the reference code is achieved. This corresponds to the zero of the discriminator (S-curve). The closed loop filter can be described by a differential equation of the same order as the loop filter.

Phase Discriminator Additionally to tracking the code phase of the signal, the carrier phase is also tracked. Firstly it is needed to de-modulate the navigation data and secondly the phase information yields a very precise estimate of the change in distance to the satellite. This is achieved trough a combination of the punctual I and Q signals. The Costas detector is a multiplication of the punctual signals in the I and Q channels, i.e. $I_P Q_P$. Another possibility is to divide the in-phase with the quadra-phase signal. Here again the zero of the discriminator corresponds to the correct phase of the signal. With this operation the navigation data can be detected.

Phase Locked Loop (PLL) The PLL has the transfer function $G(p)$ and controls the phase NCO. The role of the phase NCO is to generate the desired frequency from the reference oscillator. It compensates for Doppler shifts and drifts and shifts in the reference oscillator. The sine and cosine of the resulting frequency is then used for the down-conversion and the generation of the I and Q channels and that closes the tracking loop. The goal of the phase tracking loop is to achieve a zero phase difference between the received signal and the reference oscillator. This tracking loop can be described with a differential equation.

Carrier Aiding Evidently the code and phase tracking loops are coupled. A further coupling can be achieved by feeding the tracked carrier phase into the DLL and thus provide velocity (Doppler) information to the DLL. That enables the DLL to respond more accurately in dynamic environments. When carrier aiding is used filter bandwidth of the DLL can be reduced and thus the noise performance enhanced.

2.5. Receiver Model

2.5.1. Correlation

In this section the correlation process in the receiver is discussed. In the chapter on signal structures some properties of the signals were analyzed. The purpose of that chapter is firstly to obtain some properties of the signal structures. Secondly, it is a necessary preparation for this section.

Before we proceed, we need to extend the signal model to accommodate multipaths and thermal noise. To this end we model the multipath signal as a linear combination of

signals of the type in eq. (2.12), with arbitrary delay- and phase-shifts and amplitude[15]

$$\begin{aligned} S_{\text{mp}}(\vec{\tau}(t), \vec{\phi}(t)) &:= \sum_{j=0}^{N} a_j S(t; \tau_j(t), \phi_j(t)) + n(t) \\ &= \sum_{j=0}^{N} a_j G^c_{T_c}(t + \tau_j(t)) \sin(\omega t + \phi_j(t)) + n(t) \\ &= \sum_{j=0}^{N} a_j \sum_{k=1}^{N_c} b_k g_{T_c}(t - kT_c + \tau_j(t)) \sin(\omega t + \phi_j(t)) + n(t) \end{aligned} \tag{2.126}$$

where a_j is the amplitude of the j-th multipath, ω is the carrier frequency and $\tau_j(t)$ and $\phi_j(t)$ are the path-delays and phase shifts of the j-th signal, respectively. As earlier defined (eq. (2.12)) the function $G^c_{T_c}(\tau)$ is just a short-hand notation for the sum over the whole code. Note that the zero-th multipath in eq. (2.126) corresponds to the direct signal. The shifts in phase and delay are unknown and it is one of the purposes of the simulation to calculate them as they are very hard to obtain analytically.

The term $n(t)$ is a Gauß-ian, white noise process with zero mean and variance 1. For corrrect normalization the amplitude of the direct signal is set to [Eis97, Die96]

$$a_0 = \sqrt{2T_p \frac{S}{N_0}} \quad \text{for the reflected signals} \quad a_j =: \eta_j a_0 \tag{2.127}$$

where the parameter η_j is defined as the signal-to-multipath ratio which describes the signal amplitude of the reflected signal relative to the direct one.

In communication there exists statistical models, whose parameters are based on measurements made in typical environments. These models have been shown to work very well for example for cellular phone systems.

Due to the geometry, it can be expected that the number of multipaths in such a system will be very large and thus the statistical models can be very useful. Furthermore communication systems receivers have a different architecture; the goal is to „absorb" as much signal energy as possible, while the time of arrival is not important. To maximize the received power the typical design of a communication receiver[16] takes advantages of the reflected signals and tries to integrate the contribution from the multipath signals. This also has consequences for the assumptions, upon which such statistical models are based. The typical tap delay-line models bundle up a large number of reflections to produce one „representative" or average for a particular (constant) code phase delay. When such a model is inserted into a receiver model that is designed to track the time of arrival of the signal, the effect of the actual signal and the „representative" could be quite different.

[15] note that the phase and the delay can not be be chosen independently, in the multipath case $\tau_j(t) = \omega\phi_j(t)$.

[16] Rake receivers

In satellite navigation systems the geometry is very different; the signal comes from satellites that, in general, are above the user. It can therefore be assumed that the number of reflections is much smaller than in the communications case. Another important observation is that in navigation the geometry itself is of primary interest and not the signal power[17]. Thus the *timing* of the *direct* signal is of primary interest, while influence of the reflected signals has to be suppressed as much as possible. So a part of the signal power is actually sacrificed in order to obtain more accurate information on the time of arrival.

For all these reasons we find it natural to assume a deterministic model like eq. (2.126) for the multipath signal and to implement it in a simulator that generates the multipath environment.

In the receiver the signal is correlated with a locally generated reference signal. Basically, three copies of the signal are generated: early, late and punctual. In the first step the reference signals are multiplied with the received one and then it is integrated over a time interval T_p. The correlation process has been implied in the section on the signal model. That analysis has to be extended to the model in eq. (2.126).

Assuming a geometric model for the delays and phase shifts in eq. (2.126) we get the following

$$\tau_j(t) =: \tau(t) + \delta_j(t)/c \qquad \phi_j(t) =: \omega\tau(t)_j = \omega\tau(t) + \omega\delta_j(t)/c \tag{2.128}$$

here the $\delta_j(t)$ is the difference in geometric path length between the reflected signal and the direct signal (j = 0). These delays are functions of time and thus Doppler shifts, jerks and higher order derivatives are implied.

Using the results from the section on signal structures, we can now construct a general, powerful model for the signal after pre-detection (i.e. after the correlation process).

As already implied, the receiver generates reference signals in-phase and quadra-phase. That means that there are two reference carriers with a relative phase shift of 90°. These two branches are called the in-phase- (or I-) and quadrature- (or Q-) channels. Apart from that, a typical receiver generates an early, punctual and a late signal. Thus a basic receiver generates six signal streams for each channel (see figure (2.42)). Realizing all this in a signal model lt us define the following function:

$$\begin{aligned} &Y(\Delta\tau(t) + u, \Delta\phi(t), \Delta\omega(t), \vec{\delta}(t), \vec{\eta}(t)) \\ &\qquad := \frac{1}{T_p}\int_{t-T_p}^{t} S_{\text{mp}}(t', \tau, \vec{\delta}(t'), \vec{\eta}(t'))\, s_{\text{ref}}(t' - \tau' - u, \omega', \phi')dt' + \xi_{u,Y}(t) \\ &\qquad = \sum_{j=0}^{N} a_k R^*_{\text{signal}}(\Delta\tau(t) + \delta_j(t)/c + u)\frac{1}{T_p}\int_{T_p} \mathrm{e}^{i\Delta\omega t' + i\Delta\phi_j(t')}\, dt' + \xi_{u,Y}(t) \end{aligned} \tag{2.129}$$

[17] Obviously, it is necessary to have signal power to obtain information on the timing and the accuracy of the timing will, in general, depend directly on the received power

where the second equation was obtained using eq. (2.18). The code error, $\Delta\tau_j(t)$ and the phase error, $\Delta\phi_j(t)$ are defined as

$$\Delta\phi_j(t) = \phi_j(t) - \phi'(t) = \Delta\phi(t) + \omega\delta_j(t)/c \tag{2.130}$$

and

$$\Delta\tau_j(t) = \tau_j(t) - \tau'(t) = \Delta\tau(t) + \delta_j(t)/c \tag{2.131}$$

The expression $Y(\cdot)$ in eq. (2.129) is a bit formal, but from it all six (and more) signal streams can be obtained. The I and Q channels are obtained by taking the real and imaginary part of eq. (2.129), respectively and the early, late and punctual version are achieved by substituting appropriate values for u.

In an effort to try to write out explicitly the various dependencies, the argument of $Y(\cdot)$ shows that the pre-detection model depends on the code error, $\Delta\tau(t)$, the phase error, $\Delta\phi(t)$ and on the frequency error, $\Delta\omega(t)$. Further, it depends on the geometry and the interaction of the signal with its environment. This is expressed by the two arguments ($\vec{\delta}(t)$ and $\vec{\eta}(t)$). As implied, all these quantities in turn depend on time. It will basically be one of the main goals of the implementation of the simulator to obtain the functions $\vec{\delta}(t)$ and $\vec{\eta}(t)$. The function R^*_{signal} represents any of the (complex) correlation functions that were obtained in the section on signal structures.

In the case were only one signal is present (i.e. in the absence of multipath) it is a good approximation to assume the phase $\phi(t)$ to be constant during the integration and thus pull the phase factor in front of the integral.

In the multipath case this can in general not be done. To see why, there are several time scales that we need to consider. Taking the GPS case we get:

- The chip-length, T_c is in the order of 300 m or 1μs.
- The period of the code, p. For GPS C/A code this is in the order of 300 000 m, or 1 ms. For the P-code this is much larger.
- The integration interval, T_p. In most receivers this is about 1/50 seconds, or 6 000 000 m.

So there are four orders of magnitude separating the integration time and the chip duration.

Assuming a receiver moving away from a reflector with a constant speed, s and the receiver–reflector–satellite geometry to lie in a line, we then have $\delta_1(t) = \delta_1^0 + st$ and $\delta_0(t) = \delta_0^0 - st$. This is illustrated in figure (2.43). If, during the integration time T_p, the phase of the multipath signal relative to the direct signal rotates exactly 2π then the multipath contribution would cancel out exactly. With a wave length of 0.2 m we have a phase rotation of 2π when $2sT_p = \lambda$. With an integration time of T_p = 20 ms, this yields s = 5 m/s! If this would be the exact configuration, then the multipath error would cancel out *completely* in the pre-detection process, *before* entering the tracking loops. As we see,

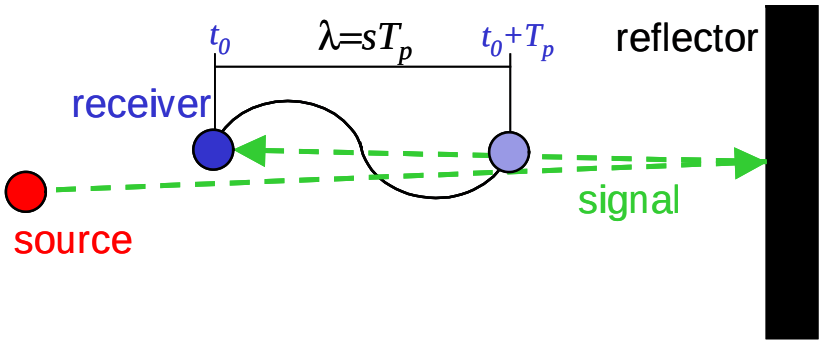

Figure 2.43.: Illustrating the pre-detection multipath fading effect. The receiver moves towards the reflector a distance $sT_p = \lambda$ within the integration interval T_p. This causes a complete cancellation of the multipath error in the pre-detection.

we are forced to keep the integral in eq. (2.129) at least in some form, particularly in the kinematic case.

It could be argued that the fading due to the tracking loop will set in much earlier, i.e. at lower frequencies, than the fading caused by the pre-detection and that is of course true. However, the mean value of the multipath error is not zero for the tracking loop fading as it is for the pre-detection fading. Therefore at some point the pre-detection fading will provide further fading, exceeding the tracking loop fading.

In the static case (i.e. s = 0 m/s) the relative Doppler shift between the direct and reflected signal caused by the satellite motion alone is only of order 1(mHz/m) $\cdot d$ (this analysis was performed in [Eis97]), where d is the distance between reflector and antenna. So for a distance d = 100 m the phase in the integral eq. (2.129) will only change about 1/1000 of a rotation during the integration interval. The assumption of the constant phase, $\Delta\phi_j(t')$ in eq. (2.129) is well justified.

To get the pre-detection multipath fading at least to first order, we assume a constant relative Doppler during the integration, i.e. $\dot{\delta}_j(t) = \text{const}$. It should be emphasized that this assumption is *only* made for the pre-detection. The signal entering the tracking loops will maintain the full description of the dynamics. In the following we will however assume the frequency error, $\Delta\omega$, to be zero. With this said, the pre-detection model now becomes

$$\begin{aligned} &Y(\Delta\tau(t)+u,\Delta\phi(t),\Delta\omega=0,\vec{\delta}(t),\vec{\eta}(t)) \\ &\qquad = \sum_{j=0}^{N} a_j \,\mathrm{sinc}(\Delta\omega_j T_p/2) R^*_{\text{signal}}(\Delta\tau(t)+\delta_j(t)/c+u)\, \mathrm{e}^{i\Delta\theta_j} + \xi_{u,Y}(t) \end{aligned} \tag{2.132}$$

where $\Delta\theta_j$ and $\Delta\omega_j$ are defined by

$$\Delta\theta_j := \Delta\phi(t_0) + \omega\delta(t_0)/c \quad \text{and} \quad \Delta\omega_j := \omega\dot{\delta}_j(t_0)/c \tag{2.133}$$

and t_0 is a time coordinate within the integration interval. Thus $\Delta\theta_j$ is the relative phase shift between the direct signal and reflection with index j and $\Delta\omega_j$ is the corresponding difference in Doppler shift.

We notice that the pre-detection multipath fading (i.e. the $\mathrm{sinc}(\cdot)$ factor between the eq. (2.132)) has the same form as the damping due to the frequency error. The origin is exactly the same; fading caused by de-tuning of the oscillator with respect to the reference. In the case of the frequency error the effect is undesirable, whereas in the multipath case this fading is very welcome. In fact, a more accurate interpretation of $\Delta\omega_j$ is the difference between the Doppler of reflection with index j and the frequency of the local oscillator. Thus $\Delta\omega_0$ has the meaning of $\Delta\omega$ in section 2.2, i.e. the difference between the frequency of the direct signal and the frequency of the receiver oscillator.

To write down the I- and Q-channels explicitly we have

$$\begin{aligned} &I(\Delta\tau+u,\Delta\phi) = \\ &\qquad \sum_{j=0}^{N} a_j \,\mathrm{sinc}(\Delta\omega_j T_p/2) Re\left\{ \mathrm{e}^{i\Delta\theta_j}\, R^*_{\text{signal}}(\Delta\tau(t)+\delta_j(t)/c+u)\right\} + \xi^I_{u,Y}(t) \\ &Q(\Delta\tau+u,\Delta\phi) = \\ &\qquad \sum_{j=0}^{N} a_j \,\mathrm{sinc}(\Delta\omega_j T_p/2) Im\left\{ \mathrm{e}^{i\Delta\theta_j}\, R^*_{\text{signal}}(\Delta\tau(t)+\delta_j(t)/c+u)\right\} + \xi^Q_{u,Y}(t) \end{aligned} \tag{2.134}$$

For most signals the correlation function R^*_{signal} is real and the I and Q channels differ only by the phase factor $\cos(\Delta\theta_j)$ and $\sin(\Delta\theta_j)$, respectively. But for side-band tracking of the signals in section (2.2) the correlation function was not purely real. Therefore eq. (2.134) is the simplest general form for the pre-detection model.

The noise terms in eq. (2.134) have been studied extensively by various authors [Eis97, Die92, Die96] and are found to be

$$E\{\xi^Q_{u,Y}(t)\} = E\{\xi^I_{u,Y}(t)\} = 0 \qquad E\{(\xi^Q_{u,Y}(t))^2\} = E\{(\xi^I_{u,Y}(t))^2\} = R_{\text{ref}}(0) \tag{2.135}$$

$$E\{\xi^I_{u,Y}(t)\xi^I_{u-d,Y}(t)\} = R_{\text{ref}}(d) \qquad E\{\xi^Q_{u,Y}(t)\xi^Q_{u-d,Y}(t)\} = R_{\text{ref}}(d) \tag{2.136}$$

$$E\{\xi^Q_{u,Y}(t)\xi^I_{u,Y}(t)\} = 0 \tag{2.137}$$

where $E\{\cdot\}$ denotes the expectation value. The correlation function in eq. (2.135) is the auto correlation function of the *reference* signal.

2.5.2. Detector Functions

In this section the detector functions for the tracking loops are presented and using the signal and the pre-detection models from previous sections the detector functions are

derived. The channels in the receiver are constructed such that the zeros of those discriminator functions are tracked, i.e. the frequency and code delay are adjusted to maintain the zero of the discriminator functions.

A common detector in use in GPS receivers today is the so-called early-late non-coherent detector. The term coherent refers to whether knowledge of the carrier phase is necessary. There are other possibilities to construct a discriminator for the code such as the coherent early-late discriminator and the cross-product detector.

The coherent detector is not as stable and robust as its non-coherent counterparts because the detection error can be shown to depend on the carrier phase error of the loop tracking the phase.

The cross-product detector is a non-coherent detector like the early-late non-coherent detector, but the noise performance is not as good. Further it is not very well suited for narrow correlation; its noise does not decrease with decreasing correlator spacing [Eis97, Die96].

In the following we will use a short-hand for eq. (2.134)

$$i_{\mathrm{e}} := I(\Delta\tau - d, \Delta\theta) \qquad i_{\mathrm{p}} := I(\Delta\tau, \Delta\theta) \qquad i_{\mathrm{l}} := I(\Delta\tau + d, \Delta\theta) \tag{2.138}$$

$$q_{\mathrm{e}} := Q(\Delta\tau - d, \Delta\theta) \qquad q_{\mathrm{p}} := Q(\Delta\tau, \Delta\theta) \qquad q_{\mathrm{l}} := Q(\Delta\tau + d, \Delta\theta) \tag{2.139}$$

were d is the correlator spacing.

2.5.2.1. Early–Late Non-Coherent Detector

The noise free non-coherent early–late discriminator is defined by the following discriminator function:

$$\begin{aligned} S_{nc}(\tau) &:= i_e^2 + q_e^2 - (i_l^2 + q_l^2) \\ &= \left[\sum_{j=0}^{N} a_j \,\mathrm{sinc}(\Delta\omega_j T_p/2)\, \mathrm{e}^{i\Delta\theta_j}\, R^*_{\text{signal}}(\Delta\tau - d + \delta_j/c)\right] \\ &\times \left[\sum_{k=0}^{N} a_k \,\mathrm{sinc}(\Delta\omega_k T_p/2)\, \mathrm{e}^{-i\Delta\theta_k}\, \overline{R^*_{\text{signal}}}(\Delta\tau - d + \delta_k/c)\right] \\ &- \left[\sum_{j=0}^{N} a_j \,\mathrm{sinc}(\Delta\omega_j T_p/2)\, \mathrm{e}^{i\Delta\theta_j}\, R^*_{\text{signal}}(\Delta\tau + d + \delta_j/c)\right] \\ &\times \left[\sum_{k=0}^{N} a_k \,\mathrm{sinc}(\Delta\omega_k T_p/2)\, \mathrm{e}^{-i\Delta\theta_k}\, \overline{R^*_{\text{signal}}}(\Delta\tau + d + \delta_k/c)\right] \end{aligned} \tag{2.140}$$

This expression is valid for all correlation functions from section 2.2. The bar over the correlation function symbolizes complex conjugation. For signals with a pure real correlation

function this is just the square of the correlation functions and the familiar non-coherent expression is obtained. From eq. (2.140) we see that a complex correlation function results in a correlation between the I and Q channels. Further it is seen that the diagonal terms (i.e. for which $j = k$) are independent of the phase error. However, the off-diagonal terms will in general depend on the phase errors.

Due to the squaring the noise term becomes a bit complicated. Using the short-hand notation introduced above, we get

$$\begin{aligned} D_{nc}(\tau) &:= (i_e + \xi_e^I)^2 + (q_e + \xi_e^Q)^2 - \left[(i_l + \xi_l^I)^2 + (q_l + \xi_l^Q)^2\right] \\ &= i_e^2 + q_e^2 - (i_l^2 + q_l^2) + 2i_e\eta_e - 2i_l\eta_l - 2q_e\xi_e + 2q_l\xi_l \\ &\quad + \eta_e^2 - \eta_l^2 - \xi_e^2 + \xi_l^2 \end{aligned} \tag{2.141}$$

Later on we will construct a dynamic model for the code and phase measurement and the discriminator will become a driving force of a differential equation. Basically, due to the noise terms we will actually be confronted with a system of stochastic differential equations. The goal is to solve those equations numerically. Looking at eq. (2.141) we notice that there will be multiplicative Gauß-ian noise terms and then there are the quadratic Gauß-ian terms ξ^2 and η^2. The multiplicative noise is not problematic but the quadratic terms are. As these random variables are not Gauß-ian, but rather have a Chi-square distribution, the standard theory of stochastic differential equations cannot be applied directly. The approach adopted here is to calculate the first two moments of eq. (2.141) and then assume Gauß-ian additive noise. This is often called **a**dditive **w**hite **G**auß-ian **n**oise (AWGN-channel). This work was already done in [Eis97] for the GPS case. The expectation value of eq. (2.141) is just the noise free case eq. (2.140). The variance is more complicated and its derivation can be found in [Eis97]. Extending those results to a general form of the auto-correlation function, we get:

$$\text{Var}\left\{D_{nc}(\tau)\right\} = 8a^2 R^*_{\text{signal}}(d)\overline{R^*_{\text{signal}}(d)}(1 - R_{\text{ref}}(2d)) + 8(1 - R^2_{\text{ref}}(2d)) \tag{2.142}$$

$$\text{Var}\left\{D_{nc}(\tau)\right\} = 8a^2 R^2(d)\left[R(0) - R(2d)\right] + 8\left[R(0) - R^2(2d)\right] \tag{2.143}$$

The starred versions of the correlation functions originate from the multiplicative noise terms, while the auto-correlation functions ($R_{\text{ref}}(\cdot)$) come from the correlation of the reference signal with the received additive noise.

As a simple example of eq. (2.140) let us solve for the multipath error in the standard, infinite bandwidth rectangular pulse shape case, under the assumption that there is one direct signal and a reflected signal. In this case the zeros of eq. (2.140) can be solved explicitly. Due to the absolute values in the expression for $R(\tau)$ several cases must be considered. Here we only consider the case of small geometric delays. In this case the

early and late correlation functions can be written as:

$$\begin{aligned} R(\tau - d) &= 1 - \frac{|\tau - d|}{T_c} = 1 + \frac{\tau - d}{T_c} \\ R(\tau + d) &= 1 - \frac{|\tau + d|}{T_c} = 1 - \frac{\tau + d}{T_c} \end{aligned} \tag{2.144}$$

Inserting eq. (2.144) into eq. (2.140) and without loss of generality setting the phase error of the PLL to zero, we then get for the I and Q components, respectively:

$$\begin{aligned} i_e^2(\tau) - i_l^2(\tau) &= \frac{4\eta^2}{T_c}(1-d)\sin^2(\omega\delta)(\tau - \delta) \\ q_e^2(\tau) - q_l^2(\tau) &= \frac{4}{T_c}(1-d)\left[\tau + \eta^2\cos^2(\omega\delta)(\tau-\delta) + \eta\cos(\omega\delta)(2\tau - \delta)\right] \end{aligned} \tag{2.145}$$

As we are considering the non-coherent case the result is independent on the phase error. Substituting eq. (2.145) into eq. (2.140) and solving S_{nc} = 0 we get:

$$\tau = \eta\delta\frac{\eta + \cos(\omega\delta)}{1 + \eta^2 + 2\eta\cos(\omega\delta)} \tag{2.146}$$

Note that for $\omega\delta = n2\pi$ the expression eq. (2.146) seems singular for $\eta = 1$. The zero frequency limit is also of interest. Taking these two limits we obtain:

$$\tau(\omega = 0) = \delta\frac{\eta}{1+\eta} \qquad \text{and} \qquad \tau(\eta = 1) = \frac{\delta}{2} \tag{2.147}$$

The zero frequency case corresponds to the case where the frequency is high and the result is low-pass filtered. This will will become evident in the simulations where dynamic effects are taken into account.

2.5.2.2. Costas Detector

The phase of the signal is measured by tracking the zero of the phase discriminator. The discriminator analyzed here is the so-called Costas discriminator and it is simply the product of the punctual I and Q channels. The noise free case is given by

$$\begin{aligned} D_{\text{PLL}} &:= i_p q_p \\ &= \left[\sum_{j=0}^{N} a_j \,\text{sinc}(\Delta\omega_j T_p/2)\text{Re}\left\{\text{e}^{i\Delta\theta_j}\, R^*_{\text{signal}}(\Delta\tau + \delta_j/c)\right\}\right] \\ &\times \left[\sum_{k=0}^{N} a_k \,\text{sinc}(\Delta\omega_k T_p/2)\text{Im}\left\{\text{e}^{i\Delta\theta_k}\, R^*_{\text{signal}}(\Delta\tau + \delta_k/c)\right\}\right] \end{aligned} \tag{2.148}$$

With noise the discriminator is written as

$$D_{\text{PLL}} := (i_p + \eta_p)(q_p + \xi_p) = i_p q_p + i_p\xi_p + q_p\eta_p + \eta_p\xi_p \tag{2.149}$$

Again there is a quadratic term in the Gauß-ian noise. The variance and mean of eq. (2.149) at the tracking point is given by:

$$\mathrm{E}\left\{S_{\mathrm{PLL}}\right\} \approx a^2 \Delta\theta \qquad \mathrm{Var}\left\{S_{\mathrm{PLL}}\right\}|_{\Delta\tau=0} = a^2 + 1 \tag{2.150}$$

2.5.2.3. Detecting a Coherent Carrier

Normally, data is always present on the signal. The consequence is that for successful tracking of the signal some sort of squaring is necessary to neutralize the sign of the data bit. In the case of the Costas discriminator this was done by multiplying with the quadra phase channel. For the code tracking the discriminator was simply squared. Coherent tracking of the code is in principle possible, but it requires knowledge of the sign of the data bit, which the data demodulation must provide. So obviously for a navigation system it would be nice if there were no data bit present at all.

This is just what the designers of the GPS modernization program have planned and the same is planned for the European Galileo system. In the GPS modernization terminology such a data less channel is called coherent carrier. The Europeans call it a pilot channel.

The most straightforward device for tracking a coherent carrier would be given by the following discriminator:

$$D_{\mathrm{PLL}} := q_p = \sum_{k=0}^{N} a_k \,\mathrm{sinc}(\Delta\omega_k T_p/2)\mathrm{Im}\left\{\mathrm{e}^{i\Delta\theta_k}\, R^*_{\mathrm{signal}}(\Delta\tau + \delta_k/c)\right\} \tag{2.151}$$

This is like the Costas discriminator without the I-channel. With noise the discriminator is written as

$$D_{\mathrm{PLL}} := q_p + \xi_p \tag{2.152}$$

Here there are no quadratic terms in the Gauß-ian noise. The variance and mean of eq. (2.152) at the tracking point is given by:

$$\mathrm{E}\left\{S_{\mathrm{PLL}}\right\} \approx a\Delta\theta \qquad \mathrm{Var}\left\{S_{\mathrm{PLL}}\right\}|_{\Delta\tau=0} = a \tag{2.153}$$

There are two very important differences between the Costas discriminator and the coherent carrier discriminator.

Firstly, the absence of the +1 in expression for the variance. The consequence is that there will be no squaring loss term for the phase noise.

Secondly, and maybe more importantly, the pull-in region of the coherent carrier discriminator is twice as large as the pull-in region of the Costas discriminator. This is a consequence of the fact that the Costas discriminator is proportional to $\sin(2\Delta\theta)$, while the coherent carrier discriminator is proportional to $\sin(\Delta\theta)$. So the coherent carrier discriminator will be less susceptible to cycle-slips and there will also be no half-cycle slips as with the Costas discriminator[18].

[18] In practice this half-cycle ambiguity is not a real problem; it is resolved by examining the data pre-ample from the data demodulation (see [Die96]).

2.5.3. Oscillator Error

When the receiver is tracking the signal, the instabilities of the reference oscillator will cause errors on the phase as well as on the code measurements. For the tracking loops this means that the dynamics induced by these instabilities must be tracked by the loops. Thus, it can be assumed that slowly varying oscillator effects, i.e. relative to the time constants of the loops, will be compensated for by the tracking loops just like a slow change in Doppler is removed from the range.

In order to incorporate the oscillator model into SNSS it must be in the form of a stochastic differential equation (SDE). As a starting point for the model development the one-sided spectral density of the fractional frequency fluctuation is assumed to be in the form [DMB84, IE02]:

$$S_y(\omega) =: S_y^{-2}(\omega) + S_y^{-1}(\omega) + S_y^0(\omega) := \frac{2\pi^2 h_{-2}}{\omega^2} + \frac{\pi h_{-1}}{\omega} + \frac{h_0}{2} \tag{2.154}$$

where y is defined as [All66, Eis97, DMB84]:

$$y := \frac{\delta f}{f} \tag{2.155}$$

and δf is the absolute frequency deviation and f is the nominal frequency of the oscillator. A frequency normal with the spectral characteristics as given above by eq. (2.154) can also be described in terms of the so-called Allan variance:

$${}^A\sigma_y^2(\tau) = \frac{h_0}{2\tau} + 2\ln 2 h_{-1} + \frac{2\pi^2}{3}\tau h_{-2} \tag{2.156}$$

where the Allan variance is defined as

$${}^A\sigma_y^2(\tau) := \frac{1}{2}E\left\{[y(t+\tau) - y(t)]^2\right\} = \frac{1}{2}E\left\{[\Delta y_t(\tau)]^2\right\} \tag{2.157}$$

From eq. (2.154) the spectral density of the phase, $S_\phi(\omega)$, and time, $S_x(\omega)$, fluctuations can be derived and are shown to be [DMB84, IE02]:

$$S_\phi(\omega) = (2\pi f_0)^2 S_x(\omega) = \frac{(2\pi f_0)^2}{\omega^2} S_y(\omega) = \frac{(2\pi f_0)^2}{\omega^2}\left(\frac{2\pi^2 h_{-2}}{\omega^2} + \frac{\pi h_{-1}}{\omega} + \frac{h_0}{2}\right) \tag{2.158}$$

A set of parameters h_0, h_{-1} and h_{-2} describing several classes of oscillators is given in table 2.1.

In order to integrate the oscillator model into SNSS, a stochastic differential equation must be found that generates a noise process, characterized by eq. (2.158). It can be shown [Kas95] that given a spectral density $S(\omega)$, the SDE defined by

$$X = H(\mathbf{p})\varepsilon(t) \qquad \text{where} \qquad S(\omega) =: H(i\omega)H(-i\omega) \tag{2.159}$$

Oscillator	White freq. noise (h_0)	Flicker (h_{-1})	Integrated freq. noise (h_{-2})
Standard quartz	$2\cdot 10^{-19}$ s	$7\cdot 10^{-21}$	$2\cdot 10^{-20}$ Hz
TCXO	$1\cdot 10^{-21}$ s	$1\cdot 10^{-20}$	$2\cdot 10^{-20}$ Hz
OCXO[1]	$8\cdot 10^{-20}$ s	$2\cdot 10^{-21}$	$4\cdot 10^{-23}$ Hz
OCXO[2]	$2.51\cdot 10^{-26}$ s	$2.51\cdot 10^{-23}$	$2.51\cdot 10^{-22}$ Hz
Rubidium[1]	$2\cdot 10^{-20}$ s	$7\cdot 10^{-24}$	$4\cdot 10^{-29}$ Hz
Rubidium[2]	$1\cdot 10^{-23}$ s	$1\cdot 10^{-22}$	$1.3\cdot 10^{-26}$ Hz
Cesium[1]	$1\cdot 10^{-19}$ s	$1\cdot 10^{-25}$	$2\cdot 10^{-32}$ Hz
Cesium[2]	$2\cdot 10^{-20}$ s	$7\cdot 10^{-23}$	$4\cdot 10^{-29}$ Hz

Table 2.1.: Parameters for the Allan variance of several oscillators

does in fact generate such a process. The function H is the transfer function of the system and ε is a Gauß-ian white noise term. Note that $H(\mathbf{p})$ is to be interpreted as an operator function. So, the goal is to generate the oscillator phase noise by injecting Gauß-ian white noise into a dynamic system.

Now we factor each of the terms in eq. (2.154) to obtain the transfer function $H(i\omega)$. We obtain:

$$\begin{aligned} H^{-2}(\mathbf{p}) &= \frac{\sqrt{2h_{-2}}\pi}{\mathbf{p}^2} \\ H^{-1}(\mathbf{p}) &= \frac{\sqrt{\pi h_{-1}}}{\mathbf{p}^{3/2}} \\ H^{0}(\mathbf{p}) &= \sqrt{\frac{h_0}{2}}\frac{1}{\mathbf{p}} \end{aligned} \tag{2.160}$$

The corresponding processes are called *random walk frequency noise* (the h_{-2} -term), *flicker frequency noise* (the h_{-1} -term) and *white frequency noise* (the h_0 -term). There is no problem generating the white and random walk frequency noise. However, the flicker term poses a major problem that cannot be solved satisfactorily in time domain. At least it is impossible to construct a stochastic differential equation with $H_{-1}(\mathbf{p})$ as a transfer function. This is because the transfer function is not a rational function of $\mathbf{p}$. This is a general problem and several approaches have been attempted to solve or circumvent this. One is due to Mandelbrot [Man71] and involves extending the definition of the differential operator to fractional derivatives (d^α/dt^α, where α is a rational number). The special case here was treated in [DMB84] and is based on obtaining meaningful expressions for the covariance matrix of a two state Kalman filter model. The most sensible way to generate the flicker noise (or in general colored noises with a spectral density of the form $1/f^\alpha$) is perhaps to perform the simulation in frequency domain and transform the result back to time domain in the end. This is done in [Kas95] and in this reference C-source code is given that generates such noise processes. Unfortunately, none of these approaches are suited for integration into the system of stochastic differential equation describing the receiver model.

The approach chosen here is to formally interpret the square-root in eq. (2.160) as a series and thus obtain a well-defined stochastic differential equation that approaches $H_{-1}(\mathbf{p})$ in some sense. This is justified by the fact that we are actually expanding the *denominator* of the spectral density. The starting point is the spectral density of the flicker noise. A rational *Ansatz* is used for the transfer function:

$$\begin{aligned}\frac{1}{S^{-1}(\omega)} &=: D^{-1}(i\omega)D^{-1}(-i\omega) = \omega^3 \\ &\approx D_3^{-1}(i\omega)D_3^{-1}(-i\omega) \\ &:= \left[a + b(i\omega) + c(i\omega)^2\right]\left[a + b(-i\omega) + c(-i\omega)^2\right] \\ &= a^2 + (b^2 - 2ac)\omega^2 + c^2\omega^4\end{aligned} \tag{2.161}$$

where the coefficients a, b and c are to be determined. Of course an *Ansatz* of higher order could be chosen, but for the purposes here the one above is considered to be sufficient.

It is very important when choosing the *ansatz* for the expansion that the coefficients in eq. (2.161) are real and that $D_3^{-1}(i\omega)D_3^{-1}(-i\omega)$ doesn't have any real roots in the interval $[0, \infty)$. If that were the case, the spectral density would diverge at that frequency. Subsequently, this would lead to a divergence of the Allan variance for sampling intervals corresponding to the singular frequency and finally to a general catastrophe in the overall receiver modeling. If the series is truncated at ω^n for $n = 3$ then D_4^{-1} does in fact have real roots in the positive half-axis. However, n = 2 is safe.

The coefficients in eq. (2.161) are chosen such that the spectral density is approximated at an angular frequency ω_0. This is done by imposing conditions on the derivatives of $D_3^{-1}(i\omega)D_3^{-1}(-i\omega)$ as follows:

$$\begin{aligned}\omega_0^3 &= D_{-1}^3(i\omega)D_{-1}^3(-i\omega)\Big|_{\omega_0} \\ 3\omega_0^2 &= \frac{\partial}{\partial\omega}D_{-1}^3(i\omega)D_{-1}^3(-i\omega)\Big|_{\omega_0} \\ 6\omega_0 &= \frac{\partial^2}{\partial\omega^2}D_{-1}^3(i\omega)D_{-1}^3(-i\omega)\Big|_{\omega_0}\end{aligned} \tag{2.162}$$

Solving the equations in eq. (2.162) for a, b and c we get

$$a = \frac{\omega_0^{3/2}}{2} \qquad b = \sqrt{\frac{\sqrt{3}}{2\omega_0}} \qquad c = \frac{\sqrt{3\omega_0}}{2} \tag{2.163}$$

The value of ω_0 can in principle be chosen arbitrarily. However, in the context here we would of course like to expand around the frequency where the flicker noise is dominant in eq. (2.154) or eq. (2.158) or where the ratio between the flicker term and the spectral density is maximal, i.e.:

$$\frac{d}{d\omega}\left[\frac{\left(\frac{\pi h_{-1}}{\omega}\right)}{S_y(\omega)}\right] = 0 \qquad \Rightarrow \qquad \omega_0 := 2\pi\sqrt{\frac{h_{-2}}{h_0}} \tag{2.164}$$

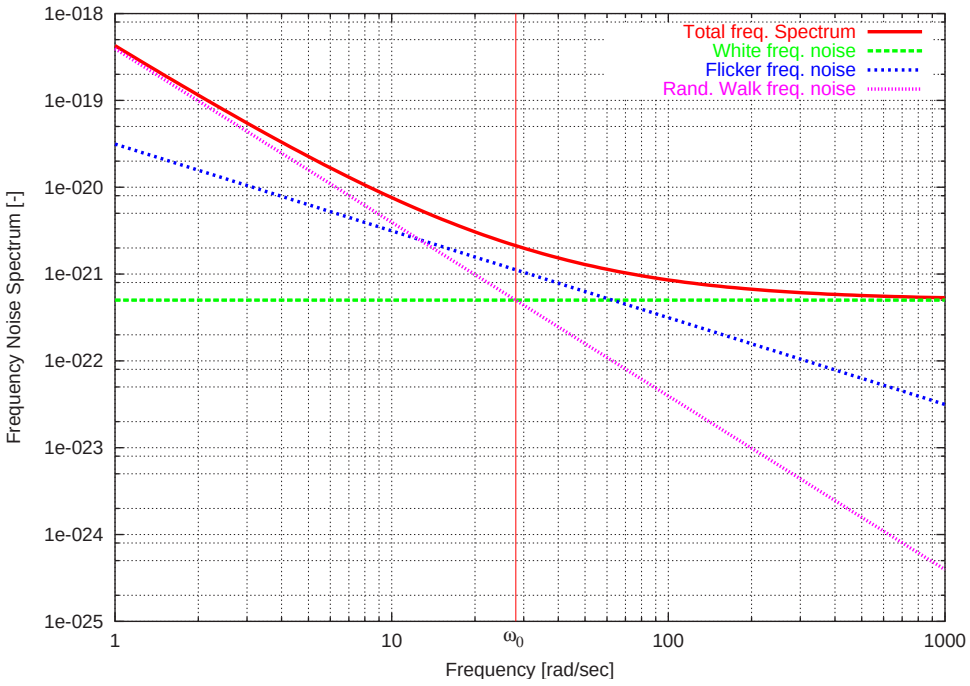

Figure 2.44.: Choice of ω_0. The individual terms of the spectral density (S^0, S^{-1} and S^{-2}) are shown as well as the total spectral density (S). The red, vertical line indicates the choice of ω_0. The graph was made using the parameters for the TCXO in table (2.1)

The choice of ω_0 is shown in figure (2.44). To the left of ω_0 the random walk frequency dominates and to the right the the white noise dominates.

Thus, the approximation of the spectral density can now be written as:

$$\begin{aligned} S_x(\omega) &\approx \tilde{S}_x(\omega) \\ &:= S_x^{-2}(\omega) + \pi h_{-1} \frac{4\omega_0}{3\omega^4 + \omega_0^4} + S_x^0(\omega) \end{aligned} \tag{2.165}$$

A comparison between the exact expression and the approximation of the frequency spectrum is shown in figure (2.45)

As to be expected a slight deviation from the exact expression is visible to the left and right of the frequency the denominator was expanded, ω_0. The quality of the approximation depends on the relationship between the parameters h_0, h_{-1} and h_{-2}. As can be seen in figure (2.45), the approximation is quite good for most oscillators considered here. For the OCXO2 the deviation is largest.

Now we can construct a model using eq. (2.159) and the expansion just developed. For the transfer function, $H^{-1}(\mathbf{p})$, we get:

$$H^{-1}(\mathbf{p}) = \frac{\sqrt{\pi h_{-1}}}{\mathbf{p}^{3/2}} \approx \frac{\sqrt{\pi h_{-1}}}{a + b\mathbf{p} + c\mathbf{p}^2} \tag{2.166}$$

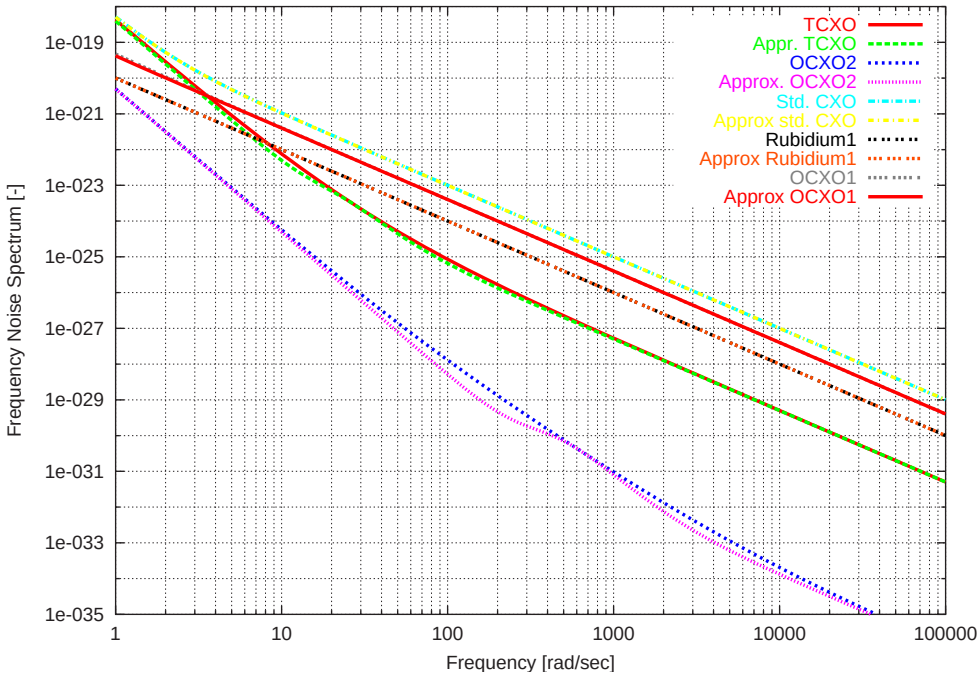

Figure 2.45.: Comparing the exact spectrum to the approximated one, for several of the oscillators in table (2.1).

where a, b and c are defined by eq. (2.163). Using this to construct a stochastic differential equation for the flicker noise of time:

$$\dot{\vec{X}} = \frac{d}{dt}\begin{bmatrix} X^1 \\ X^2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{\omega_0^2}{\sqrt{3}} & -\sqrt{\frac{2}{\sqrt{3}}}\omega_0 \end{bmatrix}\begin{bmatrix} X^1 \\ X^2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2}{\sqrt{3}}\sqrt{\pi h_{-1}\omega_0}\,\varepsilon_{-1} \end{bmatrix} \tag{2.167}$$

The time fluctuations of the white and random walk frequency fluctuations are readily described by the following equation:

$$\dot{\vec{X}} = \frac{d}{dt}\begin{bmatrix} X^1 \\ X^2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} X^1 \\ X^2 \end{bmatrix} + \begin{bmatrix} \sqrt{h_0/2}\,\varepsilon_0 \\ \pi\sqrt{2h_{-2}}\,\varepsilon_{-2} \end{bmatrix} \tag{2.168}$$

The noise terms ε_{-2} and ε_0 are independent (i.e. have correlation 0). The system of equations eq. (2.168) can now be augmented with the equations for the flicker term:

$$\dot{\vec{X}} = \frac{d}{dt}\begin{bmatrix} X^1 \\ X^2 \\ X^3 \\ X^4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{\omega_0^2}{\sqrt{3}} & -\sqrt{\frac{2}{\sqrt{3}}}\omega_0 \end{bmatrix}\begin{bmatrix} X^1 \\ X^2 \\ X^3 \\ X^4 \end{bmatrix} + \begin{bmatrix} \sqrt{h_0/2}\,\varepsilon_0 \\ \pi\sqrt{2h_{-2}}\,\varepsilon_{-2} \\ 0 \\ \frac{2}{\sqrt{3}}\sqrt{\pi h_{-1}\omega_0}\,\varepsilon_{-1} \end{bmatrix} \tag{2.169}$$

All the Gauß-ian noise processes are independent and have variance 1. The total time fluctuation is then given by the sum $X^1 + X^3$, where X^1 describes the white and random walk frequency noise and the process X^3 describes the flicker frequency noise.

In figure (2.46) and figure (2.47) a realization of two oscillators in table 2.1 are shown. The four plots are all of the same representation of the stochastic process defined by eq. (2.169), but the time scale varies.

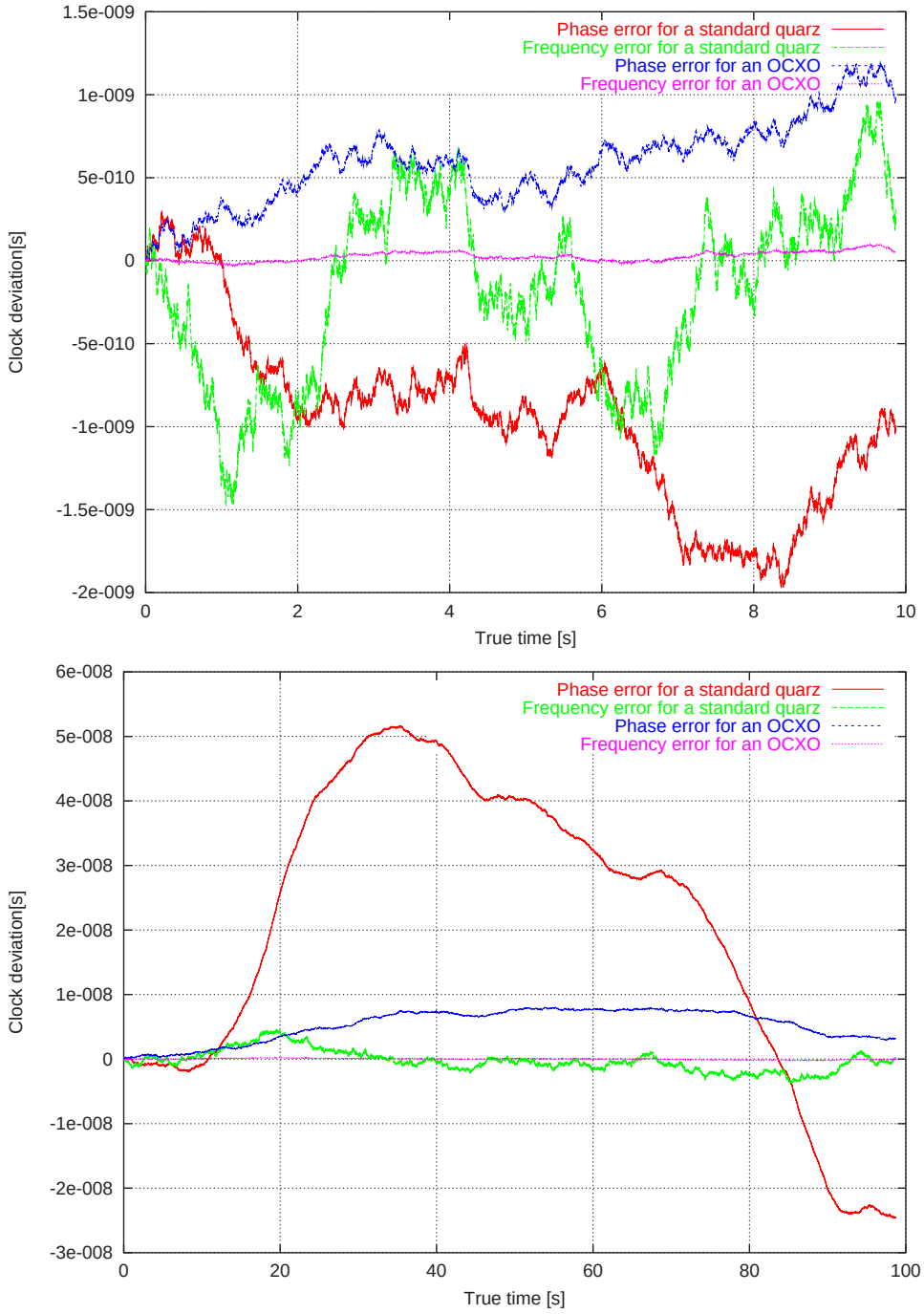

Figure 2.46.: Phase- and frequency error of the first two clock models. The curves were obtained using the four-state model described by eq. (2.169).

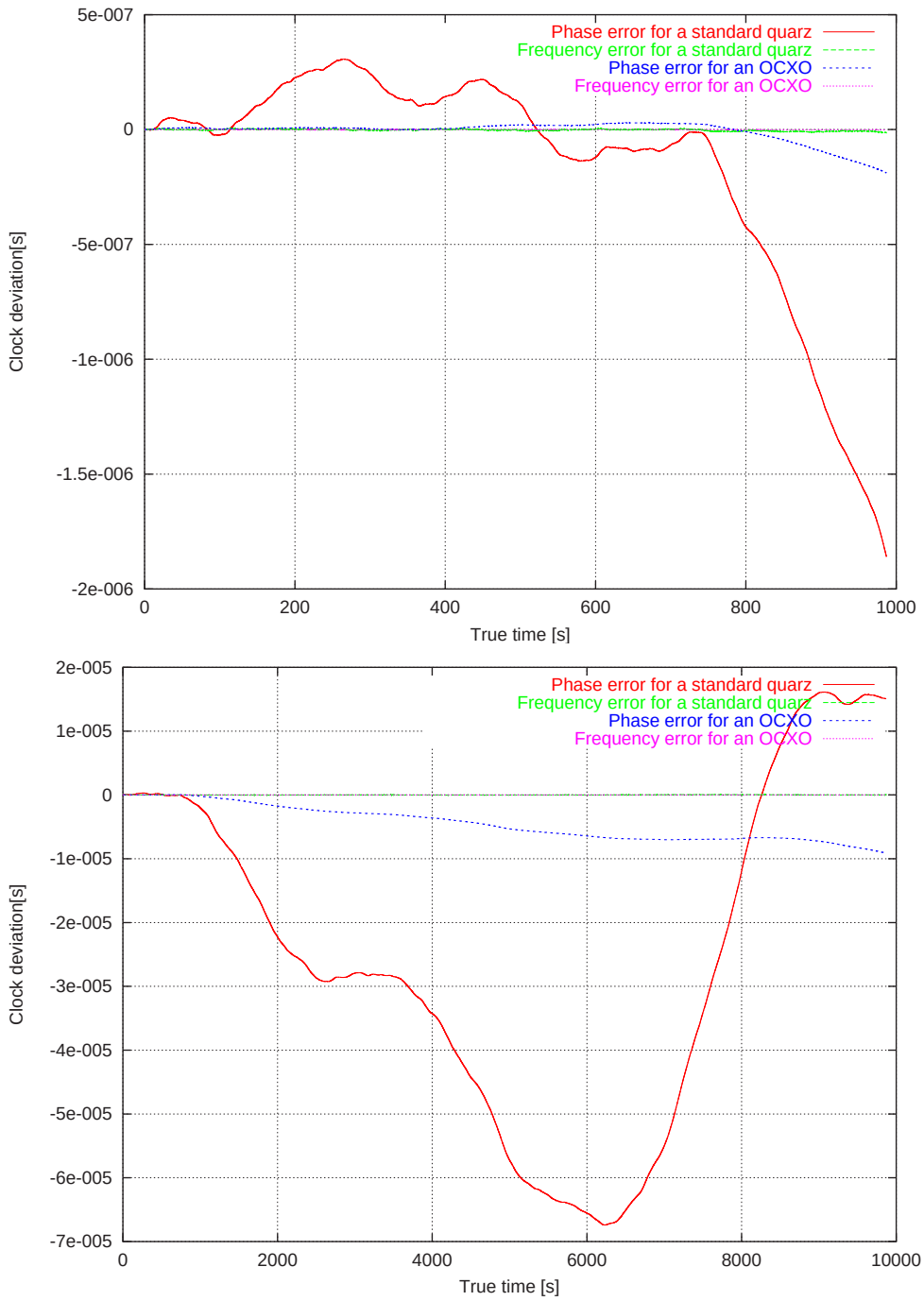

Figure 2.47.: Phase- and frequency error of the first two clock models. The curves were obtained using the four-state model described by eq. (2.169).

To consolidate the model a small program was written that calculates the Allan variance

of the realizations shown in figure (2.46) and figure (2.47) according to eq. (2.156) with

$$\begin{aligned}\Delta y_t(t') &:= \frac{\delta\tau(t) - \delta\tau(t-t')}{t'} - \frac{\delta\tau(t-t') - \delta\tau(t-2t')}{t'} \\ &= \frac{\delta\tau(t) - 2\delta\tau(t-t') + \delta\tau(t-2t')}{t'}\end{aligned} \tag{2.170}$$

where $\delta\tau := X^1 + X^3$ is a representation of the solution of the SDE eq. (2.169).

In figure (2.48) the Allan variances for six oscillators listed in table (2.1) are shown. The plot is a logarithmic plot on bothe axis and there are many orders of magnitude separating the performance curves of the six cases. Typically, the curves have three regions: For low values of τ the slope is constant negative. This is dominated by the constant h_0. The transition region where the slope changes from negative to positive is mainly controlled by the flicker frequency noise, h_{-1}. The random walk frequency noise, h_{-2} is dominant in the region of positive, constant slope.

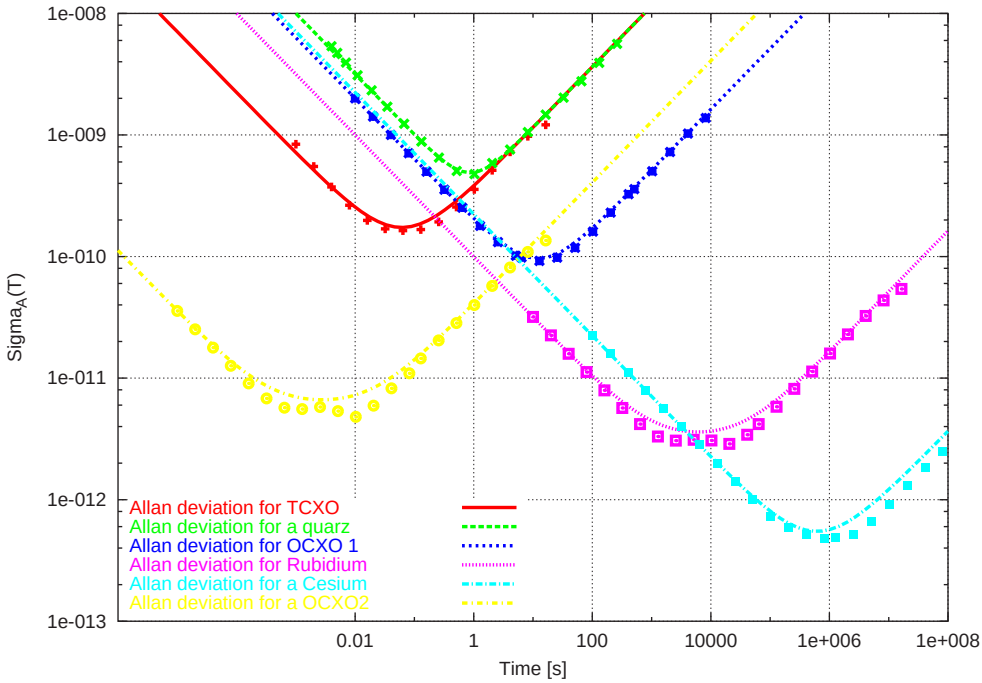

Figure 2.48.: Square root of the Allan variance given in eq. (2.156). The data points in the figure are results from a simulation based on eq. (2.169).

The transition region where the the flicker term, h_{-1} is dominant, the model only generates an approximation of the flicker spectrum. This manifests itself in small anomalies in the transition region for some of the curves. For the Rubidium and the OCXO 2 the departure from the predicted Allan deviations is largest and it is seen that the model underestimates the Allan deviation slightly in this region. This is consistent with the construction of the model as the approximation of $S^3_{-1}(\omega)$ smaller than $S_{-1}(\omega)$ (see figure (2.45)). For

the oscillators considered here the approximation of the flicker noise seems sufficient. However, for oscillators where the flicker region is more flat, the modeling errors may become too large. This may be solved by including more terms in the expansion of the flicker noise.

From a receiver point-of-view, the long term stability is not of primary importance[19]. Only effects in the order of seconds are of importance, because the slower effects can by compensated by the tracking loops.

2.5.4. Closed Loop Tracking

The final stage in the signal processing is the tracking of the zero of the discriminator functions. This is a control theoretical problem. A schematic tracking loop is shown in figure (2.49). The approach followed here is similar to the formulation in [Eis97].

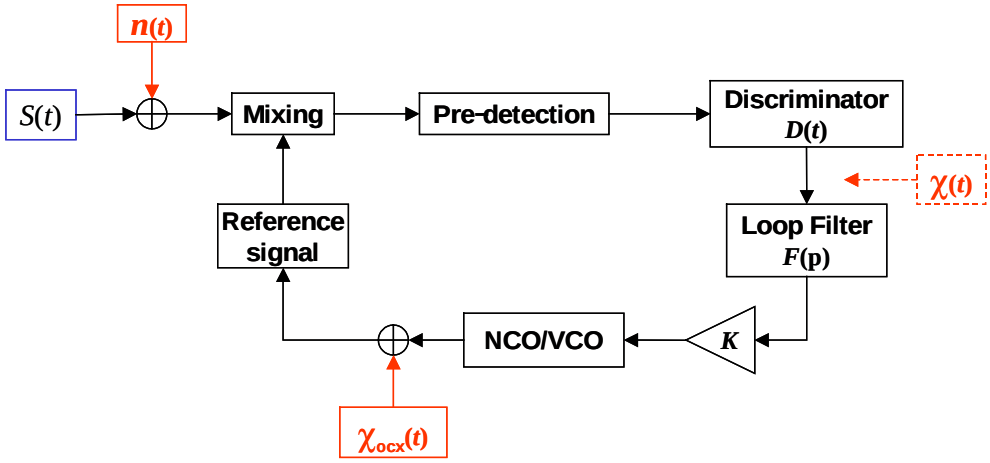

Figure 2.49.: Schematic representation of a tracking loop. See text for explanations

The noise processes (Gauß-ian white noise, $n(t)$ and oscillator imperfections, χ_{OCX}) are shown in red. The noise at the input ($n(t)$) is integrated during the correlation process. The noise after the integration process is indicated by the red, dashed box ($\chi(t)$). This process was analyzed theoretically in detail in the previous sections. The form of the oscillator noise was analyzed in section 2.5.3.

The received signal is multiplied with the reference signal („mixing"). Then it goes into the pre-detection part. This part includes the separation into Q- and I- channels, as well as the generation and integration of the early, late and punctual samples. After the pre-detection the discriminator is generated and then the signal enters the loop filter ($F(p)$).

[19] In the case of directly acquiring extremely long codes, such as the GPS P-code, the long term stability may be an issue.

The output of the loop filter is multiplied with a gain factor (K) and this controls the *frequency* of the NCO/VCO which then again controls the frequency of the code or frequency generator. Because the output of the loop filter is basically proportional to the *phase*, the NCO/VCO acts as an integrator. This holds for the code tracking loop as well as for the carrier phase tracking loop.

Denoting by $x(t)$ the phase of the reference signal we can write down a differential equation for the figure (2.49) in operator form where the differential operator is defined as $\mathbf{p} := d/dt$

$$\mathbf{p}x(t) = KF(\mathbf{p})D_x(t) \tag{2.171}$$

The discriminator $D_x(t)$ is in general a non-linear function of time and the variable $x(t)$ and corresponds to the discriminator functions analyzed in the previous sections.

If the noise processes indicated in figure (2.49) are considered, i.e. the Gauß-ian thermal noise and the imperfections of the oscillator, then eq. (2.171) is augmented by the two terms: $\chi(t)$ and $\chi_{\text{ocx}}(t)$ as follows (see [LS73, Lin72, Spi73]):

$$\mathbf{p}x(t) = KF(\mathbf{p})\left[D_x(t) + \chi(t)\right] + \mathbf{p}\chi_{\text{ocx}}(t) \tag{2.172}$$

The fundamental difference between eq. (2.171) and eq. (2.172) is that the first is a non-linear ordinary differential equation (ODE) but the second is a non-linear *stochastic* differential equation. This has consequences, which will be addressed shortly.

Note the appearance of the noise terms in eq. (2.172). The thermal noise term, $\chi(t)$ is operated upon by the loop filter, $F(p)$ but, the oscillator noise appears as a driving noise term on the right side of eq. (2.172).

The model is formulated in the time-domain and the function F is therefore formally to be understood as an operator function. F controls the order of the differential equation in eq. (2.171) and eq. (2.172) and the following table gives the form of F for the first three orders of eq. (2.171) and eq. (2.172).

Order of loop	Loop filter ($KF(\mathbf{p})$)	Loop bandwidth (B_L)
first order	K	$\frac{K}{4}$
second order	$K(1+\frac{\alpha}{\mathbf{p}})$	$\frac{K+\alpha}{4} = \frac{\omega_L}{8\xi}(1+4\xi^2)$
third order	$K(1+\frac{\alpha}{\mathbf{p}}+\frac{\beta}{\mathbf{p}^2})$	$\frac{K(\alpha K+\alpha^2-\beta)}{4(\alpha K-\beta)}$

Table 2.2.: Loop filters for the tracking loop in eq. (2.171)

In the following sections we will construct the tracking loops for the code (delay lock loop) and the phase (phase locked loop). But before we do that we will demonstrate how the results from previous sections are combined to obtain the stochastic differential equation described by eq. (2.171). The pre-detection process is basically a correlation of the received signal with the reference signal. In the process the noise is also integrated

over a time interval T_p and thus, unless it is normalized, its variance will be proportional to the integration interval. The discrete version of eq. (2.171) for a first order, linearized loop can be written as

$$\begin{aligned} \frac{x_{i+1} - x_i}{\Delta t} &= -4B_L \left(x_i - A\eta(t)\right) \\ \Rightarrow & \\ x_{i+1} &= \left(1 - 4B_L\Delta t\right) x_i + 4B_L\Delta t A\eta(t) \end{aligned} \tag{2.173}$$

where Δt is the update time interval, η is a Gauß-ian white noise process with variance 1 and A represents the standard deviation of the noise process. In a real GPS receiver the update interval is set to $\Delta t = T_p$. This is the form obtained in [Die96].

In this thesis, however, we want to formulate the model as a stochastic differential equation, which degenerates to a continuous, ordinary differential equation in the absence of the noise term. Thus we need to perform the limit $\Delta t \to 0$. In the noise-free case this is trivial and leads to a first order differential equation. In the noisy case this is more involved because the Gauß-ian noise process is discontinuous everywhere, and can neither by differentiated nor integrated using classical calculus. Still, using the Îto calculus eq. (2.173) can be given a mathematical meaning [KP95, Lin72, KPS97]. Consider the equation

$$\frac{dy}{dt} = a(t,y)y + b(t,y)\eta(t) \tag{2.174}$$

Writing this formally as a differential (multiplying with dt and re-arranging) we get

$$\begin{aligned} dy &= a(t,y)ydt + b(t,y)\eta(t)dt \\ dy &= a(t,y)ydt + b(t,y)dW \end{aligned} \tag{2.175}$$

The process η is Gauß-ian with zero mean and variance 1. The differential dW corresponds to the Gauß-ian process and its integral, W, is the well-known Wiener process or a random walk. So, very loosely speaking the integral of the Gauß-ian process is the Wiener process. The theory of SDEs will not be pursued further here and the interested reader is referred to [KP95, KPS97, Lin72].

Returning to our problem at hand, i.e. eq. (2.173) and remembering that the noise process originates from the correlation and its variance thus proportional to T_p, a discrete GNSS receiver would have an update equation in the form:

$$\Delta x_i = -4B_L T_p x_i + 4B_L A T_p \eta(t) \tag{2.176}$$

The factor $T_p\eta(t)$ in the last term is essentially a Gauß process with variance T_p^2. However, re-formulating the equation to a SDE we want to let $T_p \to 0$, but still maintain the noise properties dictated by eq. (2.176). We want to interpret eq. (2.176) in differential form and perform a transition to a continuous, stochastic differential equation. Thus we must identify the the Wiener increment in eq. (2.176). As the time step is T_p, the Wiener

increment must have variance T_p and we therefore let $\Delta W = W_{i+1} - W_i = \sqrt{T_p}\eta(t)$. We thus write

$$\Delta x_i = -4B_L T_p x_i + 4B_L A\sqrt{T_p}\Delta W_i \tag{2.177}$$

Now we can let $T_p \to 0$ (except for the $\sqrt{T_p}$ factor). The equation above is then

$$\Delta x_i = -4B_L x_i \Delta t + 4B_L A\sqrt{T_p}\Delta W_i \tag{2.178}$$

or in the limit $\Delta t \to 0$

$$dx = -4B_L x dt + 4B_L A\sqrt{T_p}dW \tag{2.179}$$

The last equation is a continuous description of the digital tracking loop with an update interval of T_p. For these reasons the noise terms in the following must be scaled with the factor $\sqrt{T_p}$, which basically controls the integration interval of the tracking loop. We now have a continuous description of the tracking loops and we are free to choose any numerical algorithm to solve the SDEs.

Loop bandwidth The most important parameter of the tracking loop is the so-called loop filter bandwidth (one sided). It is defined as

$$B_L := \frac{1}{2\pi}\int_0^\infty |H(i\omega)|^2 d\omega \tag{2.180}$$

The loop filter bandwidth for a first, second and third order tracking loop is shown in table 2.2.

Transient Response The transient behavior of a tracking loop refers to how the loop responds to dynamics in the signal. This dynamic behavior can have several reasons. The most obvious is the temporal dependence of the line-of-sight, already discussed in section (2.1.3). Another source for such errors is the frequency instability of the local oscillator.

The transient tracking error of a tracking loop of order n can be shown to be as follows (see e.g. [Spi73, Eis97]).

$$\begin{aligned} e(\infty) &= \frac{\dot{x}}{K} && \text{for a first order loop} \\ e(\infty) &= \frac{\ddot{x}}{K\alpha} && \text{for a second order loop} \\ e(\infty) &= \frac{\dddot{x}}{K\beta} && \text{for a third order loop} \end{aligned} \tag{2.181}$$

For the DLL and PLL we get

$$\begin{aligned} e_{dll}(\infty) &= \frac{\dot{s}}{4B_L^{dll}} && \text{for an unaided DLL} \\ e_{pll}(\infty) &= \frac{\ddot{s}}{\omega_L^2} && \text{for the second order PLL} \\ e(\infty) &= \frac{\dddot{s}}{4B_L^{dll}\omega_L^2} && \text{for the carrier aided DLL} \end{aligned} \tag{2.182}$$

These expressions describe the performance of the tracking loops under dynamic situations. We see that a loop of order n has a transient error proportional to the n-th time derivative of the line-of-sight.

Impact of Oscillator Instabilities In [Lin72] it is shown that the RMS noise caused by the imperfections of the oscillator is expressed as

$$\sigma_\phi^2 = \frac{1}{2\pi}\int_0^\infty S_\phi(\omega)|1 - H(\omega)|^2 d\omega \tag{2.183}$$

where $S_\phi(\omega)$ is the spectral density of the oscillator noise and $H(\omega)$ is the closed loop transfer function.

For a second order loop the variance of the oscillator error in eq. (2.183) can be evaluated to[IE02]

$$\sigma_\phi^2 = \frac{\omega_0}{2}\left[\frac{\pi^2 h_{-2}}{\sqrt{2}\omega_L^3} + \frac{\pi h_{-1}}{4\omega_L^2} + \frac{h_0}{4\sqrt{2}\omega_L}\right] \tag{2.184}$$

where ω_0 is the carrier frequency. The units here are [rad^2]. It is interesting to see how the total noise is distributed among the three terms of the oscillator noise. In figure (2.50) this is done for the two oscillators standard CXO and TCXO. In figure (2.51) the corresponding curves for $OCXO_1$ and $OCXO_2$ are shown.

For the standard CXO and the $OCXO_1$ the main contribution comes from the white frequency noise term, h_0 and for the TCXO the fluctuation in phase is dominated by the flicker frequency term, h_{-1}. The $OCXO_2$ is dominated by the random walk frequency noise term.

As eq. (2.184) is a standard deviation, no explicit reference to the stochastic process behind the sigma value is made, in other words it is possible for eq. (2.184) to yield the same σ for two different oscillators, although their properties may be quite different. For example when carrier aiding is being applied, noise from the phase-locked loop (PLL) is fed into the delay lock loop (DLL). The noise as it appears in the DLL depends strongly on the form of the oscillator noise in the PLL, i.e. which term is the dominating one in eq. (2.184). This issue will be discussed in the chapter 4.

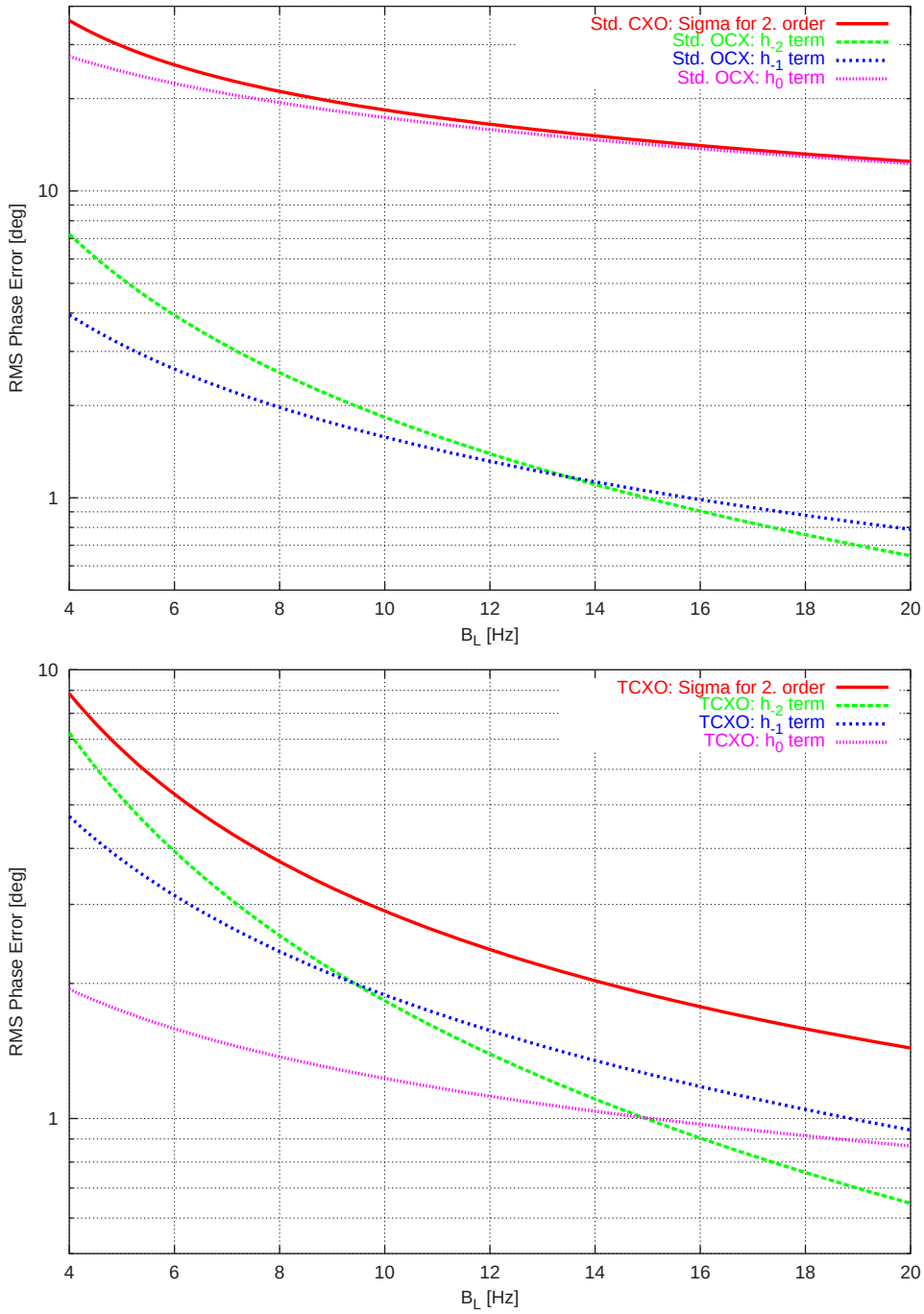

Figure 2.50.: Phase error for a second order loop, due to oscillator instabilities only for the standard CXO and the $OCXO_1$

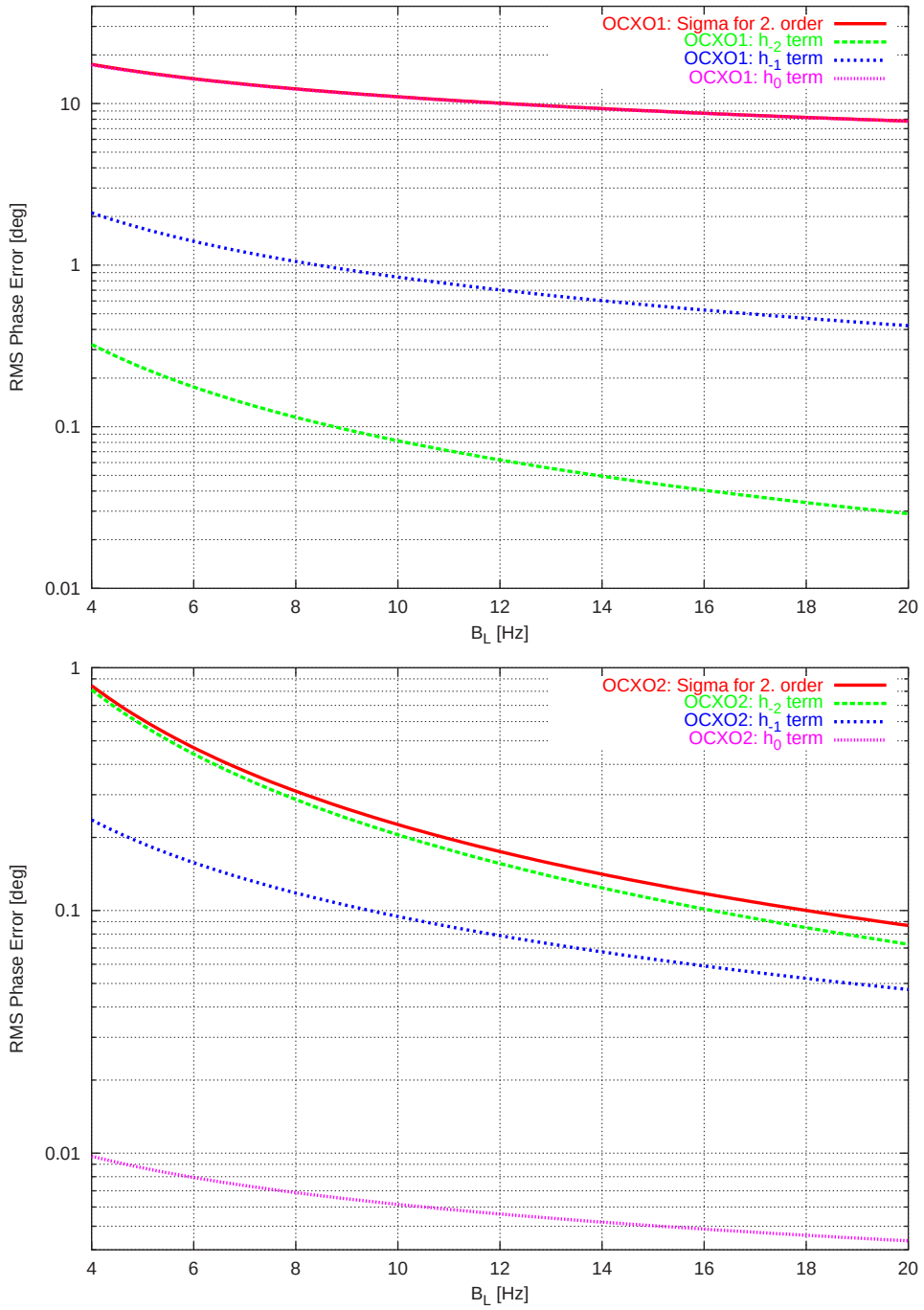

Figure 2.51.: Phase error for a second order loop, due to oscillator instabilities only for $OCXO_1$ and $OCXO_2$

2.5.4.1. Delay Lock Loop

The device responsible for the tracking of the code is called a delay lock loop (DLL). Basically, the phase of the code sequence is tracked using a device with the mathematical structure of eq. (2.171).

For the non-coherent early-late discriminator in eq. (2.141) and assuming a first order loop we get

$$\mathbf{p}\Delta\tau = \frac{4B_L^{dll}}{D'(0)}\left[D_{dll}(\tau) + \chi^{\tau}_{T_p}\right] + \mathbf{p}\chi^{\tau}_{\text{nco}} - \dot{\tau}_0(t) \tag{2.185}$$

The normalization of the discriminator is needed to make the right hand side proportional to $\Delta\tau(t) = \tau(t) - \tau_0(t)$ for small values of $\Delta\tau(t)$. In this normalization B_L^{dll} has the meaning of loop-bandwidth discussed above. The discriminator gain, $D'(0)$, depends on the form of the discriminator, i.e. coherent, non-coherent DLL. It also depends on parameters of the discriminator like correlator spacing. The front-end filter also has an impact on the discriminator gain [DPHR99, Bet00b, BF00, HRL98]. In the theory here the effect of band limiting on the discriminator gain is accounted for in the shape of the auto-correlation function.

The loop bandwidth B_L^{dll} is usually set to be in the range 0.05 to 1 Hz for modern GPS receivers, depending on the application. For static applications, where the signal dynamics are low, the loop bandwidth can be reduced. This effectively filters out noise. However, the transient behavior is slowed down and the loop cannot react to violent changes in signal dynamics. Thus, for dynamic applications, the loop bandwidth must be increased, but that "lets the noise in". The relation between the loop bandwidth B_L^{dll} and the factor K is shown in table (2.2).

The term $\chi^{\tau}_{T_p} = \sqrt{T_p}\delta S_\tau$ is an additive Gauß-ian noise term. It has mean zero and the variance of δS_τ is as calculated in eq. (2.142). It should be emphasized once more that this is a simplification from the original form eq. (2.141). It is important to realize that eq. (2.185) is a non-linear stochastic differential equation and even though it has been simplified to an equation with an additive Gauß-ian white noise, it can not simply be treated as a normal continuous differential equation. This is because the noise term is not integrable in the sense of a Riemann-Stieltjes or Lebesgue integral. To give eq. (2.141) a mathematical meaning the so-called Îto calculus was applied. The theory of stochastic integration and stochastic differential equations is rather involved and we are mostly interested in solving the equation above numerically. However, the main consequence of defining a measure such that the Gauß-ian process is integrable, is the fact that the familiar chain rule of continuous calculus is supplemented by an extra term [KP95, KPS97]. For the numerics this means that it is *not* possible to take any old run-of-the-mill Runge-Kutta scheme and just add some noise to it.

In this theses eq. (2.185) is solved both in the noise-free case (then using an implicit fifth order Runge-Kutta scheme) and in the noisy case (with a stochastic, implicit Euler-scheme).

2.5.4.2. Phase-Locked Loop

The phase-locked loop (PLL) tracks the carrier phase. Usually, it is implemented as a second order loop. Using the Costas discriminator eq. (2.149), $F(\mathbf{p})$ for the second order in table (2.2) and inserting into eq. (2.171) we obtain

$$\mathbf{p}\Delta\theta = \frac{K}{D'_{pll}(0)}(1+\frac{\alpha}{\mathbf{p}})(D_{pll}(t,\theta,\tau)+\chi^{\theta}_{T_p}) + \mathbf{p}\chi^{\theta}_{\text{nco}} - \mathbf{p}\theta_0 \tag{2.186}$$

with

$$\sqrt{T_p}\delta S_\theta = \chi^{\theta}_{T_p} \tag{2.187}$$

Here, as in the case of the DLL, we assume additive Gauß-ian noise. δS_θ is Gauß-ian with mean zero and variance given by eq. (2.150). One could now re-arrange eq. (2.186) to yield a second order differential equation. That would involve taking the time derivative of the discriminator $D_{pll}(t,\theta,\tau)$ and what is more problematic, the time derivative of the Gauß-ian white noise term. This is the approach pursued in [Eis97]. For our purposes we choose to write eq. (2.186) as a system of two first order equations:

$$\begin{aligned}\mathbf{p}\Delta\theta &= \frac{K}{D'_{pll}(0)}(D_{pll}(t,\theta,\tau)+\chi^{\theta}_{T_p}) + \mathbf{p}\chi^{\theta}_{\text{nco}} + \psi \\ \psi &= \frac{K\alpha}{D'_{pll}(0)\mathbf{p}}(D_{pll}(t,\theta,\tau)+\chi^{\theta}_{T_p}) - \mathbf{p}\theta_0\end{aligned} \tag{2.188}$$

re-arranging and substituting $\mathbf{p} \to \frac{d}{dt}$ we get

$$\begin{aligned}\dot{\Delta\theta} &= \frac{K}{D'_{pll}(0)}\left(D_{pll}(t,\theta,\tau)+\chi^{\theta}_{T_p}\right) + \dot{\chi}^{\theta}_{\text{nco}} + \psi \\ \dot{\psi} &= \frac{K\alpha}{D'_{pll}(0)}\left(D_{pll}(t,\theta,\tau)+\chi^{\theta}_{T_p}\right) - \ddot{\theta}_0\end{aligned} \tag{2.189}$$

In eq. (2.189) there are two constants K and α that need to be determined. In analogy to the harmonic oscillator these constants are written in terms of the resonance frequency ω_L and the damping factor ξ

$$K = 2\xi\omega_L = 4B^{pll}_L - \alpha \qquad \text{and} \qquad \alpha = \frac{\omega_L}{2\xi} \tag{2.190}$$

or

$$\omega_L = \sqrt{K\alpha} = \frac{8\xi B^{pll}_L}{4\xi^2+1} \qquad \text{and} \qquad \xi = \sqrt{\frac{K}{4\alpha}} \tag{2.191}$$

where the relations for B^{pll}_L were taken from table (2.2). For most technical application the damping is chosen as $\xi = \frac{1}{\sqrt{2}}$. The loop filter bandwidth B^{pll}_L is typically in the order of 20 Hz. As in the case of the DLL the exact value of the loop filter bandwidth (B^{dll}_L) depends on the actual application. For dynamic applications the loop filter bandwidth must be increased relative to static applications.

Inserting eq. (2.191) into eq. (2.186) and setting $\xi = \frac{1}{\sqrt{2}}$, the PLL then becomes

$$\begin{aligned} \dot{\Delta\theta} &= \frac{2\xi\omega_L}{D'_{pll}(0)} \left[D_{pll}(t,\theta,\tau) + \chi^{\theta}_{T_p} \right] + \dot{\chi}^{\theta}_{\text{nco}} + \psi \\ \dot{\psi} &= \frac{\omega_L^2}{D'_{pll}(0)} \left[D_{pll}(t,\theta,\tau) + \chi^{\theta}_{T_p} \right] - \ddot{\theta}_0 \end{aligned} \tag{2.192}$$

2.5.4.3. Carrier-Aided DLL; Coupling the DLL and the PLL

The loop bandwidth of the DLL is only about 1 Hz and besides it is only of first order and as we saw in the previous section the transient response depends on the loop order. Thus it is difficult to track a signal in the DLL with high dynamics on it. The code tracking loop can however be aided by feeding the first derivative of the phase from the phase locked loop into the DLL. The phase of the signal is only known to within an integer ambiguous cycle, but the first derivative does, of course, not depend on this ambiguity. Thus the aided code tracking loop becomes

$$\mathbf{p}\Delta\tau = \frac{4B_L^{\text{dll}}}{D'_{dll}(0)} \left[D_{dll}(\tau) + \chi^{\tau}_{T_p} \right] + \frac{1}{\omega}\mathbf{p}\theta + \dot{\chi}^{\tau}_{\text{nco}} - \mathbf{p}\tau_0 \tag{2.193}$$

here θ is the solution of the stochastic differential equation in eq. (2.188) (i.e. a stochastic process). Thus eq. (2.193) effectively comprises a system is non-linear stochastic differential equations. Although a number of assumptions and simplifications were made to derive eq. (2.193), it by far cannot be solved analytically. However, there are effective algorithms to solve such systems of (stochastic) differential equations.

2.5.5. DLL and PLL; A system of SDE

The equation eq. (2.193) describes the tracking loop behavior for both DLL and PLL. Basically, these are two coupled stochastic differential equations; the DLL of order 1 and the PLL of order 2. The PLL was already re-arranged into two 1. order SDEs. Before implementing them in a numerical scheme, we will formulate eq. (2.193) as with the PLL as a system of first order stochastic differential equations. We distinguish between the noise-free, the noisy case and the case with noise and imperfect oscillator.

The Dynamic Case Defining the vector $\vec{X}$ as

$$\vec{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} := \begin{bmatrix} \Delta\tau \\ \Delta\theta \\ \psi \end{bmatrix} \tag{2.194}$$

Inserting eq. (2.192) and eq. (2.185) into the definition above and ignoring the noise terms we then have

$$\dot{\vec{X}} = \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \end{bmatrix} = \begin{bmatrix} \dot{\Delta\tau} \\ \dot{\Delta\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{4B_L^{\text{dll}}}{D'_{dll}(0)} D_{dll}(t, X_1) + \frac{2\xi\omega_L}{\omega D'_{pll}(0)}(D_{pll}(t, X_1, X_2)) + \frac{\psi}{\omega} - \dot{\Delta\tau}_0 \\ \frac{2\xi\omega_L}{D'_{pll}(0)} D_{pll}(t, X_1, X_2) + \psi \\ \frac{\omega_L^2}{D'_{pll}(0)} D_{pll}(t, X_1, X_2) - \ddot{\Delta\theta}_0 \end{bmatrix} \tag{2.195}$$

When carrier aiding is off (i.e. the second and third term in the third coordinate on the right hand side of eq. (2.195) is not present) the coordinate X_1 is not influenced by X_2 and X_3. This is because the non-coherent discriminator for the DLL (D_{dll}) is independent of the phase.

The Dynamic Case with Noise Generalizing eq. (2.195) for the noisy case is now trivial. The noise terms in eq. (2.192) and eq. (2.185) simply have to be inserted at the right places:

$$\dot{\vec{X}} = \begin{bmatrix} \frac{4B_L^{\text{dll}}}{D'_{dll}(0)} D_{dll}(t, X_1) + \frac{2\xi\omega_L}{\omega D'_{pll}(0)}(D_{pll}(t, X_1, X_2)) + \frac{X_3}{\omega} \\ \frac{2\xi\omega_L}{D'_{pll}(0)} D_{pll}(t, X_1, X_2) + X_3 \\ \frac{\omega_L^2}{D'_{pll}(0)} D_{pll}(t, X_1, X_2) \end{bmatrix} + \begin{bmatrix} \frac{4B_L^{dll}}{D'_{dll}(0)} \chi_{T_p}^{\tau} \\ \frac{2\xi\omega_L}{D'_{pll}(0)} \chi_{T_p}^{\theta} \\ \frac{\omega_L^2}{D'_{pll}(0)} \chi_{T_p}^{\theta} \end{bmatrix} \tag{2.196}$$

The Dynamic Case with Noise and Oscillator Imperfections To expand the model to include imperfect oscillator eq. (2.196) is augmented using the results from section 2.5.3. Before we proceed we notice that the the relevant entity is not the time fluctuation itself, but rather the first derivative ($\dot{\chi}_{\text{nco}}$). Thus, we are interested in $\dot{X}_1 + \dot{X}_3$ and not $X_1 + X_3$ from eq. (2.169). To achieve this eq. (2.196) is augmented as follows:

$$\begin{aligned} \dot{\vec{X}} = & \begin{bmatrix} \frac{4B_L^{\text{dll}}}{D'_{dll}(0)} D_{dll}(t, X_1) + \frac{2\xi\omega_L}{\omega D'_{pll}(0)}(D_{pll}(t, X_1, X_2)) + \frac{X_3}{\omega} \\ \frac{2\xi\omega_L}{D'_{pll}(0)} D_{pll}(t, X_1, X_2) + X_3 \\ \frac{\omega_L^2}{D'_{pll}(0)} D_{pll}(t, X_1, X_2) \end{bmatrix} + \begin{bmatrix} \frac{4B_L^{dll}}{D'_{dll}(0)} \chi_{T_p}^{\tau} \\ \frac{2\xi\omega_L}{D'_{pll}(0)} \chi_{T_p}^{\theta} \\ \frac{\omega_L^2}{D'_{pll}(0)} \chi_{T_p}^{\theta} \end{bmatrix} \\ & + \begin{bmatrix} Y_\tau^1 + Y_\tau^3 + \sqrt{h_0/2}\,\varepsilon_{0,\tau} \\ 2\pi\omega(Y_\theta^1 + Y_\theta^3 + \sqrt{h_0/2}\,\varepsilon_{0,\theta}) \\ 0 \end{bmatrix} \end{aligned} \tag{2.197}$$

where Y_i^1 and Y_i^3 are solutions to the following system:

$$\frac{d}{dt} \begin{bmatrix} Y_i^1 \\ Y_i^2 \\ Y_i^3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{\omega_0^2}{\sqrt{3}} & -\sqrt{\frac{2}{\sqrt{3}}}\omega_0 \end{bmatrix} \begin{bmatrix} Y_i^1 \\ Y_i^2 \\ Y_i^3 \end{bmatrix} + \begin{bmatrix} \pi\sqrt{2h_{-2}}\,\varepsilon_{-2,i} \\ 0 \\ \frac{2}{\sqrt{3}}\sqrt{\pi h_{-1}\omega_0}\,\varepsilon_{-1,i} \end{bmatrix} \tag{2.198}$$

These are actually two systems of equations; one for the code phase NCO ($i = \tau$) and one for the carrier phase oscillator ($i = \theta$). Comparing eq. (2.198) to the eq. (2.169) we notice that Y yields the needed derivatives of X. The dimension of the system of equations is reduced by one.

Note that the NCO part (i.e. Y_τ^1, Y_τ^3, Y_θ^1 and Y_τ^3) does not depend on the rest, so the oscillator simulation can be formulated separately. However, the time parameter must be the same in all three equations.

The equation above encompasses the most general form of the receiver model that will be used in this thesis and is implemented in SNSS.

2.6. Multipath Mitigation Techniques

There are many ways to attack the multipath problem. One is to use a choke-ring antenna, that blocks out reflections coming from a low elevation angle. An enhanced version of this method is to dynamically steer the antenna gain such that a high gain is only achieved in the direction of the satellite. Such devices are known as phased array antennas. These techniques are very difficult to implement and not state-of-the-art in civilian receivers yet.

Another approach to reduce the multipath effect is to use modified discriminators and/or multiple correlators in the receiver channel.

2.6.1. Narrow Correlator

The most straightforward multipath mitigation method is the so-called narrow correlator technique, which is achieved by simply decreasing the correlator spacing, i.e. setting $d < 1$.

This technique is implicit in the formulation of the models above and thus the case of narrow correlating has already been covered.

There are also more elaborate techniques and some of them will be treated in the following.

2.6.2. Double Delta Correlator

The double delta discriminator is basically an attempt to approximate the second derivative of the correlation function, instead of the first and use that as a discriminator.

The coherent double delta discriminator is given by:

$$D_{dd}(\tau) := 2(i_e - i_l) - (i_{ee} - ill) + 2(\eta_e - \eta_l) - (\eta_{ee} - \eta_{ll}) \tag{2.199}$$

where the subscripts (ee) and (ll) stand for very early and very late and represent the correlation function at $\tau \pm d$. The noise terms are correlated as described in [Eis97]. Thus we get

$$\begin{aligned} Var(D_{dd}) &= E\left\{(2(\eta_e - \eta_l) - (\eta_{ee} - \eta_{ll}))^2\right\} \\ &= 8\left[R(0) - R(d)\right] - 8\left[R(d/2) - R(3d/2)\right] + 2\left[R(0) - R(2d)\right] \end{aligned} \tag{2.200}$$

For the rectangular, infinite bandwidth case eq. (2.200) simplifies to

$$Var(S_{dd}) = 4d \tag{2.201}$$

which is identical to the results in [MB99], when the different normalization and definition of d that are used here are considered.

2.6.3. Mattos-Style Detector

At ST Micro Electronics a multipath mitigation technique was developed, which is based on the simple idea of tracking the correlation function on the early slope of the correlation function only[Mat96]. Instead of having an early and a late correlator, both correlators are early. To achieve this the discriminator function must have its root on the early slope of the correlator function and not at its maximum. For example this can be achieved by the following discriminator function:

$$S_M(\tau) := \eta - \frac{i_e(\tau) + q_e(\tau)}{i_l(\tau) + q_l(\tau)}\,. \tag{2.202}$$

Firstly we notice that the absolute signal power cancels out and secondly for η = 1, $S_M(\tau)$ basically reduces to the coherent early – late discriminator. If η is chosen such that the tracking point (i.e. the root of eq. (2.202)) lies far to the left the signal strength at the tracking point will be correspondingly low, i.e. the signal-to-noise ratio will suffer.

For a given correlator spacing D the tracking point will is given by:

$$\begin{aligned} S_M(\tau) &= 0 \\ \Rightarrow \eta &= \frac{\tau_0 - D + T_c}{\tau_0 + T_c} \\ \Rightarrow \tau_0 &= \frac{D}{1-\eta} - T_c \end{aligned} \tag{2.203}$$

This equation is only valid if both e_1 and e_2 are in fact on the early slope of the correlation function so setting η = 1 has no meaning (then e_2 would be on the late branch of the correlation function).

2.6.4. Model Summary

The development of the model described by eq. (2.197) and the development of the previous sections was quite extensive and to give the reader an overview of what the model actually does and does not model the following list was compiled.

Dynamic modeling The dynamic process of keeping the local oscillators in-step with the received signal (i.e. the actual signal tracking). This implies that effects like signal dynamics and oscillator dynamics are modeled. These effects were discussed in section 2.1.3 and will be addressed in the chapter on simulations. This feature is in contrast to a simple root finding of the S-curve.

Non-linear modeling The full form of the S-curve is considered. This is especially important in combination with dynamic modeling. The combination of dynamic *and* non-linear modeling makes it possible to simulate effects such as loss-of-lock, i.e. when the tracking point leaves the pull-in region of the discriminator, the loop will no longer converge. This is a well-known practical problem, particularly with phase tracking.

Pre-detection multipath fading modeling The multipath fading caused by the tracking loops is automatically taken into account by the dynamic features of the model. There is another kind of multipath fading, related to the frequency de-tuning of the oscillator, caused by the integration in the correlation process, before the signal enters the loops (see section 2.5.1).

Multipath modeling Multipath effects are simulated in the receiver deterministically (see section 2.5.1).

General pulse shape modeling By pulse shaping the following features are meant:

- Intentional shaping, like raised cosine pulse shaping.
- Pulse shaping caused by band-pass filtering.
- „Pulse shaping" caused by modulation, like binary or sinusoidal offset carrier modulating (BOC or SOC).

The modeling assumes an analytic form of the correlation function generated by the correlation process.

Noise modeling The effect of thermal noise is treated within the non-linear, dynamic framework of the model. The noise model has, of course, to fit into the general scheme of things and combined with the analytic derivation of the correlation function the following two effects are considered:

- Discriminator gain controlled by the change in the shape of the correlation function

- Reduction in signal power controlled by the correct normalization of the correlation function

These effects in turn must be combined with the correct normalization of the discriminator of the tracking loops. At the time of this writing there seems to be some confusion in the literature regarding these issues. They are set straight in [BK00, BF00, Bet00b].

What is *not* considered in this simulator is

Finite code lengths All the analysis and implementation assumes an infinite code. However, a finite integration time is assumed in the pre-detection.

Inter-system noise The impact of multiple access interference is not considered in the models.

Data demodulation No data demodulation is implemented in SNSS. The data transmission capabilities of a satellite navigation system are of secondary interest. Whether the bit error rate is 10^{-4} or 10^{-6} is not so important. If the receiver misses a bit, it just has to wait for the data (ephemeris, clock etc.) to come around again.

Acquisition Acquisition of the signal is implemented in a limited way. As the simulation is dynamic, the receiver must of course look for the signal and for that the same basic strategy as in real receivers is used. The receiver looks for signal power, by comparing $i_p^2 + q_p^2$ to an acquisition threshold, defined by the user. There is no search in frequency.

3. Algorithms and Implementation

In this chapter implementation of the models described in the previous chapter is discussed. The chapter is designed to give the reader

- an overview of the software design and
- a detailed description of how the models are implemented in algorithms

From an implementation and algorithmic point of view the core element is the receiver model, which is basically modeled as a set of differential equations. The entire framework around the receiver model (satellite dynamics, user dynamics, propagation of the signal, its interaction with the environment etc.) serves to generate boundary conditions for the receiver model.

3.1. Overview

In figure (3.1) an overview of the simulator is shown. In the following figures the details of the components are explained. The components shown are:

- *Input persistent storage:* By this all kinds of input data is meant, which is read from file. This could be almanac data to describe the satellite orbits or a reference path, recorded with a GPS receiver or a high-precision inertial navigation system (INS). Data about the environment like 3-D terrain data or city models is also held in persistent input storage.
- *External influences:* are object models that generate boundary conditions for the receiver model. Basically, these are geometric and electromagnetic entities that influence the signal. Among others they are the constellation of the satellites and the signal they emit, the reference path of the receiver, terrain and buildings as well as the vehicle that the receiver is attached to. These objects may have a time evolution (motion of the vehicle and the satellites) and electromagnetic properties (dielectricity and conductivity of the buildings) and their combined interaction with the signals define the boundary conditions for the receiver.

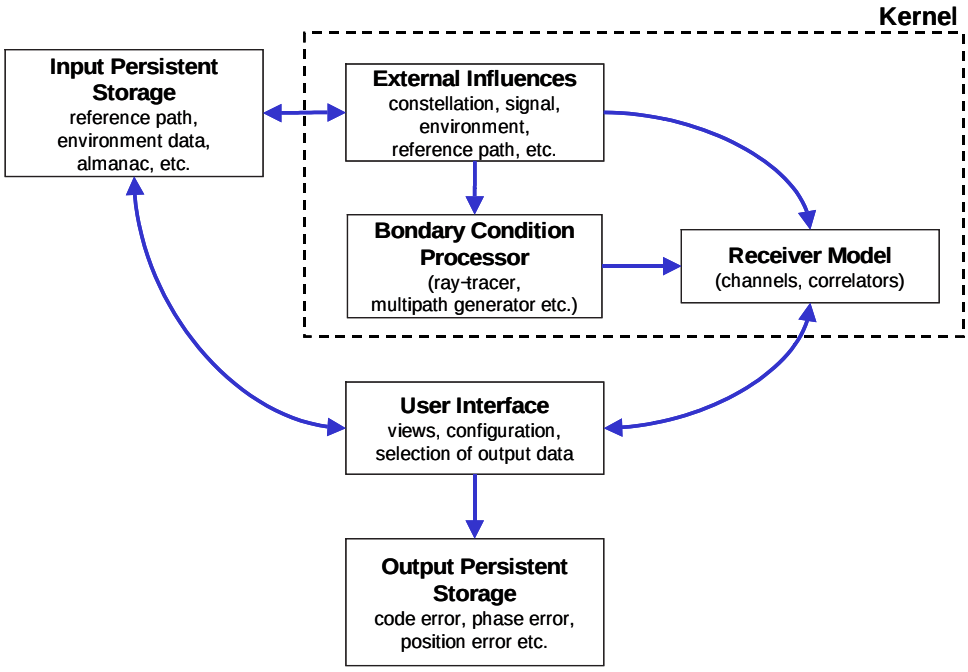

Figure 3.1.: Overview of the SNSS program structure. See text for explanations.

- *BC Processor:* The external influences define the signal that is actually received at the antenna. The Boundary Condition Processor (BC Processor) „translates" the external influences into boundary conditions for the receiver model. It performs geometric and electrodynamic ray-tracing, generates multipaths and keeps track of the time evolution of the multipaths. This data is then ready for the receiver model

- *Receiver Model:* The receiver model is the central point of the simulator. The receiver is divided into channels, each of which tracks one satellite (or pseudolite) using the stochastic dynamic model introduced in the previous chapter. The External Influences, BC Processor and the Receiver Model modules are refered to as the kernel.

- *User Interface:* This part handles the interaction between the computer and the user. Here the external influences as well as the results are displayed graphically. From here the user can configure the receiver model and choose what external influences to use via the input persistent storage. As the simulator generates a huge amount of data the user can choose interactively what results he/she wants and send them to a file.

- *Output Persistent Storage:* Via the User Interface the user can choose the desired results and save them on the hard drive. It is also possible to generate orbits, environments and signals over the User Interface which can be saved to persistent storage.

The list above gives a general idea of how SNSS is organized. The kernel is of course of most interest, since that is where the actual processing takes place. In figure (3.2) an overview of the kernel is given. In the following list the numbers refer to the numbered circles in figure (3.2).

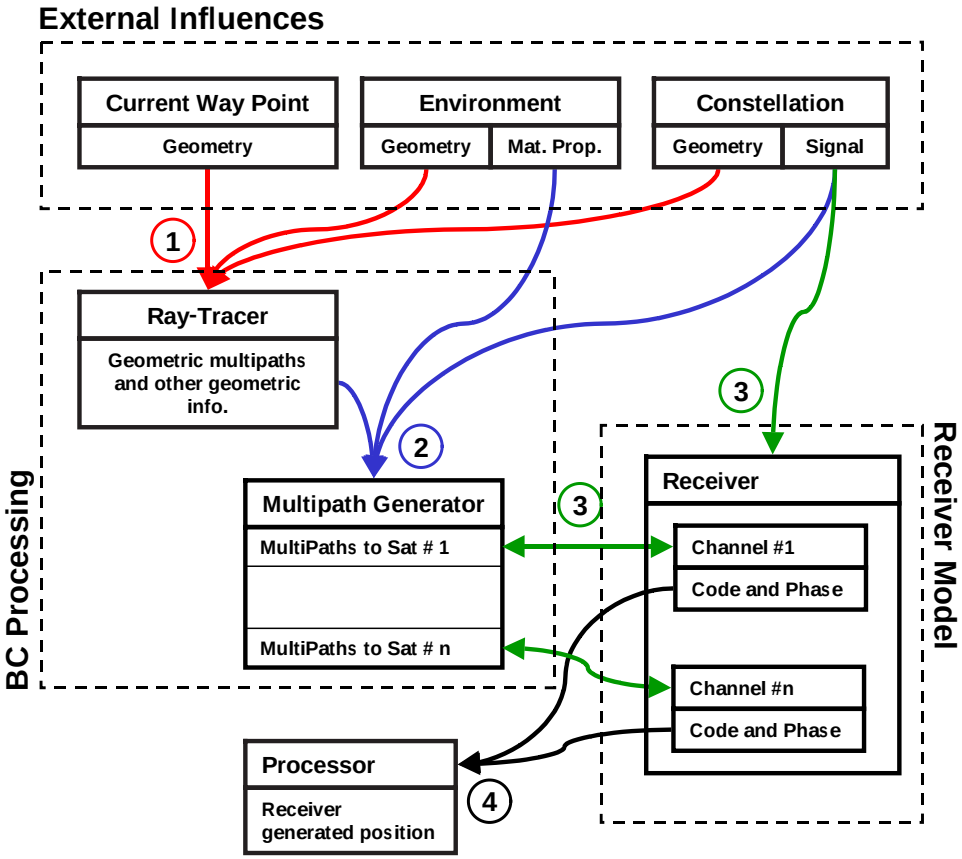

Figure 3.2.: Overview of the kernel of SNSS . See text for explanations.

1. In the first stage the information on the geometry is gathered. These are:
   - Satellite position and dynamics (as many time-derivatives as necessary).
   - The terrain geometry. The terrain in SNSS is made up of a list of flat polygons. This can be just one rectangle or a polygon model generated from raster data[1]
   - A polygon model of the vehicle.
   - A polygon model of the environment.

[1] flat means that all the vertexes of the polygon must lie in the same plane. For a polygon with three vertexes this is always the case, but if there are more there is a constraint on them.

With this purely geometric information the ray-tracer of the BC Processor performs an exact ray-tracing and generates the geometric multipaths. This is done for each transmitter.

2. In the second stage the multipaths are generated. For this task the geometric multipaths from the ray-tracer are needed as well as information on the signal emitted from the transmitters and the material properties of the external influences. Furthermore the multipaths must be given a signature, which depends on the ID of the transmitter and the IDs of the polygons that reflected the signal[2]. Now the boundary conditions for the signal entering the antenna have been generated.

3. The receiver consists of an array of channels, each of which consists of a tracking loop (code and phase). Governed by the output of the BC Processor each channel in the receiver tracks a signal from a satellite. First the correlation process is simulated and then the result is fed into the tracking loops.

4. After the receiver has generated the pseudo-ranges and phase measurements the data is processed to yield a position.

Data flow All this information on the external influences can either be entered manually or read from file. The organization of the input data is shown in figure (3.3). The possible input data is

Constellation: Orbit data for the transmitters can be read from file. The file formats are the well-known Yuma almanac format used by GPS. Stationary transmitters like pseudolites and geostationary satellites can also be positioned by one 3-D vector position.

Signal Data: The signal format contains information on the signal strength emitted from the satellite as well as carrier frequency, pulse shape etc.

Reference Path: Recorded position data can be loaded as a reference path. The simulator assumes the reference path to be the „true" path and uses it to generate the proper boundary conditions. Then the receiver generates a position, which is the „measured" position. Comparing the reference position with the measured position defines the measurement error.

Terrain: A Terrain can be loaded in VRML format, but SNSS is also able to generate a polygonal model from a raster format (digital elevation model) provided the data is a rectangular region[3].

Environment Buildings, bridges etc. can be loaded, provided they exist in the VRML 1.0 format. Basically they are treated in the same way as the terrain.

[2] This is needed in order to be able to interpolate the dynamics of the multipaths

[3] „Rectangular" here means that the data set has n columns and m rows

Vehicle: 3-D Polygon data for a vehicle model. A vehicle model can be loaded from a VRML 1.0 file.

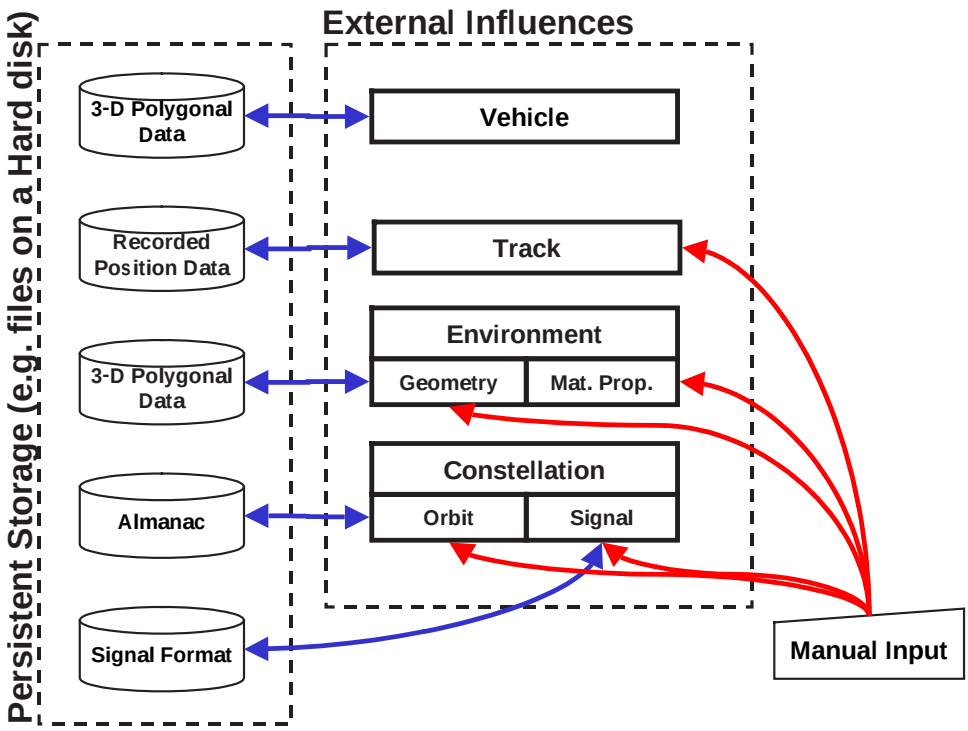

Figure 3.3.: Overview of the data flow and storage in SNSS . See text for explanations.

SNSS can also write all these file formats to file. Through the user interface satellite constellations and signal formats can be generated. Buildings and reference tracks can also be entered and written to file.

User interface The results of the simulation are displayed graphically on the user interface. Via the user interface the receiver model can be configured and selected results can be written to file. An overview of this is given in figure (3.4). There are four views:

2-D View: A projection of the environment onto a 2-D plane. In this view buildings and a reference path can be edited. Positioning of pseudolites is also possible. The editing option for the buildings supports entering and modification of 3-D boxes with any number of vertexes. The reference path edit option supports entering of piece-wise linear paths with constant acceleration. Using the reference-path a polygon model of the terrain can be generated. In this manner the road the vehicle was driving along can be reconstructed from recorded GPS data alone.

3-D View: A 3-D view of the environment, vehicle and buildings. As the objects comprising the External Influences are rather complex, it is of interest to be able to visualize the situation. In this view the whole scenario is displayed in 3-D including the geometry of the multipaths.

Sky Plot: A polar plot of the constellation. In this view the properties of the signal can also be viewed and set. The user is able to generate a constellation in a set (Walker constellation) or he/she can set the almanac parameters of each satellite individually. For each satellite the signal structure can be controlled. The satellite can be set to emit any number of frequencies, using any signal structure described in the previous chapters.

Data View: An output data view, where all results of the simulation can be displayed. Among others there are the geometrical multipath, the signal-to-multipath ratio, the code and phase errors for each channel, the error in position; the total error, the horizontal error and vertical error and many more.

Configuration Dialog: The user interface contains a dialog where the various parameters of the receiver and the antenna can be adjusted. The antenna gain pattern can be adjusted to the characteristics of any given antenna (with or without choke-ring), dependent on elevation. The antenna gain pattern can then be viewed in the Data View. All dynamic parameters of the receiver are set in this dialog as well as the size and resolution of internal buffers for the generated data. Parameters for the processing of the resulting ranges and phases can be adjusted in this dialog such as whether differential or single point positioning is preferred, what frequency bands to use etc.

3.2. Algorithms

As already stated it is not the goal of this chapter to give a detailed description of the code of SNSS , but rather to explain the algorithmic context. Before we go into details a list of utilities that are assumed to exist is presented and are not explained in detail. A developer, who has access to the source code and wishes to add features is referred to the comments in the source code.

- Vectors: 3-D vectors with the usual mathematical properties
- Vectors with WGS-84 properties: provides a transformation between Cartesian and elliptic WGS-84 coordinates. It can also generate a local coordinate system with axis parallel to the local East, North and Up direction
- Matrices: Mathematical matrix object

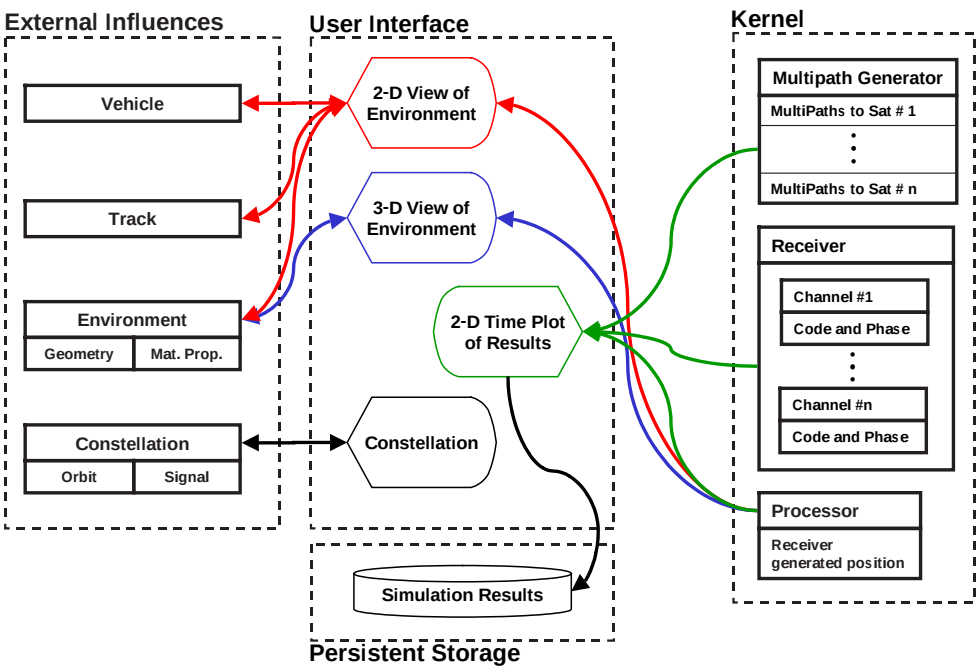

Figure 3.4.: Overview of the man-machine interface in SNSS . See text for explanations.

- Time object: manages transformations between UTC-time, GPS-time etc.
- Complex numbers with the usual mathematical operations on complex numbers

These mathematical objects are all implemented as C++ classes and possess the usual properties one would expect, e.g. two vectors can be added, subtracted etc. The scalar products as well as the cross product is defined. Matrices can be inverted and multiplied etc. so for example a least square adjustment is performed in a trivial manner.

The implementation of SNSS uses several libraries. The most important are Microsoft Foundation Classes (MFC) for the Graphical User Interface (GUI). Windows, dialogs, buttons etc. are handled by this library. The Standard Template Library (STL) provides standard tools of the trade for handling data structures (linked lists, vectors, stacks and buffers etc.) It also provides template objects for file input/output (file I/O). The 3-D visualization in SNSS is based on the well-known OpenGL 1.2 library from Silicon Graphics, available for most platforms. Many of the mathematical objects mentioned above contain algorithms and modified source code from the Numerical Recipes in C [PTVF97]. Other parts of SNSS also contain material from the Recipes.

3.3. External Influences

3.3.1. Constellation

For the purposes of the simulation an orbit description using data corresponding to the almanac data of the GPS system are sufficient. In the simulator two methods of describing the orbit of a transmitter are implemented, a Kepler orbit based on almanac data and a stationary position in WGS-84. The basis object (class `CNavSource`) contains information for over-head information like ID, structures containing the buffered data such as time, position, velocity and status flags. The signal format is also a member of class `CNavSource`. Specialized versions of the satellites are then described with the objects class `CSatAlmanac` and class `CPseudolite`.

The signal parameters are contained in the structure `SSignalData`.

The structure `SNavSourceData` contains administrative data such as the current time and position, along with identifiers and some state variables.

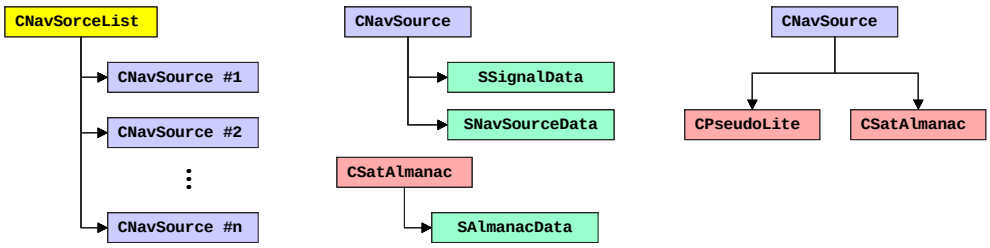

Figure 3.5.: Organization of the satellite objects in SNSS . The left diagram shows how the satellites are organized in a list of pointers. The diagrams in the middle show the basic data in the base class `CNavSource` and the derived class `CSatAlmanac`. The right diagram implies the class hierarchy for the satellites/pseudolites.

Signal Every type of satellite carries a description of the signal it emits. The actual generation and simulation is performed by the Boundary Condition Processor and the receiver module, thus the satellites only deliver a parametric description on the signal. The signal parameters maintained at the satellite are shown in table 3.1.

Orbits The more important satellite type is the `class CSatAlmanac`. Its orbit is described with the equations for the description of the almanac orbits was given in section 2.1.1.1. The input data available from an almanac file are: Using the quantities in table 3.2 and the description in section 2.1.1.1, the following algorithm for the position of the satellite as a function of time is obtained.

Parameter	Description
Carrier frequency	ω [MHz]: This is the center frequency of the signal
Chip length	T_c [m]: The chipping length of the signal
Power	P [w]: The power emitted from the antenna of the satellite
Side lobes	N_s The number of side lobes emitted by the satellite. This parameter controls how many side-lobes of the spectrum is let through the filter at the satellite.
Roll-off factor	β: This factor controls how abrupt the spectrum of the signal decays. See eq. (2.35) in section 2.2.6
Band ID	This index enumerates the frequency band of the corresponding signal.
Pulse Shaping	This parameter defines what pulse-shape is to be assumed for the signal. The values possible are: • INFINITE_BANDWIDTH: Rectangular chips with infinite bandwidth • NO_PULSE_SHAPING: Band limited rectangular pulse-shape • RAISED_COSINE: Raised cosine pulse shaping scheme • SPLIT_SPECTRUM: Autocorrelation of a BOC signal • SUB_CARRIER: Tracking a BOC signal using the code only • LATE_TRACKING: Special setting for GPS altimetry

Table 3.1.: Signal description at the satellite. These parameters can be stored and read from file.

Description	Param.	Description	Param.
Square root of the semi major axis	a [$\sqrt{\text{m}}$]	Longitude of ascending node	Ω [°]
eccentricity	e	Reference week of almanac	
Inclination	i [°]	GPS time of reference	t_{oa} [s]
Argument of perigee	ω [°]	Mean anomaly	M_0 [°]
Right ascension of ascending node	Ω_0 [°]	True anomaly	ν [°]

Table 3.2.: Almanac parameters loaded from a Yuma-file

Assuming the data from an almanac file has been read from a Yuma file, i.e. data described in table 3.2 (a, Ω, e, i, t_{oa}, ω,M_0, Ω_0, ν) and the Earth rotation ($\dot{\Omega}_E$) to be available we then proceed to calculate the orbit as described in the ICD-GPS-200C-004 [ARI93] except for the clock corrections.

1. WGS-84 value for the earth's rotation universal gravitational constant: $\mu := 3.986005 \cdot 10^{14}\mathrm{m}^3/\mathrm{s}^2$
2. Earth's rotation: $\dot{\Omega}_E := 7.2921151467 \cdot 10^{-5}$ rad/s
3. Semi-major axis: $A = a^2$
4. Mean motion: $n_0 = \sqrt{\mu/A^3}$
5. Time from almanac reference epoch: $t_k = t - t_{oa}$
6. Mean anomaly: $M_k = M_0 + nt_k$
7. Solve iteratively for the eccentric anomaly, using Kepler's equation: $E_k^{i+1} = M_k + e\sin(E_k^i)$ with initial value $E^0 = M_k$
8. To obtain the position and higher time derivatives thereof, the equations in section 2.1.1.1 are used.

The second type of satellites are the stationary pseudolites. Although they are called pseudolites they are equally well suited for the description of geostationary orbits. These are simply described by a position in WGS-84. This is used to describe geostationary satellites and pseudolites.

3.3.2. Polygons

The basic 3-D primitive in SNSS is a flat polygon. All objects are constructed from polygons. The algorithmic implementation of the polygon is based on the mathematical description given in section 2.3.2. It is a C++ object (a class) called `CFace` which has the following properties:

- **Data**
  - Verticies: list of 3-D vectors
  - Polygon normal: 3-D normalized vector
  - Material parameters: parameters describing electromagnetic properties of the polygon, i.e. dielectricity and conductivity.
- **Methods**

- Loading data from file
- Writing data to file
- The obvious setting and retrieval routines
- Initialize: here it is important to ensure that the polygon is flat.
- `Mirror`: mirrors a 3-D vector according to eq. (2.90) in section 2.3.2
- `IsOnTheOutside`: Checks if a 3-D point is in the positive half-space, defined by the polygon using eq. (2.96)
- `IntersectsFace`: checks whether a line-segment defined by two 3-D vectors intersects the face defined by the polygon. First the intersection of the line-segment with the plane in which the face lies is determined using eq. (2.92) and eq. (2.93). Then it is checked if the intersection point is on the face, using the algorithm described on page 68.

The polygons are contained in a list called `CFaceObject`. This is shown in figure (3.6)

The following objects which are composed of a set of polygons (i.e. the terrain, vehicle and environment) are basically lists of polygons. They may differ slightly in their specialization.

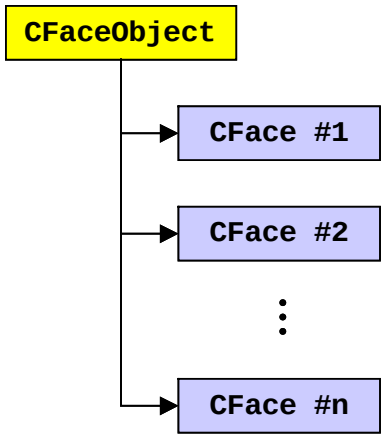

Figure 3.6.: Organization of the polygon objects in SNSS . The polygons `CFace` are contained in a list of pointers `CFaceObject`.

Terrain The terrain consists of a list of polygons. There are three ways to load/generate the terrain. The simplest form is just a rectangle (a polygon of the type described above (class `CFace`)), which is placed tangential to the WGS-84 ellipsoid.

The terrain can also be generated from the reference path. This option generates a list of triangles, a certain height below the reference-path. The algorithm is sketched in figure (3.7).

To generate the part of the terrain between way-points i and $i+1$ the spanning line-segments ($\vec{l}_i$ and $\vec{l}_{i+1}$) are calculated (black, dashed double-arrows at way-points i and $i+1$ indicated in figure (3.7)). The line segment $\vec{l}_i$ is defined as lying in the plane spanned by $\hat{e}_{i-1}$ and $\hat{e}_i$ (shown as red vectors in figure (3.7)) with the angles between $\hat{e}_{i-1}$ and $\vec{l}_i$ and $\hat{e}_i$ and $\vec{l}_i$ equal, i.e. $\angle(\vec{l}_i, \hat{e}_i) = \angle(\vec{l}_i, \hat{e}_{i-1})$ (see figure (3.7)). Now two triangles are constructed, both with a normal pointing „upwards". The user is asked for the length of $\vec{l}_i$.

The third possibility to obtain a terrain in SNSS is to generate a polygon model from raster data, i.e. data in form of coordinates and a height. The current implementation allows only raster data organized on a rectangular grid.

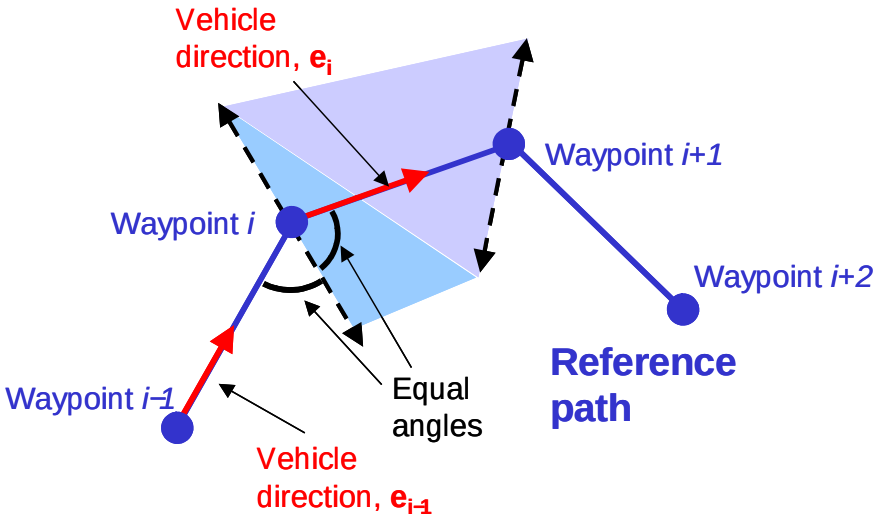

Figure 3.7.: Generating a terrain from a reference path. See text for explanations.

The generation of the terrain from raster data is easier. The coordinates are assumed to be on a grid, i.e. the x-values repeat themselves every N data points. This is shown in figure (3.8). The routine reading the data simply checks when the first x-coordinate is (almost) repeated. This yields the value of N (the number of rows). The list of triangles comprising the terrain is then easily generated as indicated in figure (3.8). It is necessary to generate triangles in order to ensure flat polygons.

Environment The environment in SNSS is represented by a list of closed „voxels" class (`CBuilding`), which are composed of a set of polygons. The buildings must be closed so that the ray-tracing can be made efficient. If open constructs are needed, then this can be achieved by replacing each polygon with two polygons which are identical except for the normal vector. Furthermore things like bridges and tunnels can be made out of a set of closed „voxels".

The internal representation of the environment is shown in figure (3.9). When the buildings are entered manually, each building has a separate representation, so that each building can be manipulated individually, independent of other buildings. This is very helpful when setting up a virtual environment which is meant to demonstrate some particular phenomena.

The current implementation supports the import of environment data in the VRML 1.0 format. However, only so-called indexed face sets are supported. Because most data providers make no distinction between buildings in this file format, all the data is imported as one single building. Consequently, editing of individual buildings is not supported. For the processing this does not matter, it only limits the editing possibilities.

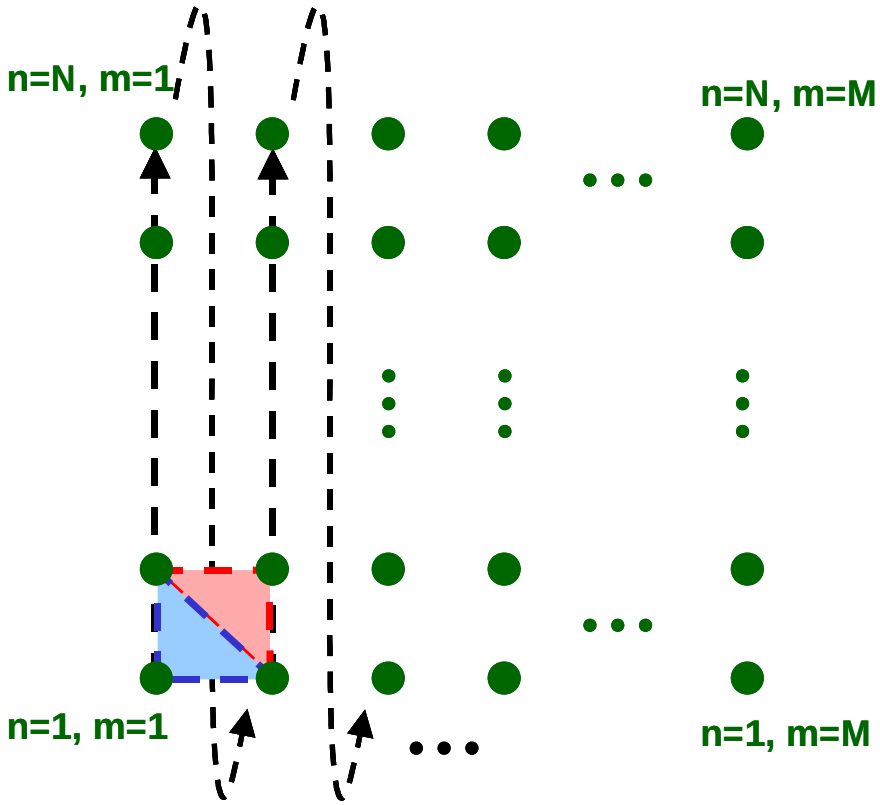

Figure 3.8.: Generating a polygon model from raster data.

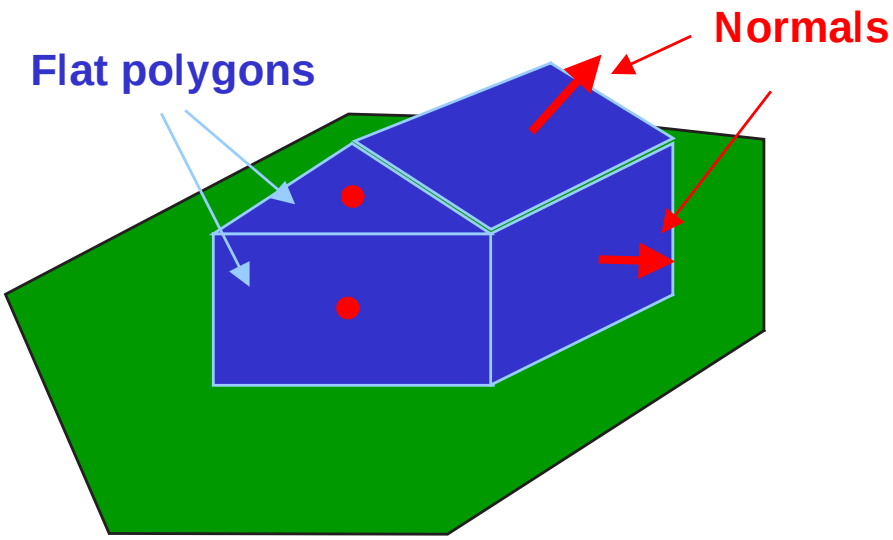

Figure 3.9.: 3-D representation of the environment in SNSS .

Vehicle Basically, the only difference between a vehicle (`CVehicle`) and a building in SNSS is that a vehicle can move and rotate. This difference is not as trivial as it seems at first, because the vehicle must move along the reference path and it is also desired that it faces in the right direction (a car which always points to the north is not going to yield a very realistic scenario. For planes this is more difficult, because „up" is not always perpendicular to the ground).

In SNSS all information on the dynamics of the reference path is kept in the reference path (see next section). Therefore the vehicle must retrieve the information on the dynamics from the reference path. After that the vehicle is able to translate and rotate to the given way-point. The question of what the dynamics look like is thus transferred to the reference path in the next section.

The vehicle data is exclusively loaded from a VRML 1.0 file as one single indexed face set object. Authoring a VRML object is a quite cumbersome task, but on the Internet there is a huge amount of vehicles in this format available which can be downloaded free of charge. VRML models are also commercially available.

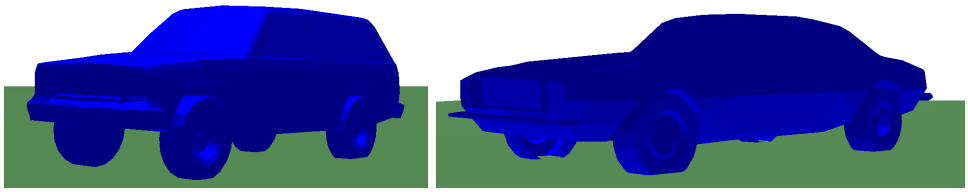

Figure 3.10.: Examples of land vehicles in SNSS .

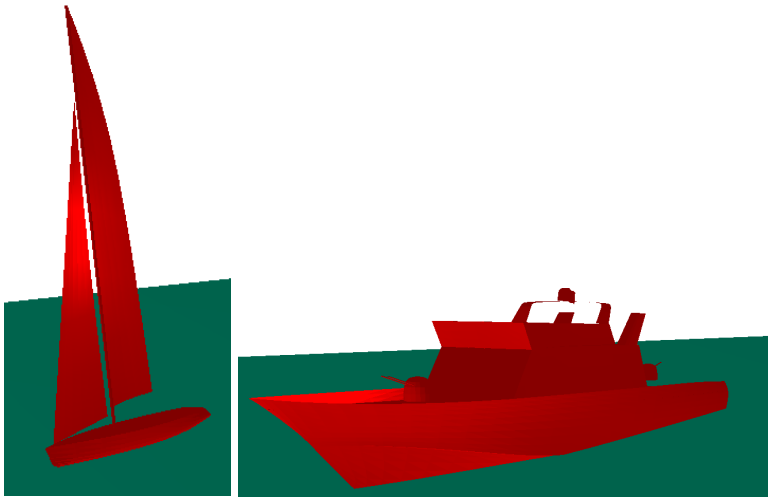

Figure 3.11.: Examples of maritime vehicles in SNSS .

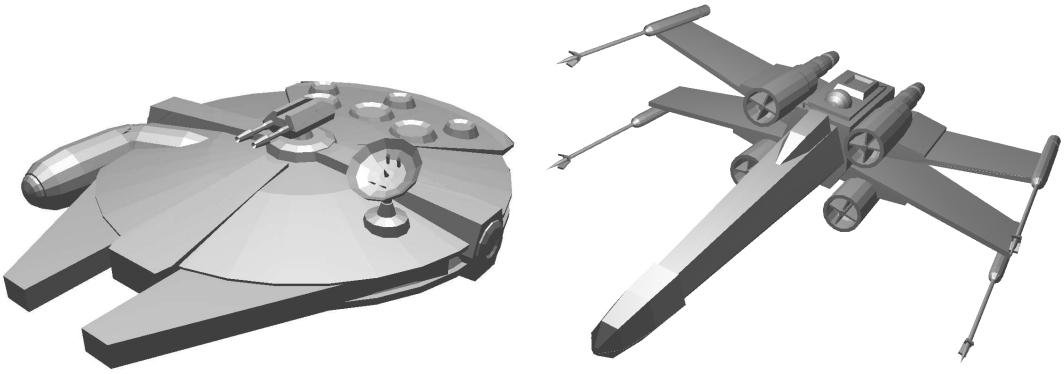

Figure 3.12.: Examples of fantasy vehicles in SNSS

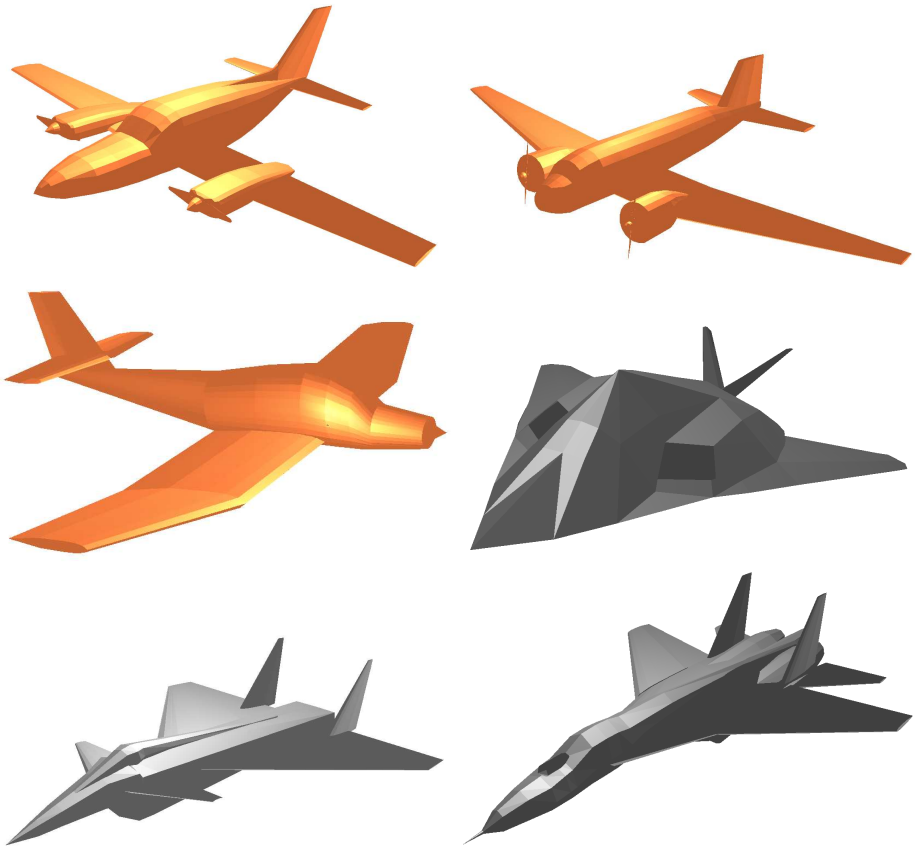

Figure 3.13.: Examples of airborne vehicles in SNSS .

Therefore it is a good idea to write a flexible import routine. To further enhance this flexibility the vehicle in SNSS has it own internal orientation, which the user can change. In this way what is „front" and „up" can be controlled by the user. This has turned out to be a very useful feature, because most of the VRML models available do not have a well-defined orientation.

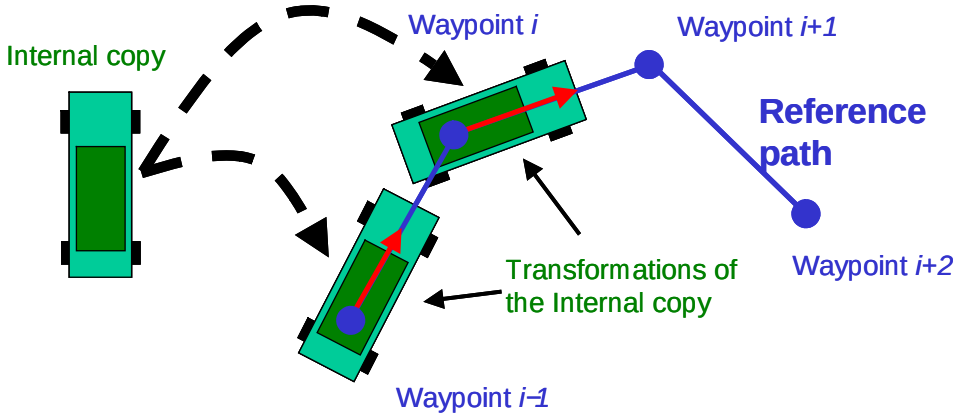

Figure 3.14.: Motion of the vehicle in SNSS . An original, local copy is transformed each time. This avoids accumulative transformation errors and makes it easy to adjust the general orientation of the vehicle. See text for explanations.

Even though the VRML standard length-measure is meters, very few authors design their models correctly to scale. Therefore the user can scale his/her vehicle model at will.

When the simulation is running SNSS chronologically works its way along the reference path. At each point the boundary conditions are computed. For the vehicle this means that it gets transformed from one point to the other. If there are many waypoints and the vehicle would in fact be transformed from one point to the other, the orientation and the vehicle itself would be distorted due to accumulating rounding errors. Therefore the vehicle keeps a copy of the geometry as it was loaded from file. Each time the vehicle is needed at a particular track point, the original copy is transformed to that point. This is sketched in figure (3.14).

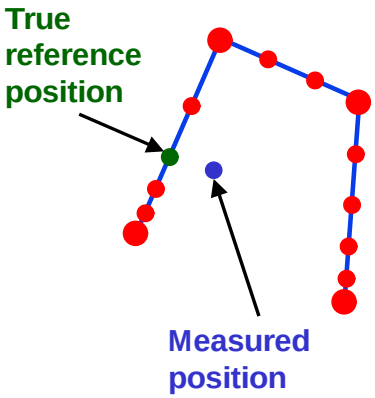

Figure 3.15.: 3-D representation of the reference path. See text for explanations.

As mentioned earlier, the orientation of the vehicle when loaded from file, may not be the desired one (a car may be lying on the side or an aircraft may be standing upside-down). Therefore it must be possible to change the orientation of the vehicle for *all* reference points. This is easily accomplished by rotating the *local copy* of the vehicle.

3.3.3. Reference Path

As already stated the reference path represents the „true" trajectory of the receiver. The reference path (class `CTrack`) is a list of way-points (class `CTrackPoint`). It is instructive to take a closer look at those two objects. In figure (3.16) the organization of those classes is shown.

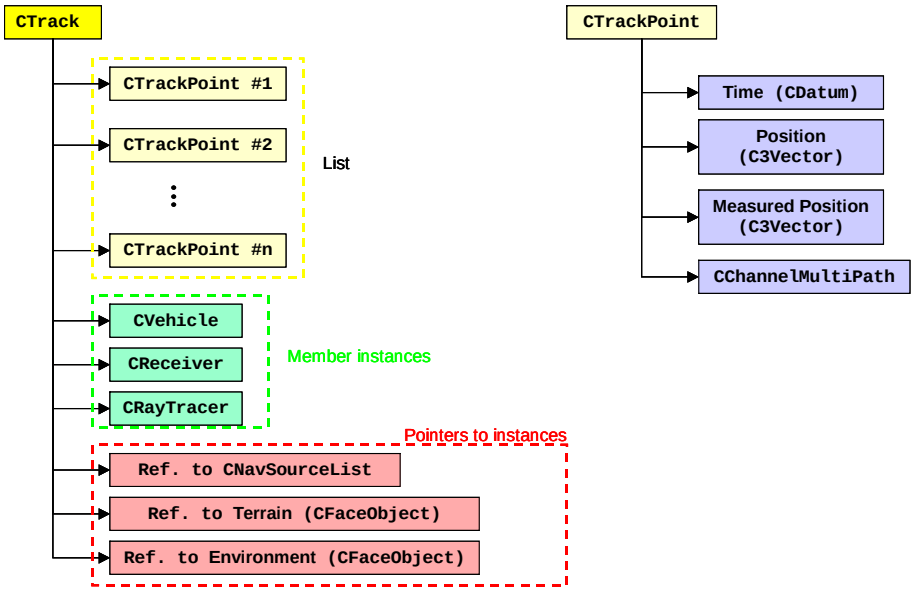

Figure 3.16.: The left diagram shows the members of the class `CTrack`. The right diagram shows the data contained in each track point (`CTrackPoint`).

The `class CTrack` object is organized as follows

- **Data**
  - A list of way-points class `CTrackPoint`
  - References to the environment, terrain, satellites and Ray-tracer
  - Receiver
  - A vehicle object (class `CVehicle`)
- **Methods**
  - Loading data from file
  - Writing data to file
  - Interpolating position and velocity between two track points (see figure (3.17) for explanations)

  - Managing the Ray-tracer, terrain, vehicle and satellite list for geometric ray-tracing. Basically, the ray-tracer is initialized from within the class `CTrack-Point` object with the above-mentioned objects.

The class `CTrackPoint` is organized as follows

- **Data**
  - Time
  - Position, velocity and acceleration of the reference way-point
  - Measured position
  - List of multipaths `CChannelMPathList`
- **Methods**
  - Loading data from file
  - Writing data to file
  - Retrieving multipaths
  - Starting ray-tracing for this track point.

Each track point `CTrackPoint` in the reference path has a time and a position coordinate (and higher derivatives thereof). In SNSS each track point keeps a list of multipaths, which uniquely define the boundary conditions *at the corresponding track point*. The track points are collected in a chronologically ordered list, the track or reference path (`CTrack`).

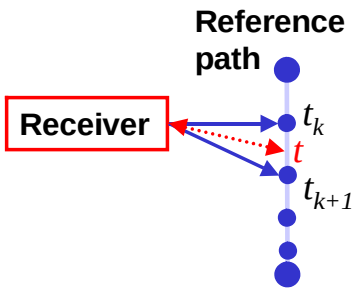

Figure 3.17.: Interpolation in time between adjacent reference positions. See text for explanations.

Virtually, the track can be thought of as a list of track points, which are chronologically ordered and in a certain way connected. As the track points only have knowledge of the boundary conditions for that particular track point and the track points are only available for a relatively coarse time-resolution, it must be possible to interpolate the boundary conditions between two adjacent track points. This is necessary because we want to perform a dynamic simulation of the receiver, i.e. integrate a set of differential equations. The interpolation task, indicated in figure (3.17), is carried out by the track (`CTrack`). The red dotted arrow in, figure (3.17) implies a request from the receiver for boundary condition data for a certain time point t. Using the information on position, velocity and acceleration of two adjacent track points, the position and velocity are

interpolated using

$$\begin{aligned}\vec{R}(t) &= \vec{R}_k + \tau_{k,k+1}\vec{R}_{k+1} \\ \vec{V}(t) &= \vec{V}_k + \dot{\tau}_{k,k+1}\vec{V}_{k+1} \\ &= \vec{V}_k + \frac{t-t_k}{t_{k+1}-t_k}\vec{V}_{k+1}\end{aligned} \tag{3.1}$$

where τ is given by

$$\tau_{k,k+1} = v_k(t-t_k) + \frac{v_{k+1}-v_k}{2(t_{k+1}-t_k)}(t-t_k)^2 \tag{3.2}$$

and v_k is the speed of the receiver in WGS-84, t is a time coordinate between t_k and t_{k+1}. Thus we are talking about an interpolation of second order in time.

The same procedure is used for the geometric multipaths and the equation for that will be written down in the subsequent section.

3.4. Boundary Condition Processor

The propagation of the signal in SNSS can generally be divided into two domains; the geometric ray-tracing, which solely depends on the given geometry and the electromagnetic domain, which based on the results of the geometric analysis and the electric properties of the environment (i.e. terrain, buildings and vehicle), yields the electromagnetic properties of the signal at the antenna of the receiver.

3.4.1. Geometric Ray-Tracing

There are many ray-tracing algorithms described in the literature. However, for the purposes of this work a geometrically exact ray-tracing algorithm was developed. It is based on the operations described in section 2.3.2.

To construct the ray-tracing algorithm we use the analysis and equations from section 2.3.2. The algorithm is exact in the sense that there are no approximations or assumptions made on the resolution, like for example the angular resolution of the source. The objects are all composed of polygons and simple, exact geometric analysis is used to obtain the reflection points of the rays. It is, however, assumed that Snell's law is valid, i.e. incident angle equals the reflection angle. For smooth surfaces and plane waves this is rigorously true, but for rough surfaces the wave is scattered in other directions as well.

In figure (3.19) the ray-tracing algorithm is sketched. Referring to that figure the algorithm is described in algorithm (1). The required input of a list of polygons ($\{\mathcal{L}\}_{i=0}^{n}$) is to

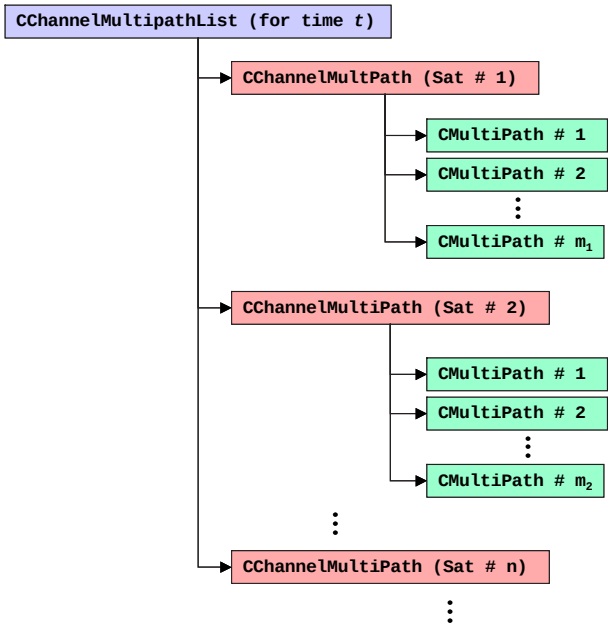

Figure 3.18.: Overview of the relevant data objects, managing the results of the analysis of the boundary conditions. See text for explanations.

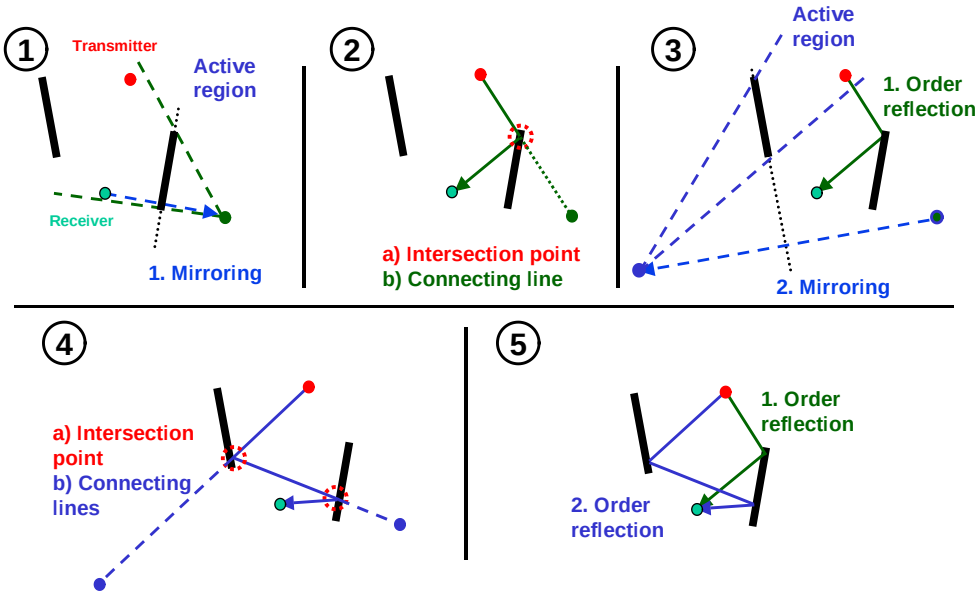

Figure 3.19.: Understanding the ray-tracing algorithm.

be understood as a pre-defined set of polygons, that are tested for reflections. It is important to note that assuming we want to find all reflections to order n (typically in the order of 1 or 10) and given a large number of polygons N (in the order of 100, 1000 or more), we must analyze each combination of n polygons that can be selected out of N polygons. It is therefore obvious that the algorithm is of the order $\mathcal{O}(N^n)$. So for $N = 1000$ polygons and, say, a maximum reflection depth of $n = 4$, we must analyze something in the order of $N^n = 1000^4 = 10^{12}$ reflections! The recursive nature of algorithm (1) therefore suggests to simply unfold the recursion and define an absolute, hard coded maximum possible reflection depth of 4. Programmatically, this corresponds to four nested loops. The ray-tracing is numerically a very demanding algorithm. Unfolding the code like this may enable the compiler to generate more efficient machine code.

Require: Pre-defined reflection-depth, N List of polygons (`CFace` objects), $\{\mathcal{L}\}_{i=0}^{N}$. Source $\vec{A}$ and receiver $\vec{B}$. Vehicle is on the correct position and orientation. Satellite dynamics updated and path dynamics updated

1: Allocate a list of 3-D vectors, $\{\vec{M}\}_{i=0}^{n}$ for the mirror points
2: Allocate a list of 3-D vectors, $\{\vec{R}\}_{i=0}^{n}$ for the reflection points
3: $\vec{M}_0 \leftarrow \vec{A}$
$\vec{M}_n \leftarrow \vec{B}$
4: **for** i from 1 to n **do**
5: Mirror $\vec{M}_i$ about $\mathbf{l_i} \in \mathcal{L}$ (using eq. (2.90)) and insert into $\vec{M}_{i+1}$ (figure (3.19) 1, blue arrow)
6: $\vec{M}_{i+2} \leftarrow \vec{B}$
7: **for** j from 0 to $i+1$ **do**
8: **if** $\vec{M}_{j+1}$ is visible through $\mathbf{l_j}$ **then**
9: Calculate intersection point between the line segment $\vec{M}_{j+1}$, $\vec{M}_j$ and $\mathbf{l_j}$ (using eq. (2.92) and eq. (2.93)) and store as a reflection point in $\vec{R}_j$ (figure (3.19) 2, red circle)
10: **if** none of the line segments $\vec{R}_k, \vec{R}_{k+1}$ is blocked (eq. (2.91)) **then**
11: create a geometric multipath
12: **end if**{check for blocking}
13: **end if**{check for visibility}
14: **end for**{j from 0 to $i+1$}
15: **end for**{i from 1 to n}

Algorithm 1: Ray-tracing

The properties of the object `CFace` are described in section 3.3.2. The results of the ray-tracing are organized in lists of multipaths. In order not to loose track of the individual multipaths, there exists a list of multipaths for each space-time coordinate of the track point and satellite. This list is called `CChannelMultipath` and it contains objects of the type `CMultiPath`. The organization of this data is shown in figure (3.18).

After the ray-tracer has successfully generated a multipath, a `CMultiPath` object is generated and the results stored in it. In order to identify the multipaths, a unique signature

`CSignature` is generated. The signature is composed of the ID of the satellite and an *ordered* list of the IDs of the reflecting polygons (i.e. the `CFace` objects in the pointer list of the ray-tracer).

Generally, more than one multipath is generated for a particular space-time coordinate and a particular satellite. As indicated in figure (3.18) all `CMultiPath` objects belonging to a certain satellite and point in space-time are collected in a so-called `CChannelMultiPath`. For each satellite and space-time coordinate there exists such an object. These are then all collected in a `CChannelMultiPathList`. Each `CTrackPoint` contains exactly one such list (`CChannelMultiPathList`).

So a `CMultiPath` contains the signal path for a given space and time, satellite and reflector set. A `CChannelMultiPath` contains the signal paths for a given space-time coordinate and satellite (i.e. all `CMultiPath` for a given satellite). A `CChannelMultiPathList` contains all signal paths for a given space-time coordinate.

The `CTrack` object is able not only to interpolate the positions and velocity between two track points, but also the *multipaths* of two track points. The geometric multipaths are interpolated analogously to the position ($\vec{R}(t)$) in eq. (3.1). In order to be able to do so, one must be sure that the „same" multipath is interpolated. More specifically, two multipaths from track points k and $k+1$ will be interpolated if they have the same signature, i.e. they origin from the *same source* and are being reflected by the *same reflectors* in the *same order*. This situation is shown in figure (3.20). The blue reflections at both track points have the same signature; i.e. there is a reflection from the source, to reflector i, to reflector j and then to the receiver at *both* track points. The red reflections, however, have *different* signatures, because at track point k the ray is reflected off reflector j, but at track point $k+1$ the reflector is i. Thus it does not make sense to interpolate between these two.

It should be noted that this procedure does not always yield the desired result. For certain kinds of reflectors, e.g. ones with holes in them, the interpolation may yield a continuous reflection, when it is in fact interrupted. Another drawback is that short term reflections, for example like the red ones indicated in figure (3.20) will be totally ignored. These effects are simply due to the discrete nature of the data and can be reduced by densifying the track, which is tantamount to increasing the update-rate of the receiver. If it is not desired to increase the update-rate of the receiver, but still increase the geometric resolution, one simply ignores the necessary amount of way-points. A real receiver integrates continuously even though the update rate is decreased. In the simulation exactly the same is achieved in this manner.

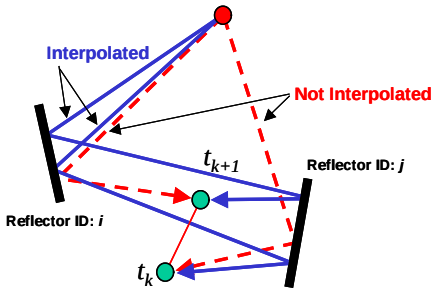

Figure 3.20.: Interpolation in time between adjacent multipaths. The blue reflections are interpolated, while the red ones are ignored. See text for explanations.

3.4.2. Electromagnetic Ray-Tracing

After the geometry has been analyzed it is straightforward to obtain the electromagnetic properties of the signal, given the electric properties of the reflectors and the analysis in section (2.3.3).

The first step is to obtain the Stoke's parameters after a reflection. This is implemented in the `CMultiPath` object. In this object all the necessary information is contained and thus it is natural to equip `CMultiPath` with the ability to perform all or most of the electromagnetic processing. Below the object `CMultiPath` is shown:

- **Data**
  - Reflection points (including source and receiver): list of 3-D vectors. These are the results from the geometric ray-tracing.
  - CSignature: A list of integers, containing the indexes of the reflectors
  - Stoke's parameters, one set for each frequency band. These parameters describe the state of polarization immediately before entering the antenna
  - Amplitude and phase at each reflection point for both polarization states.
  - Surface roughness (σ) at each reflection point
- **Methods**
  - Retrieval functions to obtain information on geometric multipath and the electromagnetic state of the signal
  - `SetMultiPath`: receiving a list of reflection point and a reference to the reflecting polygons, this function calculates the Fresnel coefficients and Stoke's parameters at each reflection point parameter (σ) is also set here
  - `HasSameSignature`: Compares the signature to that of another `CMultiPath` object

This object (`CMultiPath`) implements the first model for a rough surface, section 2.3.4.1, eq. (2.121), where the implied reflection coefficient for finite conductivity is assumed to be the Fresnel coefficients for both polarization states.

If the surface is smooth the reflection is calculated using the Fresnel equation eq. (2.101) for both polarization states. Originally the signal is assumed to be RHC polarized. The complex reflection coefficients are calculated and the change in amplitude and phase of the fields correspondingly updated and stored for *each* reflection point.

After this has been calculated the scattering amplitude is corrected according to the roughness at each reflection point using eq. (2.121).

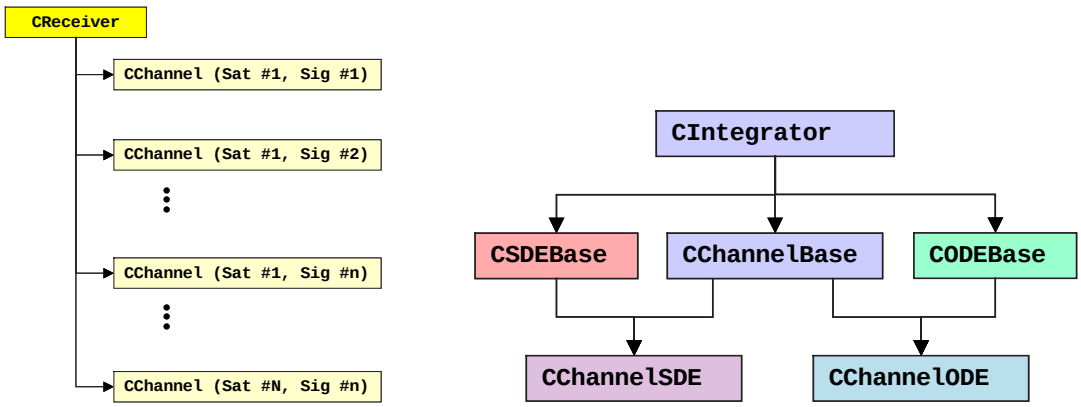

Figure 3.21.: Left figure: Overview of the CReceiver class. The CReceiver object is a list of CChannel instances. Right figure: C++ object inheritance to accommodate the ODE and the SDE within the same structure. Refer to text for more information.

3.5. Receiver

The implementation of the receiver model basically consists of a list of channels. Apart from that the receiver class `CReceiver` manages the interfaces to other objects, related to the generation of the correct boundary conditions. The receivers role is only to establish these connections (in the form of a reference to the correct object) and the channel model manages the data communication by itself, once the connection is established.

After the channels have calculated their ranges and phases, the result is transferred to a position calculator, which performs a least-squares for each epoch independently or a Kalman filtering to produce the measured position. The position processor is a member of the `CReceiver`.

3.5.1. Receiver Channel Implementation

The ODE algorithm is a fifth order implicit Runge-Kutta scheme with automatic step size control. The implementation was taken directly from the Numerical Recipes in C [PTVF97]. The algorithm itself is explained in detail in this reference and the reader is referred to the Recipes for further information. However, the Recipes implementation is in C and consists of a group of functions that form the core of the object-oriented implementation. These functions are members of the `COdeBase` class.

The SDE algorithm is based on the Euler-scheme described in [KP95].

In order to re-use as much code as possible and also accommodate the implementation of the SDE in the same object hierarchy, an object inheritance tree as shown in figure (3.21) was implemented.

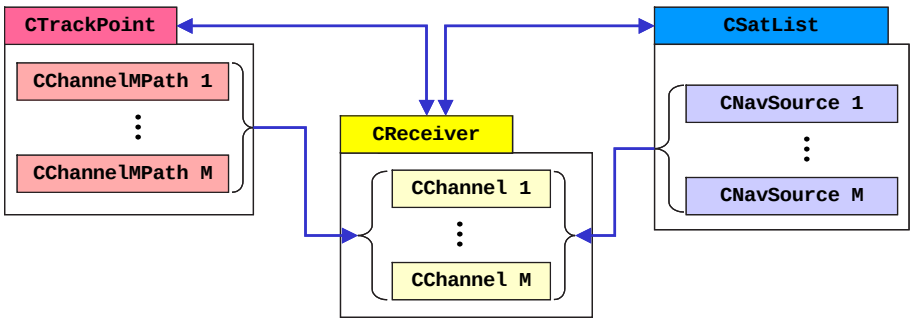

Figure 3.22.: Information needed by the CChannel class to calculate boundary conditions. The information from CChannelMPath contains geometric and electromagnetic information on the multipaths. The CNavSource provides data on the dynamics and power of the signal. Refer to text for more information.

The `CIntegrator` object implements the common features of the ODE and SDE schemes as well as define the interface of both `CSDEBase` and `CODEBase`. The `CSDEBase` and `CODEBase` take care of the special code for the SDE and ODE, respectively. The classes `CIntegrator`, `CSDEBase` and `CODEBase` are general classes, designed to integrate (stochastic) differential equations and can therefore easily be used elsewhere.

The members and methods of `CIntegrator` are as follows:

- **Data**
  - State vector and its derivative
  - Buffers containing the time history of the results
  - Variables containing configuration and over-head information like the number of equations
- **Methods**
  - Initialization routines
  - Retrieval routines to access the time history of the results
  - `virtual Derivatives = 0`: Pure virtual function. A derived class defines the system of equations by overriding this function
  - `StepTo`: Integrates the system of equations to a given time

The `CChannelBase` inherits the properties of `CIntegrator` and implements special code needed for the implementation of a simulated receiver channel. This includes the interface to the `CChannelMultiPath` and `CNavSource` classes. These are needed in order to set the appropriate boundary conditions for the differential equation integrators. This situation is shown in figure (3.22). As shown in the figure, each channel needs

information on the multipaths and on the signal strength and dynamics. The multipath information (see section (3.4)) contained in `CCHannelMPath` provide data on the relative code and phase delay and relative signal strengths between the direct and reflected signals (SMR) for each multipath. The data from the `CNavsource` is needed to calculate the signal-to-noise ratio and to calculate the dynamics of the line-of-sight, which represents the signal dynamics.

The additional members and methods of `CChannelBase` are as follows:

- **Data**
  - Variables containing data on the multipaths
  - Configuration data for the signal structure, the satellite the channel is tracking, the signal power of the direct signal
- **Methods**
  - Initialization routines
  - `AcquireSignal`: Looks for a signal in at the current range. This is similar to the acquisition a real receiver must perform, except that there is no search in the frequency domain
  - `Derivatives`: In this override the structure of the tracking loops is defined
  - `Interpolate`: Performers the interpolation for the motion of the receiver as well as the multipaths, as described in the text
  - `R`: Returns the correlation function, defined by the configuration parameters. This can be rectangular chips, Raised Cosine chips, a BOC signal etc.

Correlators The correlation process is not implemented explicitly in SNSS . The form of the correlation function is provided analytically according to the results from section (2.2) and (2.5.1). Therefore the `CChannelBase` class provides a range of functions corresponding to eq. (2.134) and the equations referred to there.

The thermal noise is also obtained analytically and it depends on the signal structure and the loop filter parameters chosen as well as the radiated power (EIRP: Equivalent Isotropic Radiated Power) and distance of the navigation source. The signal-to-noise ratio is calculated as described in eq. (2.81) and then the noise terms in eq. (2.134) are calculated using eq. (2.135).

3.5.2. Processing

Mathematical model The processing of the (simulated) raw data in SNSS is done by the standard least squares single point positioning algorithm. There are of course other

more powerful positioning algorithms available, but it is not the goal with this simulator to assess positioning algorithms. On the contrary a simple, transparent positioning algorithm simplifies the analysis of the receiver performance.

The least squares single point positioning algorithm is described in every book on GPS and therefore the description here will be very brief.

The position can be obtained from the pseudo-ranges, P^k, the satellite coordinates, $\vec{R}^k = [x^k, y^k, z^k]^t$ and an approximate receiver position, $\vec{R}_0 = [x_0, y_0, z_0]^t$ by solving the (linearized) matrix equation below in a least squares sense:

$$\vec{l} = \mathbb{A}\vec{\delta x} \qquad \Rightarrow \qquad \vec{\delta x} = \left(\mathbb{A}^t\mathbb{A}\right)^{-1}\mathbb{A}^t\vec{l} \tag{3.3}$$

where $\{\vec{l}\}^k := P^k - \rho^k + \delta_{\text{clock}}$, $\rho^k := |\vec{R}^k - \vec{R}_0|$ and δ_{clock} is the local oscillator error. The matrix $\mathbb{A}$ is the so-called design matrix, defined by:

$$\mathbb{A} := \begin{bmatrix} a_x^1 & a_y^1 & a_z^1 & 1 \\ a_x^2 & a_y^2 & a_z^2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_x^n & a_y^n & a_z^n & 1 \end{bmatrix} \quad \text{with} \quad a_x^k := \frac{x_0 - x^k}{\rho^k},\ a_y^k := \frac{y_0 - y^k}{\rho^k},\ a_z^k := \frac{z_0 - z^k}{\rho^k} \tag{3.4}$$

The first three coordinates of the solution, $\vec{\delta x} = [\Delta x, \Delta y, \Delta z, \delta_{\text{clock}}]^t$ is the correction that must by applied to $\vec{R}_0$ to yield the measured position of the receiver, $\vec{R} = \vec{R}_0 + [\Delta x, \Delta y, \Delta z]^t$. The fourth coordinate is the clock correction (in meters).

This scheme above is then iterated a couple of times, replacing $\vec{R}_0$ with $\vec{R}$ each time, until $[\Delta x, \Delta y, \Delta z]$ is „small" (e.g. its length is smaller than a millimeter). Given the usual MEOs the scheme converges very rapidly even if Earth's center is used as an initial guess for $\vec{R}_0$. For pseudolites this may be more delicate and in a real situation one may have to use a converged GPS solution, *before* including the pseudolite measurements in the solution. In SNSS this is solved in a more pragmatic way; the „real" position is used as an initial guess for the measured position. This does not alter the converged result, it just ensures convergence and reduces the number of iterations.

To process the data generated by the receiver simulator a group of template classes was developed. An overview of the class hierarchy is given in figure (3.23). As the signals simulated in SNSS can lie on any frequency and there can be any number of them, the class structure must be very general.

Implementation The basic unit is the structure `BandData`. It contains the observations (code, phase and wave length) and represents the smallest unit of data in the hierarchy.

Each satellite can emit signals on any number of frequencies. For example the current state of GPS is two frequencies, but the GPS modernization envisions three frequencies.

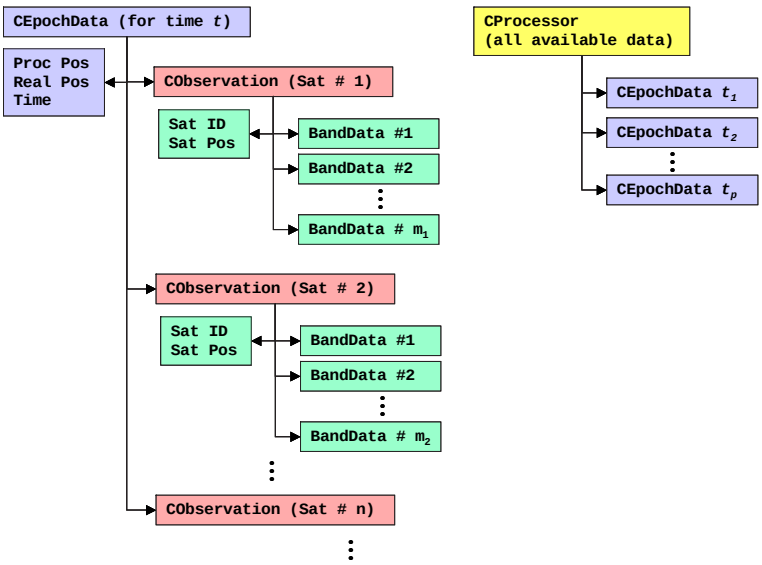

Figure 3.23.: Overview of the processing class hierarchy. The left part shows the class structure for the data corresponding to one epoch. The right part shows how the overall structure.

As of the time of this writing the number of frequencies is not fixed for Galileo yet. To meet this requirement of a flexible number of frequencies a class called `CObservation` was designed. It is derived from the STL template class `vector<BandData>` and is thus a vector with `BandData` structures as elements. Additionally, a unique identifier for the satellite (or pseudolite) and the current satellite position is stored in the `CObservation` object. Apart from constructors, destructors and the insertions and retrieval methods inherited from the STL base-class, `CObservation` has the following members and methods:

- **Data**
  - Vector of `BandData` (inheritance from the SLT `vector<T>`)
  - Satellite ID
  - Position of the satellite
- **Methods**
  - `GetDesignMatrixLine`: Retrieves the row of the design matrix, corresponding to the data available in `*this`
  - `GetLeftSideCode`: Retrieves the element of the left hand side of the matrix equation

It is convenient to collect all data belonging to one epoch into one container class. This is the role of `CEpochData`. It is derived from the STL template list `list<CObservation>` and is therefore a list of observations. A `CEpochData` instance contains all

relevant data retrieval methods necessary to conveniently calculate a position at a given time. The processed position and the real position as well as the time coordinate are stored in the `CEpochData` object. The relevant members and methods of `CEpochData` are

- **Data**
  - List of `CObservation` (inheritance from the SLT `list<T>`)
  - Time
  - Processed position
  - Real position
- **Methods**
  - `GetDesign`: Retrieves the design matrix and the left hand side of the position matrix equation
  - `GetProcessedPos`: Retrieves the position calculated by `CProcessor`
  - `GetRealPos`: Retrieves the reference (true) position
  - `GetTime`: Retrieves the time coordinate of the observation
  - `SetDatum`: Sets the time and reference position

The object actually processing the data to yield a position is the `CProcessor` class. This again is derived from the STL template list `list<CEpochData>`.

The `CProcessor` object is responsible for retrieving the data from `CEpochData` and calculate the measured position using the least-squares algorithm described by eq. (3.3). The relevant members and methods of `CProcessor` are

- **Data**
  - List of `CEpochData` (inheritance from the SLT `list<T>`)
  - Design matrix
  - Left side of the system of equations
- **Methods**
  - `Process`: Calculates a position for the current epoch and store the resulting position in the `CEpochData` element
  - `ProcessAll`: Calculates a position for all epochs and store the resulting positions in the corresponding `CEpochData` elements

The class `CProcessor` basically retrieves the necessary data from the `CEpochData`, calculates the position and puts the result back into `CEpochData`. Thus, in order to retrieve a calculated position for a given time t, the mechanism provided by the STL is used to browse through the `CProcessor`[4] and call GetTime, GetRealPos etc. of the `CEpochData` objects.

[4] remember: `CProcessor` is derived from `list<CEpochData>`.

4. Simulations

In this chapter results from the simulations using SNSS are presented. Before launching into full-scale simulations the models used in SNSS and their implementations must be verified as thoroughly as possible. After the verification various simulation scenarios are considered. The scenarios vary from very simple, constructed ones that are designed to demonstrate certain aspects of the receiver model, to scenarios that aim to simulate a realistic environment.

For the static case, i.e. where the position of the receiver is fixed in the WGS-84 system, the results are easily predicted, because the boundary conditions allow simple calculations and assumptions for the behavior of the receiver. Thus a verification is possible.

In the dynamic case, i.e. a moving receiver, the situation is a bit more complicated. The two main reasons are that a *dynamic* modeling of the receiver behavior is necessary and the second reason is the more complicated behavior of the boundary conditions. The domain where the dynamic modeling becomes necessary starts with receivers moving slower than 1 m/s, when multipath is present.

4.1. Verification of the Models and Their Implementation

This section considers mainly very simple scenarios. The scenarios are chosen such that the receiver behavior can be predicted by theory. Thus the modeling can be verified for those situations.

First the receiver behavior is verified in the presence of thermal noise. This is done for all forms of the signal structure developed in the previous chapters. Then the implementation and integration of the oscillator model into SNSS is considered.

The verification of the receiver behavior in the presence of multipath will be considered in the next section where simple scenarios are investigated.

4.1.1. Verification of Thermal Noise Behavior

The thermal noise behavior is verified in two ways. The first has to do with the dynamic modeling and shows that the model responds to thermal agitations as it should. This is

done by verifying the transfer function of the tracking loop. The second type of verification is simply to show that in certain simple setups, the simulator yields the expected results.

4.1.1.1. Verification of the Transfer Function of the Loops Driven by Thermal Noise

Before analyzing the receiver behavior for various signal structures the response of the tracking loops in the presence of thermal noise is investigated. This is achieved by injecting white noise into the loops and calculate the auto-correlation function of the resulting stochastic process.

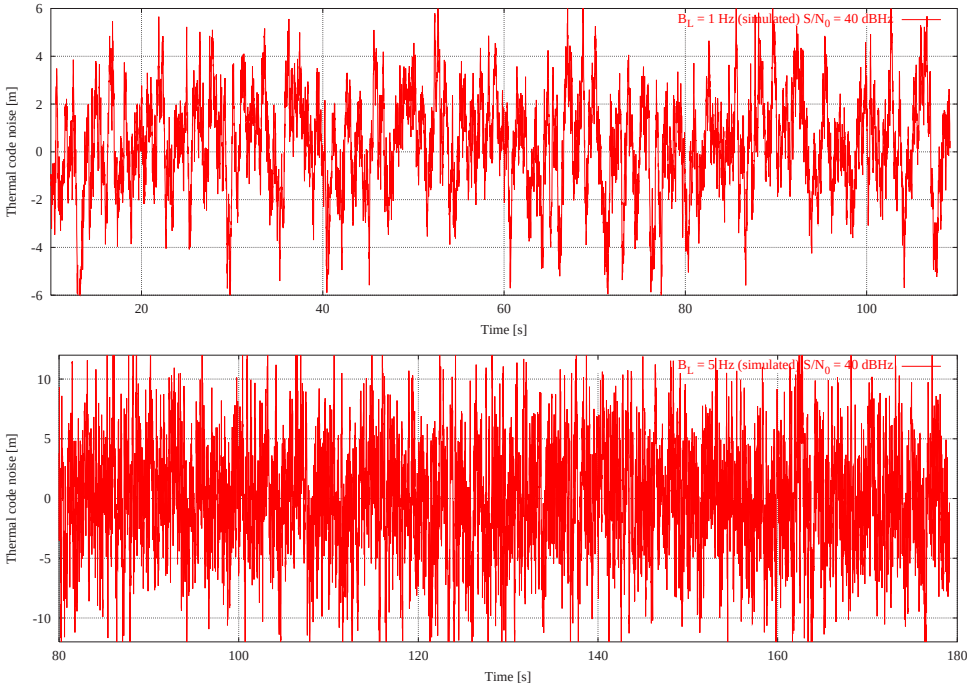

Figure 4.1.: Code tracking error as a function of time for C/N_0 = 40 dB-Hz. The upper figure corresponds to B_L^{dll} = 1 Hz and the lower to B_L^{dll} = 5 Hz.

The simulation set-up was very simple and it is shown schematically in figure (4.3). A receiver and a pseudolite were placed side-by-side without any environmental objects. Thus, the only source of error was due to thermal noise. The power level of the pseudolite was adjusted such that the desired signal-to-noise ratios were obtained. The output of the DLL tracking error was logged for 100 s and the tracking loops were read out every 0.011 seconds. The pre-detection time was T_p = 0.02 s, corresponding to the GPS data rate.

Two examples of the obtained data are shown in figure (4.1) and figure (4.2) for the code tracking loop and phase tracking loop, respectively. A careful look at those figures shows

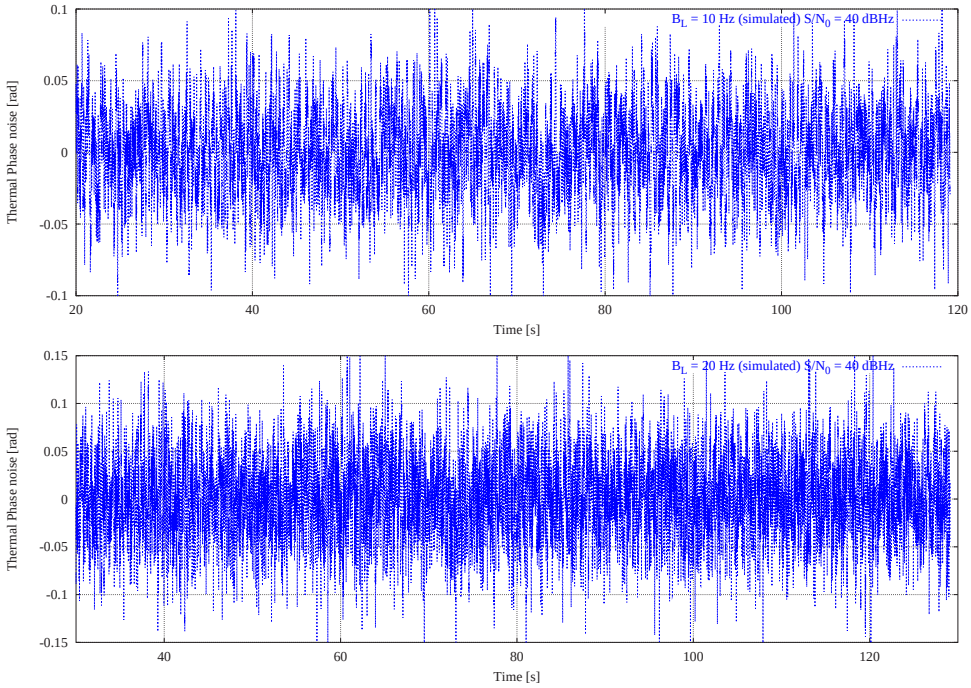

Figure 4.2.: Phase tracking error as a function of time for C/N_0 = 40 dB-Hz. The upper figure corresponds to B_L^{pll} = 10 Hz and the lower to B_L^{pll} = 20 Hz.

that the two stochastic processes must have a different auto-correlation function, as to be expected.

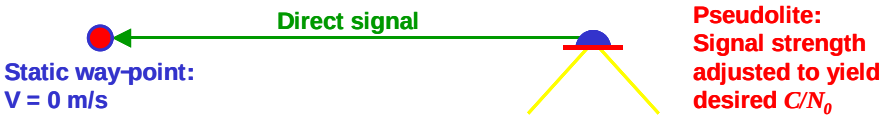

Figure 4.3.: Simulation setup for a simple static simulation

The output of the DLL corresponds to the DLL part of eq. (2.196), i.e. the first equation in the system of SDE. Under the assumption that the error from the carrier aiding can be ignored, its solution should approach a first order Markov process in the linear region of the discriminator. According to [Gel74] the auto-correlation function in this case is given by:

$$\frac{R(\tau)}{R(0)} = \mathrm{e}^{-\beta\tau} \qquad \text{where} \qquad \beta := 4B_L^{dll} \tag{4.1}$$

This is compared to the auto-correlation function of the noisy output of the DLL.

The solution of the PLL is also a stochastic process. In appendix C it is shown that the auto-correlation function of the output of the PLL driven by a white Gauß-ian noise

Configuration	B_L^{dll}	B_L^{pll}
Config 1	0.1 Hz	10 Hz
Config 2	2 Hz	10 Hz
Config 3	5 Hz	10 Hz
Config 4	1 Hz	10 Hz
Config 5	1 Hz	15 Hz
Config 6	1 Hz	20 Hz

Table 4.1.: Receiver configuration for the verification of the transfer function of the receiver model

process eq. (2.196) in the linear region is given by

$$\frac{R(\tau)}{R(0)} = \frac{\mathrm{e}^{-\omega_n \tau/\sqrt{2}}}{3}\left(3\cos(\omega_n \tau/\sqrt{2}) - \sin(\omega_n \tau/\sqrt{2})\right) \tag{4.2}$$

The auto-correlation functions of the stochastic processes[1], resulting from the simulation are shown in figure (4.4).

As seen in figure (4.4) the agreement between simulation and theory is excellent for low values of τ. For higher values of τ the simulation results and the theoretical results diverge somewhat. This is mainly due to the fluctuations in the simulated auto-correlation function. The region where auto-correlation function is zero there is some statistical noise, roughly proportional to $1/\sqrt{T/\tau_c}$, where T is the observation time and τ_c is the correlation time. Therefore fluctuation in the "zero-region" of the auto-correlation function should be about 0.3 in the our case, which fits the data. The divergence could also be contributed to the non-linear effects of the tracking loops.

[1]The auto-correlation functions discussed here should not be confused with the auto-correlation functions in 2.2. Here we are merely concerned with the stochastic analysis of the output of the tracking loops.

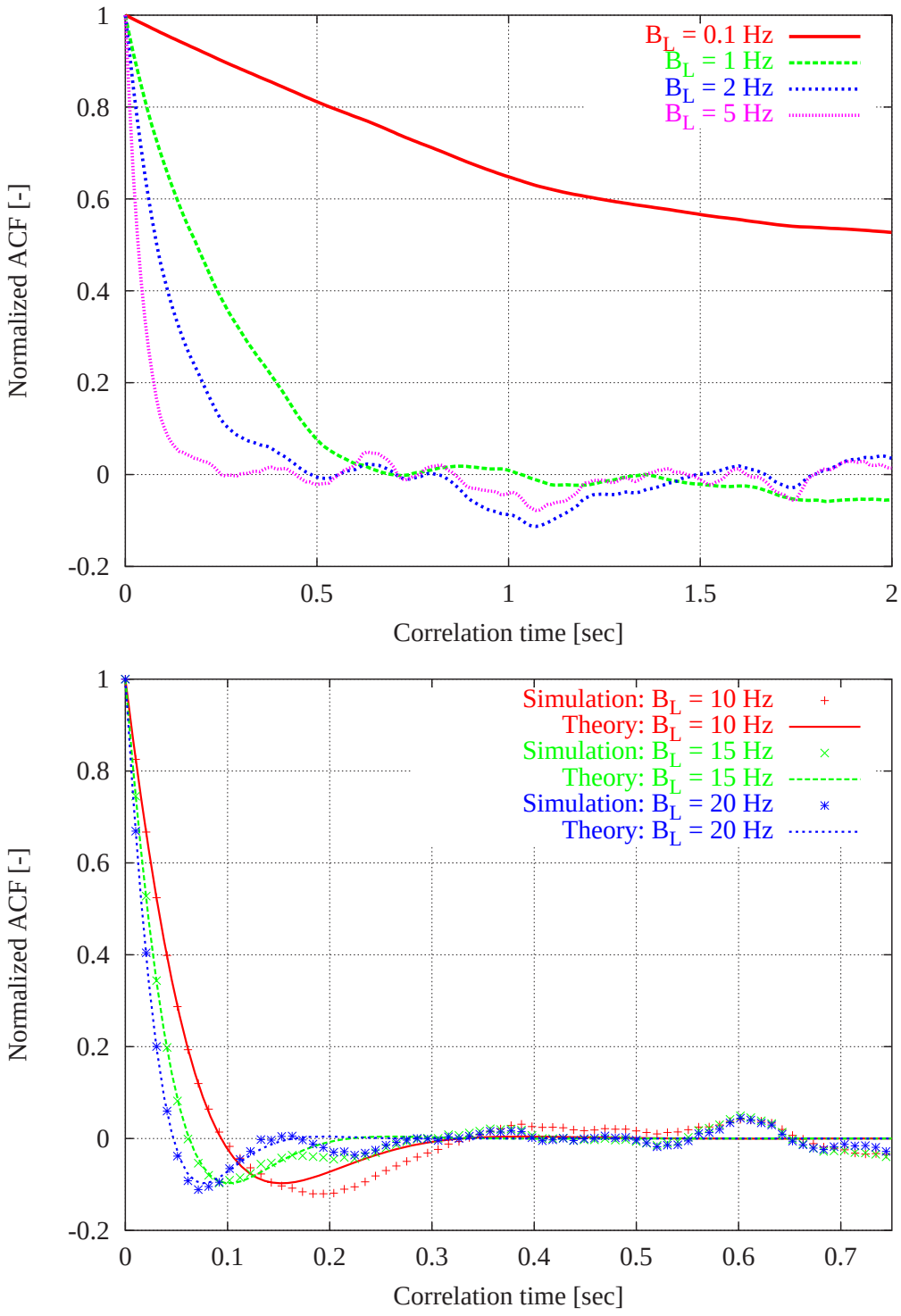

Figure 4.4.: Auto-correlation function of the tracking error for C/N_0 = 40 dB-Hz. In the left figure the ACF of the DLL is shown for the values B_L^{dll} = 0.1, 1, 2, and 5 Hz. In the right figure the ACF of the PLL is shown for the values B_L^{dll} = 10, 15 and 20 Hz.

4.1.1.2. Verification of the Code Noise Behavior for Rectangular Chips

In this section the implementation of the SDE is verified by comparing the output of SNSS to the theoretical expressions of thermal noise on code and carrier.

Infinite bandwidth To verify the noise behavior for rectangular chip shapes, a GPS L1 C/A code like signal structure was chosen (carrier frequency 1.575 GHz and chip length 293 m). The correlator spacing was d=1. The obtained data was compared to the well-known expression [Die96, Eis97]

$$\sigma^2_{\text{code}} = \frac{dT_c^2 B_L^{dll}}{2(C/N_0)} \left(1 + \frac{2}{T(C/N_0)(2-d)}\right) \tag{4.3}$$

for the non-coherent early-late DLL.

For the Costas PLL a similar expression is obtained for the noise on the phase:

$$\sigma^2_{\text{phase}} = \frac{B_L^{pll}}{(C/N_0)} \left(1 + \frac{1}{2T_p(C/N_0)}\right) \tag{4.4}$$

In figure (4.5) and figure (4.6) the results of the simulations are compared to eq. (4.3) and eq. (4.4). The simulations were performed for various values of signal-to-noise ratio (C/N_0) and various values of the loop filter bandwidth (B_L^{dll} and B_L^{pll}). Thus it was possible to verify the performance of the DLL and PLL regarding the behavior in those parameters. The simulations were performed for 4 values of C/N_0 = 28, 34, 40 and 46 dB-Hz. In the case of the code tracking loop, 4 values of B_L^{dll} = 0.1, 1, 2 and 5 Hz were simulated.

In figure (4.5) it is obvious that the agreement between theory and simulation is excellent. The theoretical results stated in eq. (4.3) and eq. (4.4) are very well established for this kind of signal-receiver configuration. For the phase error the values of the loop bandwidth were: B_L^{pll} = 10, 15 and 20 Hz.

Band-limited case Band-limited rectangular signals are also of interest. This applies for example to the new Galileo signals on E5a (L5) and E5b, where high chip-rates are planned [HGI+02] as well as to the GPS L5 signals [SD99]. For those signals the front-end bandwidth of the receiver may only let in the the first main-lobe of the signal spectrum.

At the time of this writing, there has been some discussion on the validity of the theoretical results regarding the code noise for low front-end bandwidth and narrow correlating signal-receiver configurations [BF00, BK00, Bet00b]. These discrepancies were mainly due to wrong normalization of the discriminator function of the tracking loop and incorrect treatment of the noise-level due to the band-limitation. In the SNSS receiver model

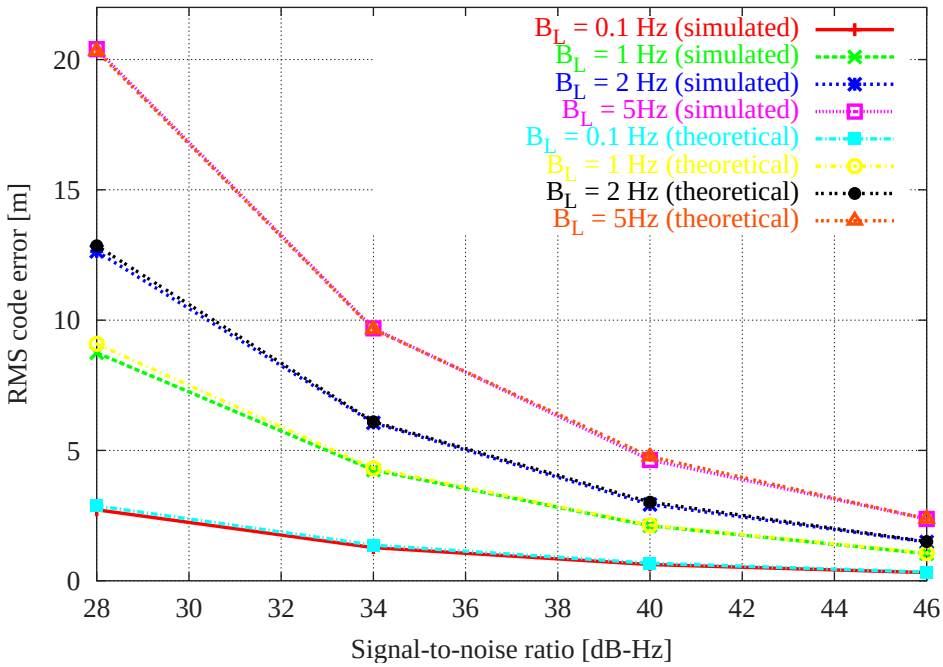

Figure 4.5.: RMS code tracking error for a rectangular, infinite bandwidth chip shaped signal as a function of signal-to-noise ratio.

these things are treated correctly and to demonstrate this the receiver-signal model was configured corresponding to Case 9 in [BF00]. The parameters for Case 9 are listed in table (4.2). This case was designed to maximize the divergence between the correct theoretical result and the older expressions. The signal receiver configuration parameters are shown in table (4.2).

Chip rate	1.023 Mcps
Receiver front-end bandwidth	4 MHz
Band-limiting parameter b	1.955
Correlator spacing	0.05 chips
DLL loop bandwidth B_L^{dll}	1 Hz

Table 4.2.: Signal-receiver configuration parameters for Case 9 in [BF00]

The correct expression for the code-noise of a receiver tracking a band-limited, rectangular signal is given by [BK00]

$$\sigma^2_{celp} = \frac{B_L \int_{-b}^{b} \mathrm{sinc}^2(\pi f) \sin^2(\pi f d) df}{(2\pi)^2 \frac{C}{N_0} \left(\int_{-b}^{b} f \, \mathrm{sinc}^2(\pi f) \sin(\pi f d) df \right)^2} \tag{4.5}$$

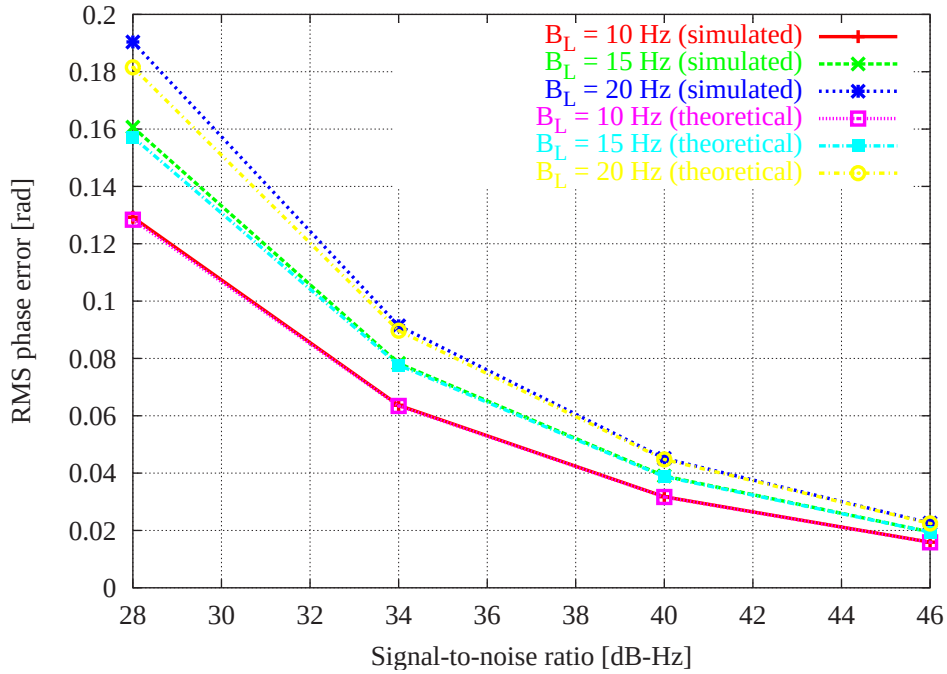

Figure 4.6.: RMS phase tracking error as a function of signal-to-noise ratio. The loop filter bandwidth used was B_L^{pll} = 10, 15 and 20 Hz.

for a coherent tracking loop.

For the non-coherent case, a squaring-loss term is added:

$$\sigma^2_{nelp} = \sigma^2_{celp}\left[1 + \frac{\int_{-b}^{b} \mathrm{sinc}^2(\pi f)\cos^2(\pi f d)df}{T\frac{C}{N_0}\left(\int_{-b}^{b} f\,\mathrm{sinc}^2(\pi f)\cos(\pi f d)df\right)^2}\right] \tag{4.6}$$

where d is the correlator spacing and b is the band limiting parameter. The subscripts $celp$ and $ncelp$ refer to coherent and non-coherent early-late processing, respectively.

In figure (4.7) the results from the simulations and the theoretical results are compared. For reference the theoretical infinite bandwidth expression is given. The infinite bandwidth approximation is obviously not valid for this signal structure-receiver configuration. However, the agreement between the correct theory and simulation is excellent. The correct theoretical values were also confirmed experimentally in [BF00].

4.1.1.3. Verification of the Code Noise Behavior for BOC Modulated Signals

To verify the correct implementation of the BOC signal structure, results from simulations were compared with the analytical results in [Bet00b]. As the theory developed in

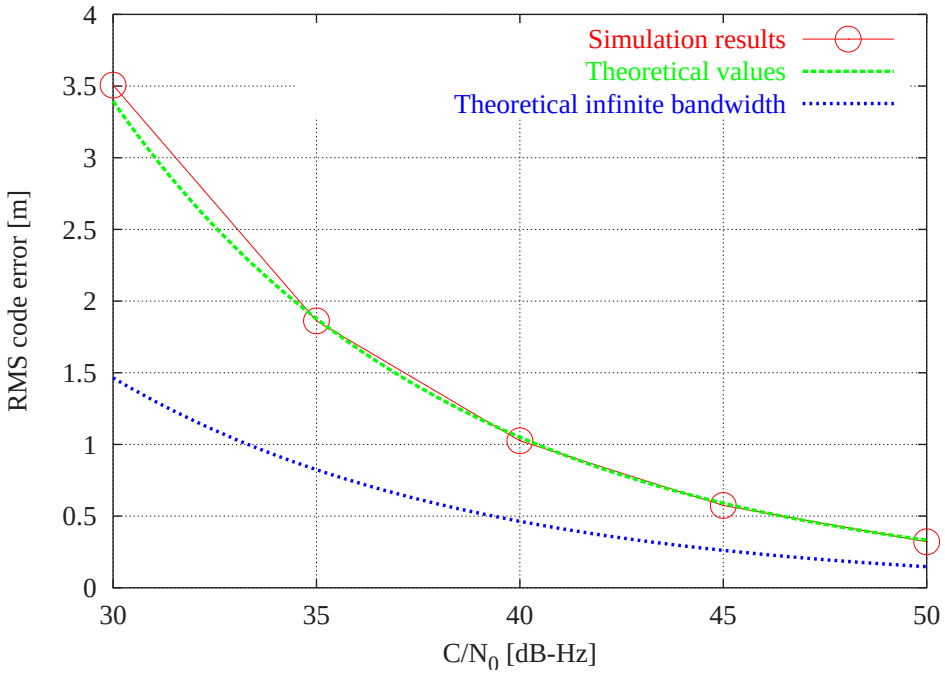

Figure 4.7.: RMS code tracking error as a function of signal-to-noise ratio for case 9 in [BK00]. For reference the infinite bandwidth approximation is also shown. The parameters for this simulation are listed in table (4.2).

[Bet00b] were also confirmed experimentally, a match between the theory and the simulation results is therefore considered an indirect experimental verification of the simulation results.

The expression for the code-noise is given in [Bet00b] and is found to be

$$\sigma^2_{elp} = \frac{B_L(1 - B_L T_p/4)\int_{-b}^{b} G_s(f)\sin^2(\pi f d)df}{\frac{C}{N_0}\left(2\pi\int_{-b}^{b} f G_s(f)\sin(\pi f d)df\right)^2} \tag{4.7}$$

for the coherent case. For the non-coherent early minus late case the code noise is given by

$$\sigma^2_{nelp} = \sigma^2_{elp}\left[1 + \frac{\int_{-b}^{b} G_s(f)\cos^2(\pi f d)df}{T_p\frac{C}{N_0}\left(\int_{-b}^{b} G_s(f)\cos(\pi f d)df\right)^2}\right] \tag{4.8}$$

The parameters for the signal structure-receiver configuration are listed in table (4.3).

The RMS code tracking error for the trials configured according to table (4.3) is shown in figure (4.8). The signal used was an M-code like signal, i.e. a BOC(10,5). The bandwidth of the receiver front-end was set to 24 MHz and the correlator spacing was 40 ns (corresponding to d=0.2032). The tracking loop bandwidth was 1 Hz. This signal-receiver

Chip rate	5.115 Mcps
Square wave rate	10.23 Mcps
Receiver front-end bandwidth	24 MHz
Band-limiting parameter b	0.5865
Correlator spacing d	0.2023 (40 ns)
DLL loop bandwidth B_L^{dll}	1 Hz

Table 4.3.: Signal-receiver configuration parameters used for the verification of the BOC signal structure. The same parameters were used in [Bet00b].

configuration was chosen, because it matches the configuration that has already been verified experimentally in [Bet00b] and represents a realistic set of parameters for M-code tracking in terms of bandwidth, correlator spacing and so on.

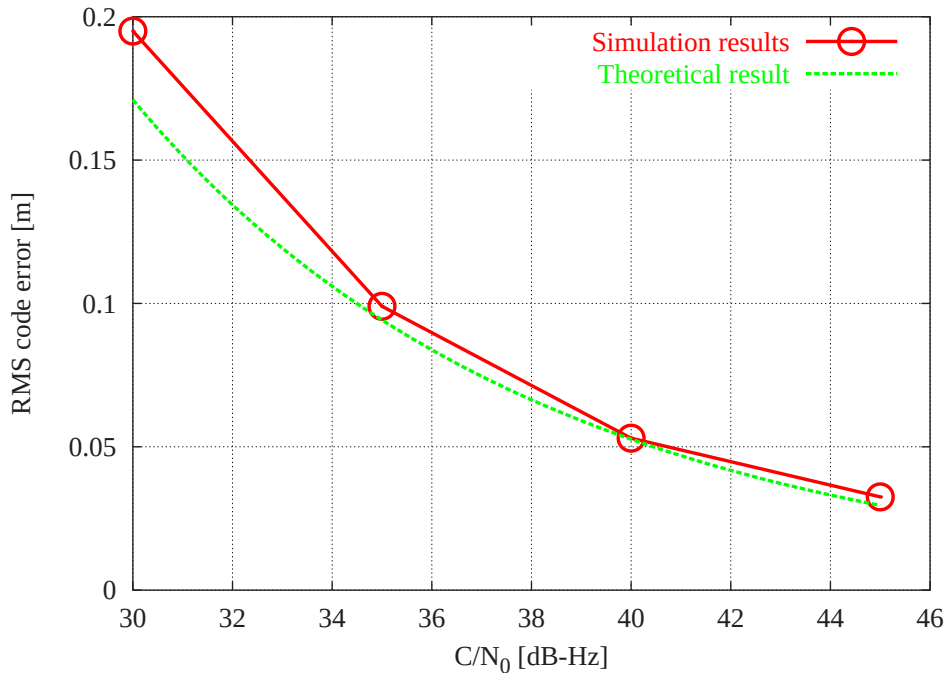

Figure 4.8.: RMS error of a DLL tracking a BOC modulated signal as a function of carrier-to-noise ratio. Loop filter bandwidth BL=1 Hz.

As can be seen from the figure the simulated and theoretical values match very well. However, the simulated values seem slightly higher than the theoretically predicted ones. A similar trend can be observed in [Bet00b], figure 21. As the deviation increases with decreasing signal-to-noise ratio, it is plausible that it is caused by non-linearities of the discriminator.

4.1.1.4. Verification of the Code Noise Behavior for SRC Signals

For the raised cosine chip shape the analytical expression for the code phase noise is given by:

$$\sigma^2_{\text{src}} = \Gamma^2 T_c^2 \frac{B_L^{dll}}{2C/N_0}\left(1 + \frac{\rho}{T_c C/N_0}\right) \tag{4.9}$$

where B_L^{dll} is the loop bandwidth, C/N_0 is the carrier-to-noise ratio. The expression for the parameters Γ and ρ are rather complex and depend on the roll-off factor α and the shape of the chip. For the SRC signal with roll-off factor of $\alpha = 0.2$ their numeric values are $\Gamma = 0.7$ and $\rho = 0.9$. A more detailed explanation of these two parameters can be found in [GLV93]. The parameters used are listed in table (4.4)

Chip rate	3.0 Mcps
Receiver front-end bandwidth	4 MHz
Roll-off parameter α	0.2
Correlator spacing	1.0
Pre-detection bandwidth	0.02 s

Table 4.4.: Signal-receiver configuration parameters for the SRC code noise verification

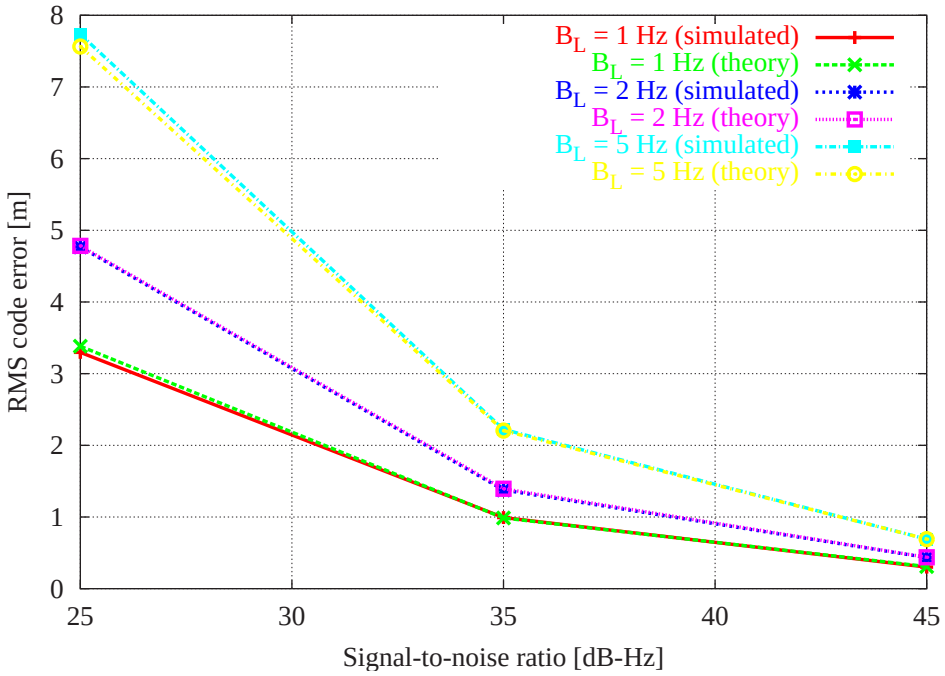

Figure 4.9.: RMS error of a non-coherent receiver tracking a raised cosine signal

In figure (4.9) a comparison between theory and simulation is shown.

4.1.2. Verification of the Integration of the Oscillator Model

In this section the integration of the oscillator model into the tracking loops is verified. In section (2.5.3) the oscillator model itself was confirmed by comparing the empirical Allan deviation of the oscillator model to the theoretical expression for the Allan deviation. However, the correct integration within the framework of SNSS must also be verified.

To achieve this the very simple simulation set-up already shown in figure (4.3) was used. As before the receiver was positioned next to a pseudolite. The power of the pseudolite was set to 1000 W. Of course no realistic receiver could take that kind of power, but in the simulation this was done to boost the signal-to-noise ratio to astronomic values and thus effectively exclude the influence of the thermal noise. As no multipath and no signal dynamics of any kind were present, the resulting error was solely due to the oscillator instabilities.

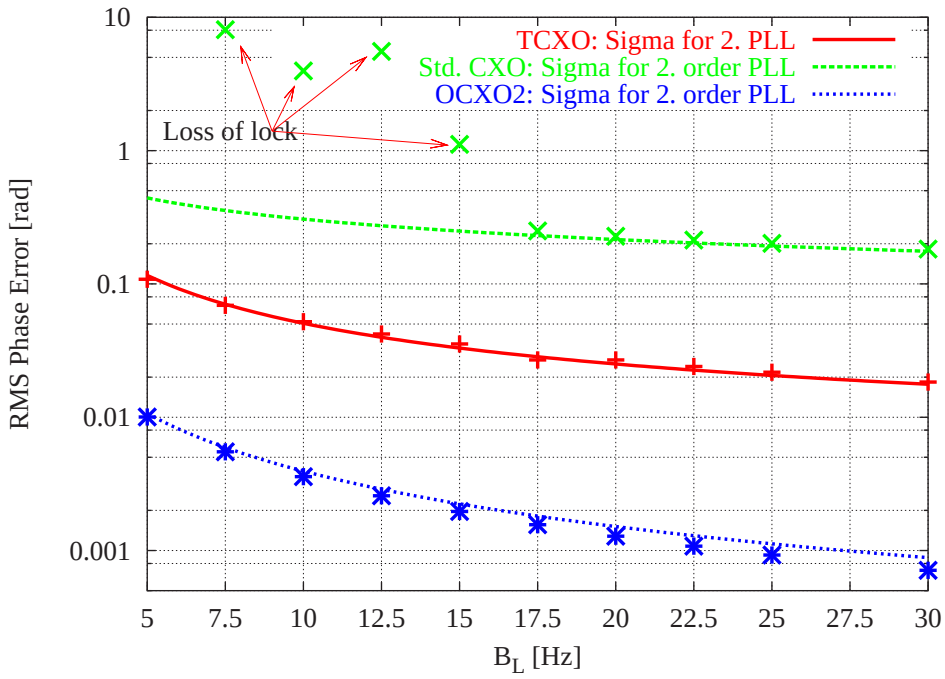

Figure 4.10.: Verification of the integration of the oscillator model into SNSS . The oscillators simulated were CXO, TCXO and OCXO2. For the CXO the PLL lost lock for a loop bandwidth of 15 Hz and smaller.

For each point in figure (4.10) the simulation was run for 10 minutes and then the standard deviation of the phase error was calculated.

In figure (4.10) the results from simulations using three different oscillators are compared to the linear theory analysis obtained in [IE02]. The oscillators tested were the CXO, the TCXO and the OCXO2 from table (2.1). As was pointed out in section 2.5.4 the main

contribution to the tracking error for these particular oscillators comes from different types of processes of the oscillator instabilities. For the CXO the white frequency noise is dominant, for the TCXO the flicker term and for the OCXO2 the random walk is the largest term. These oscillators were chosen in order to be sure the oscillator model is properly accounted for in all cases.

In general the results of the simulations match the theory very well as can be seen in figure (4.10).

For the CXO the phase lock loop lost lock for loop bandwidths (B_L) of 15 Hz and smaller, as indicated in figure (4.10). This is consistent with the 15° rule-of-thumb [IE02]. For higher values of B_L the simulated tracking error is slightly higher than the theoretically predicted one. The theoretical curves are based on linear analysis of the transfer function of the PLL thus, the deviation from the predicted value may be attributed to non-linearities, as the PLL is operating in a region just before loosing lock.

For the TCXO one would have expected some slight deviations because of the approximation for the flicker term developed in previous sections, but as can be seen in the figure the agreement is very good. The same can be said for the OCXO2.

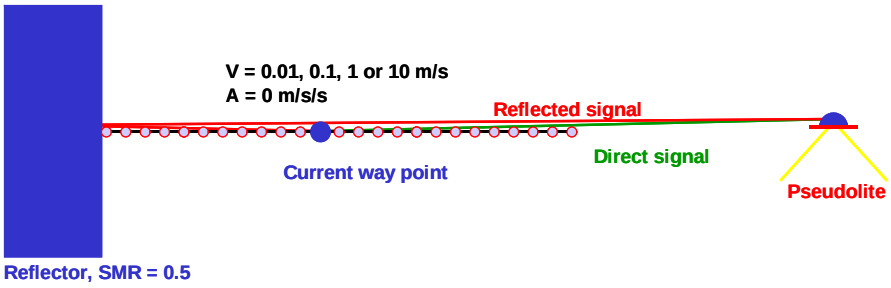

Figure 4.11.: Schematic simulation scenario for the multipath simulation

4.2. Simple Scenarios

In this section simple scenarios are used to demonstrate the impact of multipath, in particular in dynamic situations. However, some low dynamic simulations were performed to verify the impact of multipath on the receiver simulation, but these simulations are also of interest when comparing different signal structures.

4.2.1. Comparing Signal Shapes with Respect to Multipath Performance

In this section the performance of the signal structures introduced previously will be analyzed in the simulator. In order not to loose track of the relevant features a very simple

scenario is analyzed for each of the signal structures.

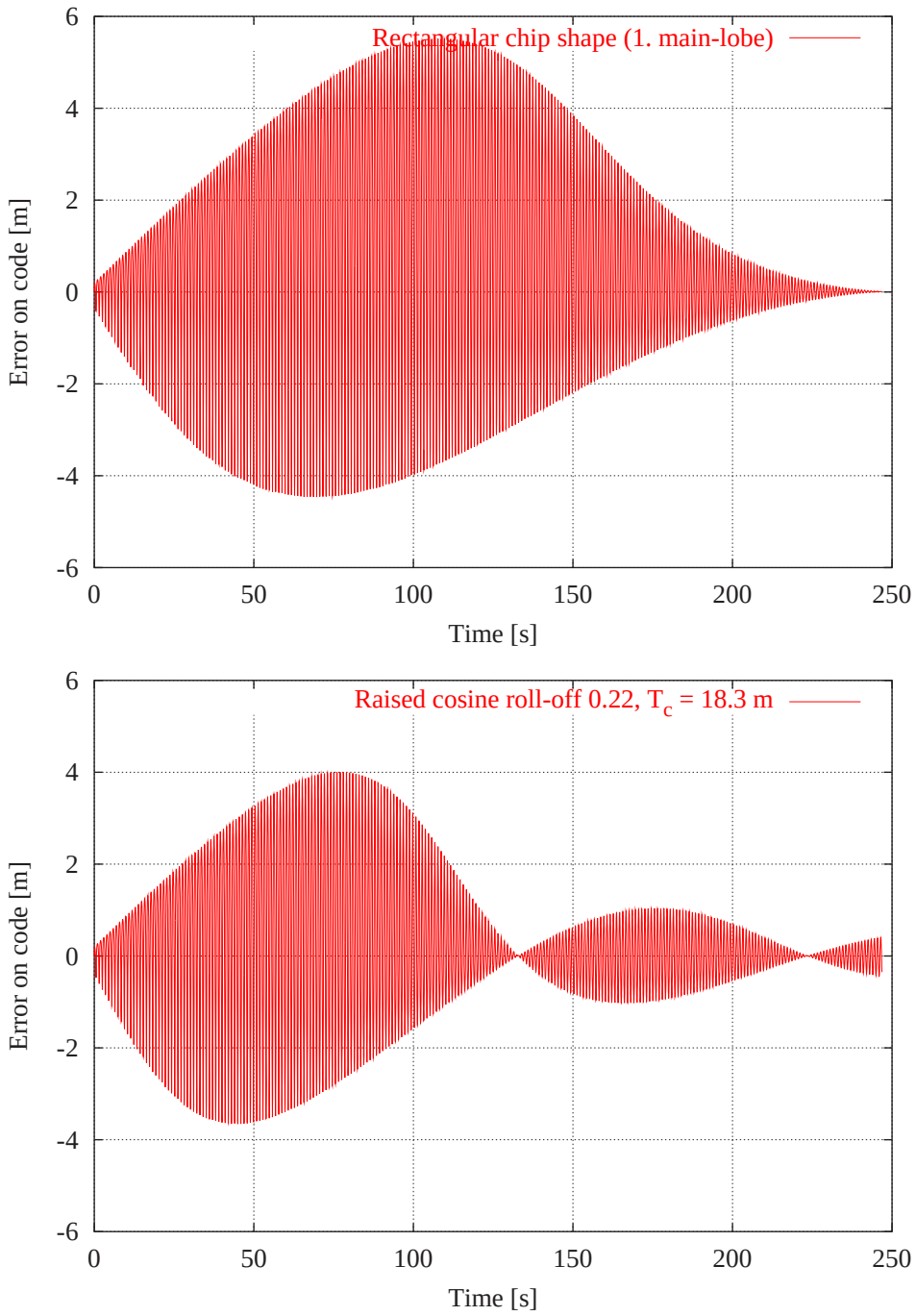

Figure 4.12.: Multipath error on the code for two different kinds of signal structures. Upper figure: Rectangular signal with bandwidth 20.46 MHz. Lower figure: Raised cosine with α=0.22, configured to a bandwidth of 20.46 MHz. For further parameters, see text and table 4.5.

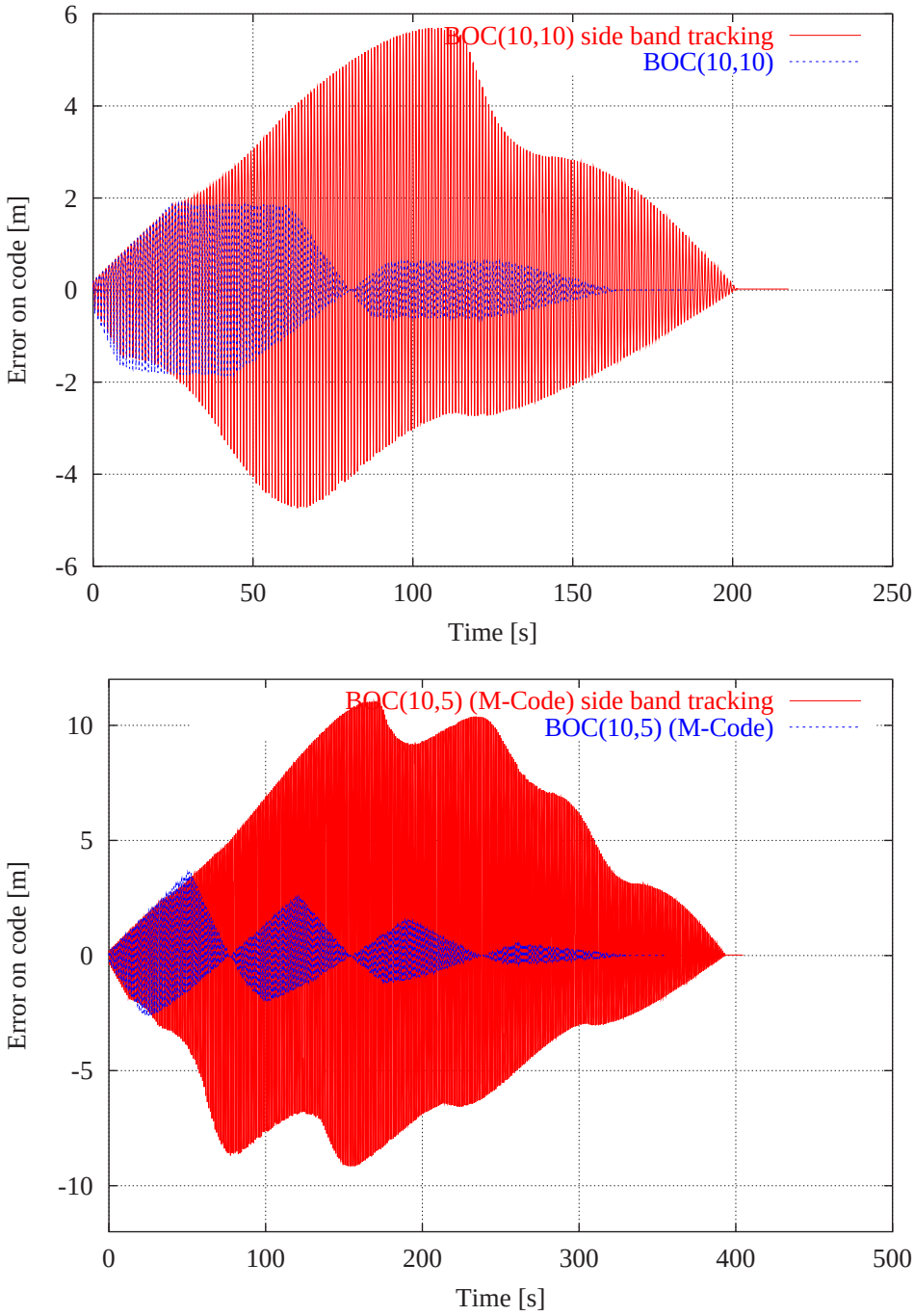

Figure 4.13.: Multipath error on the code for two different kinds of signal structures. upper figure: BOC(10,10) using full auto-correlation (green curve) and side-band tracking (red curve). Lower figure: BOC(10,5) using full auto-correlation (green curve) and side-band tracking (red curve). For further parameters, see text and table 4.5.

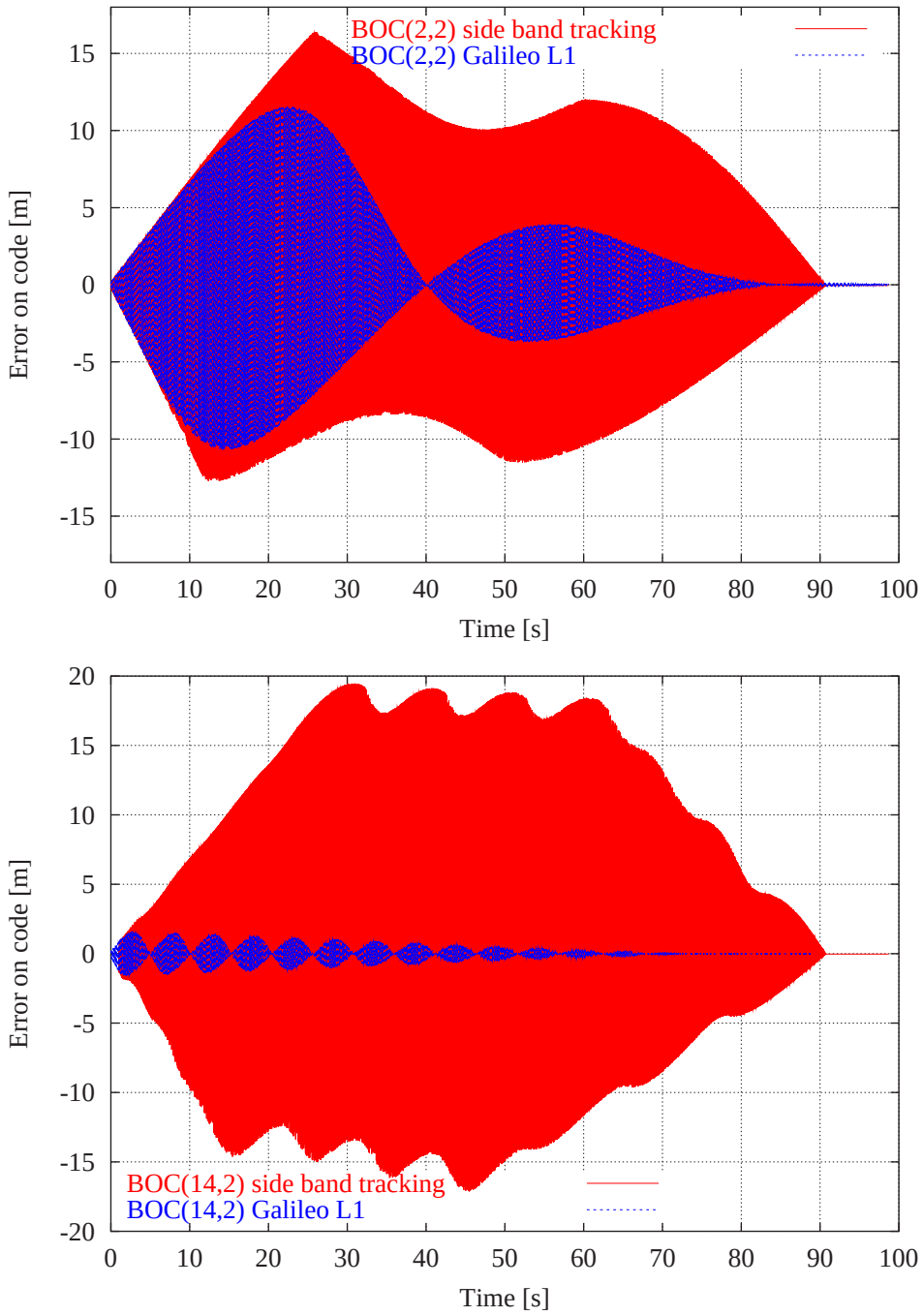

Figure 4.14.: Multipath error on the code for the BOC(2,2) and BOC(14,2) and side-band tracking of a BOC(2,2) and BOC(14,2) signal structure. The red curves (in both plots) represent the side-band tracking. The blue curves correspond to the full auto-correlation function tracking and the green curves are bandwidth limited versions. In the BOC(2,2) case (upper figure) the bandwidth was 8.184 MHz MHz and in the BOC(14,2) the bandwidth was 32.736 MHz. For further parameters, see text and table 4.5.

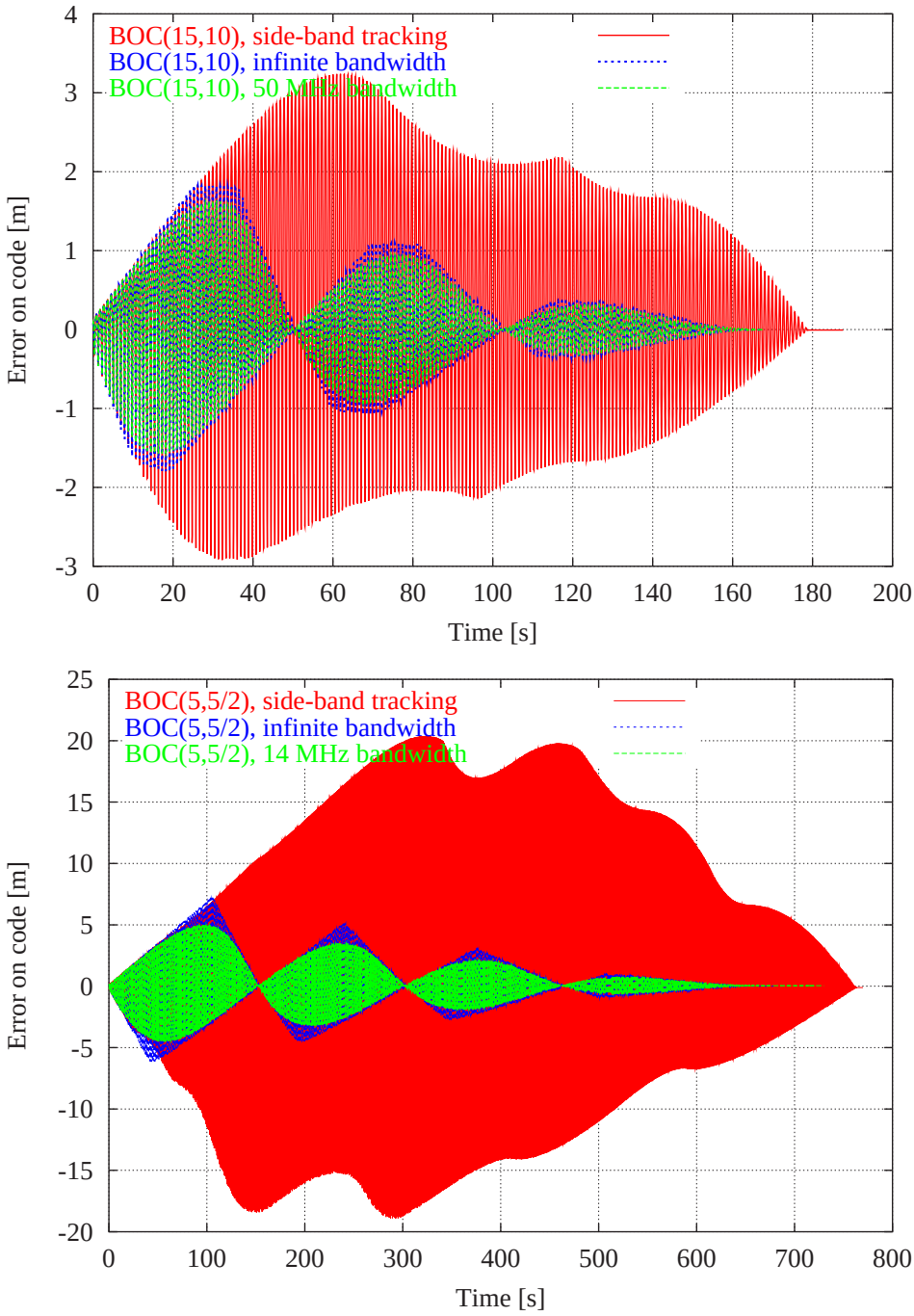

Figure 4.15.: Multipath error on the code for the BOC(15,10) and BOC(5,5/2) and side-band tracking of a BOC(15,10) and BOC(5,5/2) signal structure. The red curves (in both plots) represent the side-band tracking. The blue curves correspond to the full auto-correlation function tracking and the green curves are bandwidth limited versions. In the BOC(15,10) case (upper figure) the bandwidth was 50 MHz and in the BOC(5,5/2) the bandwidth was 14 MHz. For further parameters, see text and table 4.5.

The setup is shown schematically in figure (4.11). A reflector (the blue box to the left in the figure) with a constant SMR ratio of 0.5 was used. The vehicle moved slowly away from the building with a constant speed of 0.1 m/s. To suppress the multipath fading, the loop filter bandwidth of the DLL was set to 5 Hz. In the next section the impact of the dynamics and loop filter bandwidth will be examined.

The signal structures and the corresponding receiver configuration investigated in this section are shown in table (4.5). The rectangular band-limited signal resembles the Galileo signal on E5a and E5b as well as the GPS P-code and the future GPS III signal on L5 (E5a). The raised cosine signal corresponds to a signal, occupying the same bandwidth as a rectangular signal with a chip length of 29 m confined to the first main-lobe.

The BOC(15,10) signal corresponds to the latest proposal of the EC STF for a coherent overlay on E5a (L5) and E5b [HGI+02]. This is an extremely wide-band signal, occupying more than 50 MHz of bandwidth by the first main-lobes only. For reference the GPS M-code signal is also shown. The same signal structure (among other) is under discussion for the Galileo Public Regulated Service (PRS).

As was shown in the previous chapters, it is theoretically possible to track the side-lobe of the BOC signals. This is of course a sub-optimal use of the signal, but due to the aforementioned problems with the acquisition of the correct peak in the auto-correlation function, it may serve as an intermediate stage towards tracking the full auto-correlation function.

It is also possible to use the side-band tracking as a monitoring mechanism; a channel tracking a side-band could be used to compare it to the tracking of the full auto-correlation function and thus raise an alarm if the difference in the pseudoranges significantly indicates that the wrong auto-correlation peak is being tracked. This is of course only possible if the performance of the side-band tracking is good enough to distinguish between two adjacent tracking points of the auto-correlation function. This, again, puts a constraint on the value of n in BOC(nx, x).

Finally, side-band tracking presents a possibility to implement a low-cost receivers, designed to only track the side-band. This reduces the necessary bandwidth and the sampling rate of the receiver, hence reducing costs and power consumption. Looking at the plots for the band-limited rectangular signal and the raised-cosine signal in figure (4.12), we see that the maximal multipath error for the raised cosine shape is smaller than that for the rectangular pulse shape. However, the multipath errors for large delays for the raised cosine chip-shape remain. This means that a reflections from far away objects will be injecting noise into the receiver, no matter how far away they are. This suggests that in the case of very stringent out-of-band emission requirements on the signal for whatever reason (technical reasons in the payload or on the receiver side or for regulatory reasons), the raised cosine chip-shape may have advantages. However, that is very unlikely to be the case. The satellites emit a very weak signal and are thus unlikely to disturb services located in the neighboring frequency bands. This has been the situation with GPS since the beginning; the C/A code is emitted unfiltered, overlaying the GPS P-code and does

Chip shape	Chip length	Bandwidth	Corr. Space	Comment
Rectangular	29.3 m	20.46 MHz	$1T_c$	GPS-P, Gal. E5a and E5b
Raised cosine	18.3 m	20.46 MHz	$1T_c$	Roll-off= 0.22
BOC(15,10)	29.3 m	∞	$0.25T_c$	Galileo E5a and E5b
BOC(15,10)	29.3 m	50 MHz	$0.25T_c$	Galileo E5a and E5b
BOC(10,10)	29.3 m	∞	$0.25T_c$	Galileo E5a and E5b
BOC(10,5)	58.6 m	∞	$0.25T_c$	GPS M-code, Gal. PRS
BOC(5,5/2)	117.2 m	∞	$0.25T_c$	GPS M-code, Gal. PRS
BOC(2,2)	146 m	8.184 MHz	$0.1T_c$	Galileo L1
BOC(14,2)	146 m	32.736 MHz	$0.0357T_c$	Galileo E2-L1-E1
BOC(15,10)	29.3 m	∞	$0.5T_c$	Side-band tracking
BOC(10,10)	29.3 m	∞	$0.5T_c$	Side-band tracking
BOC(10,5)	58.6 m	∞	$0.75T_c$	Side-band tracking
BOC(5,5/2)	117.2 m	∞	$0.75T_c$	Side-band tracking
BOC(2,2)	146 m	∞	$0.5T_c$	Side-band tracking
BOC(14,2)	146 m	∞	$0.75T_c$	Side-band tracking

Table 4.5.: Overview of the signal structures used in this section.

not disturb the GPS P-code. The wide bandwidth of the GPS C/A code signal has been successfully utilized by civil GPS receiver manufacturers and numerous multipath mitigation techniques have been developed, which rely on the large bandwidth of the C/A code signal. Not only does the large bandwidth offer many ways to mitigation techniques, but the noise performance can also be substantially improved.

For a communication system, which has a two-way link to the satellite, the situation is different. If the up-link (i.e. from the receiver to the satellite) had a very slight spill-over into a neighboring service that was trying to receive a signal from a satellite, the power-level of the spill-over signal might become large, due to the much smaller distance. Thus, the spill-over situation is entirely different in a system utilizing a two-way link like a communication system than for a system using only a down-link like a satellite navigation system.

The plots in figure (4.13) show a comparison between the BOC(10,10) and the BOC(10,5) signals. Note that the scales of the BOC(10,10) plot is half the scale of the BOC(10,5) signal. The signals were tracked using a correlator spacing of d = 15 m in the BOC(10,5) case and d = 7.5 m in the BOC(10,10) case. The difference in correlator spacing reflects the larger bandwidth of the BOC(10,10) signal. The superior performance of the BOC(10,10) signal is evident. It should also be noted that the tracking device used in this simulation is not optimized for a BOC-type signal. It has been suggested to use more correlators, one very early and one very late, to take advantage of the structure of the auto-correlation function and thus increase the performance even further [BBC$^+$00].

The plots in figure (4.13) also show the side-band tracking of the BOC(10,10) and BOC(10,5)

signals, respectively. In figure (4.14) the side-band tracking of a BOC(2,2) and a BOC(14,2) are shown. To a rough approximation the side-band tracking corresponds to a rectangular signal with the same chipping rates as the BOC signals. The curious wavy structure of the multipath envelope is caused by the wavy structure of the correlation function (see section 2.2.8). Note that as n in BOC(nx,x) becomes larger, the larger will the difference between full auto-correlation tracking and side-band tracking become. This is easily explained as the bandwidths involved scale with n. This applies to multipath performance as well as to the thermal noise performance.

In figure (4.14) the multipath error on the code phase for a receiver tracking the full auto-correlation function and the side-band of the BOC(2,2) and BOC(14,2) are shown.

Note that the BOC(2,2) and BOC(10,10) differ only in the higher chip rate and thus the BOC(10,10) would be just a scaled-up version of the BOC(2,2). The difference between figure (4.14) and the lower plot in figure (4.13) is due to the different relative bandwidth and the difference in correlator spacing (see table 4.5).

In figure (4.15) the multipath envelopes for the BOC(15,10) and the BOC(5,5/2) are shown. The BOC(15,10) signal has the largest bandwidth of the signals analyzed in this thesis. In the figure the full auto-correlation is shown both for infinite bandwidth and for a bandwidth of 50 MHz as well as for side-band tracking (see table 4.5 for more parameters). In analogy to the BOC(10,10) and BOC(2,2) case, the BOC(5,5/2) is a scaled-down version of the BOC(10,5).

Receivers not able to sample the entire bandwidth of the BOC signal can take advantage of the side-band signal. In the BOC(15,10) case the tracking of the side-band is comparable to tracking the GPS L5 (or GPS P-code) signal. Comparing the plot of the side-band tracking of the BOC(15,10) signal with the GPS P-code like signal above, we see that the multipath performance is comparable, although it is slightly better in the side-band case. Apart from the multipath issue the side-band tracking "picks up" power from the other side-band coherently. A consequence of this is firstly the structure in the multipath envelope seen above and secondly the coherent power from the other side-band improves the signal to noise ratio slightly.

4.2.2. Multipath Dynamic Fading

The goal of this section is to show how important it is to use dynamic modeling for the receiver in kinematic applications, when assessing the multipath properties of the receiver. First the classical multipath fading is examined for code and phase. This is the fading caused by the low pass filtering characteristics of the tracking loops. Then the effect of the fading caused by the averaging in the pre-detection process is demonstrated. This fading is caused by the averaging resulting from the correlation process and not by the filtering due to the tracking loop characteristics. This was discussed in section 2.5.1.

In this scenario the same simple set-up was used as in section 4.2.1, figure (4.11).

To eliminate the dynamics of the satellites, a pseudolite was used. Thus the signal dynamics are caused by the motion of the receiver only. A satellite in orbit would, however only increase the signal dynamics slightly. The receiver moves away from the wall, perpendicular to it and directly towards the pseudolite with a constant velocity of 0.01, 0.1, 1 and 10 m/s for each case. The geometry was chosen as shown to obtain a linear relationship between geometric multipath delay and time. This way the familiar multipath envelope curves should be reconstructed for the extremely slow case and the curves of the successive simulation are therefore comparable to the envelop curves we are used to. The four cases have identical geometry, except for the speed. The acceleration is zero for all scenarios.

Multipath on the Code Measurement If the pre-detection fading is ignored, the error in range and phase are obtained as shown in figure (4.16). In this figure the error on the DLL (i.e. the code measurement) is shown. As was pointed out in the section on the receiver model, the tracking loop acts as a low-pass filter, where the loop bandwidth B_L controls the cutoff frequency. The tracking loops used here were a first order, non-coherent DLL loop coupled with a second order PLL. Carrier aiding was enabled and the filter bandwidths were B_L = 1 Hz for the DLL and B_L = 20 Hz for the PLL. The correlator spacing was set to d = 1.

The signal emitted by the pseudolite is a GPS C/A code like signal on L1; infinite bandwidth, chip length of 293 m and carrier frequency 1.575 GHz.

As one would expect, the multipath error for the slow case (i.e. v = 0.01 m/s) reflects the multipath error for the static case. This is equivalent to a simple root finding of the S-curve.

For the other three cases we see the impact of the signal dynamics. For the second case (i.e. v = 0.1 m/s), which is an extremely slowly moving receiver (a pedestrian walks with a speed of approximately 1-2 m/s), the lower branch of the multipath error is more affected by the receiver filtering. This is plausible if one considers the detailed form of the multipath error; it has rather sharp peaks in the negative region, while in the positive region the peaks are more rounded. This is shown in figure (4.17). The sharp peaks correspond to higher frequencies, which get filtered (low-pass characteristics of the tracking loops). This explains the fact that the minimum/maximum multipath error is reduced to approximately -26 m (corresponding to the -75 m in the static case) and 55 m (as opposed to the 75 m in the static case).

As the speed picks up and we reach speeds comparable to normal walking speed, we notice that the multipath error is always positive. For a velocity of 10 m/s the multipath has almost stopped oscillating and approaches a line. The maximum variation in this case is about 1 m at the maximum, which is about 20 m.

In general we can distinguish between two kinds of multipath. The first category are reflections from objects stationary with respect to the receiver. For the stationary receiver

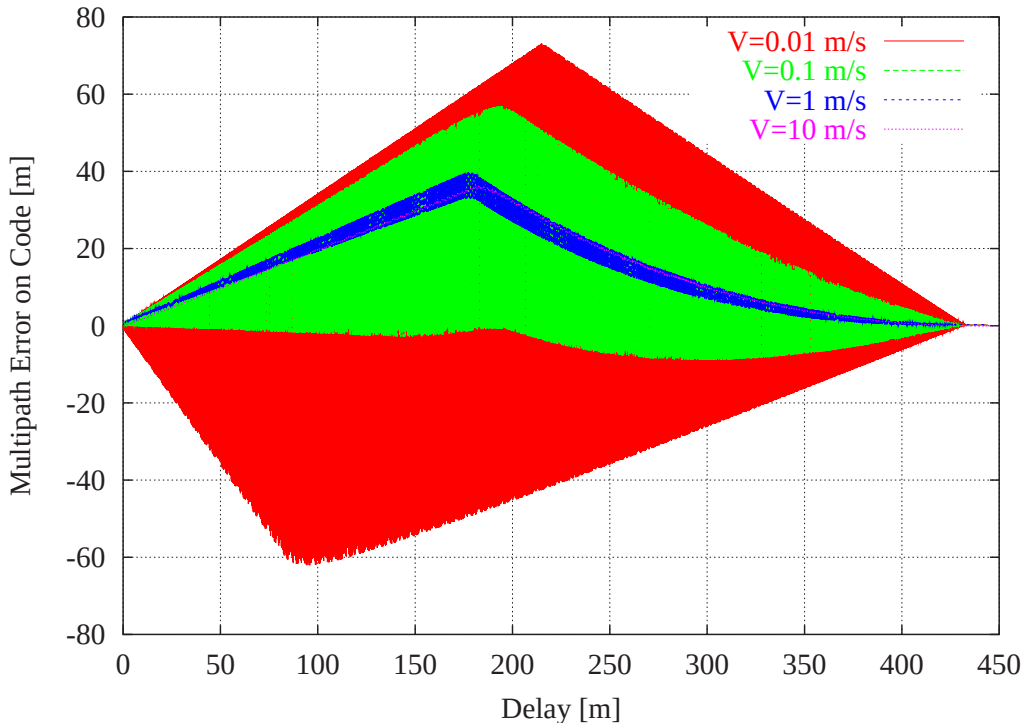

Figure 4.16.: Multipath error as a function of time. The geometric setup was identical in all cases except for the velocity. Starting from the top left figure the speed was 0.01 m/s, 0.1 m/s, 1 m/s and 10 m/s for the bottom right figure.

reflections from buildings or the ground for example would be of that kind. For a moving receiver those are reflections from the vehicle itself. If the change in orientation is not too violent (the vehicle is not rotating too fast) then the change in geometric multipath for reflections off the vehicle itself is not large and one may expect to see multipath which resembles the static case.

The second are signals reflected off objects moving with respect to the receiver. For a stationary receiver that would be reflections from moving objects like cars and trains passing the receiver. For a moving receiver it would be reflections from buildings and other stationary objects, but also from other moving vehicles.

There is a big difference in the nature of the two types of reflections. The dynamics of the stationary multipaths are governed by the satellite motion, thus they exhibit a very small frequency offset relative to the direct signal; in the region of 1-10 mHz [Eis97]. As soon as the receiver starts moving, even with a moderate velocity of 1 m/s, the Doppler difference is in the region of 1-10 Hz.

Obviously, most kinematic applications involve speeds comparable to or higher than 1 m/s. It is therefore questionable whether the classic multipath envelopes describe the situation adequately in the kinematic case.

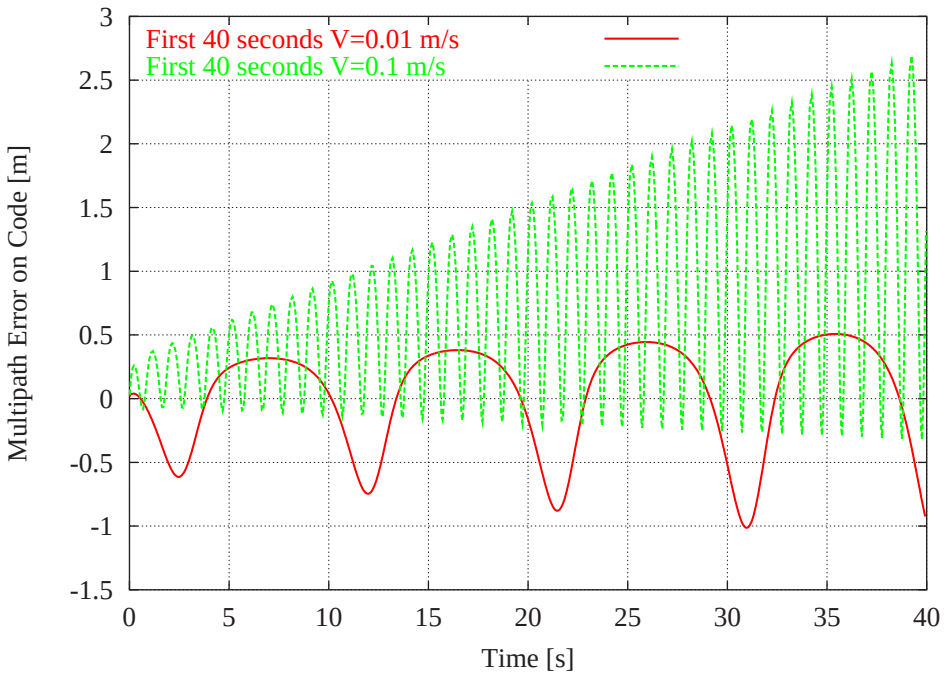

Figure 4.17.: Multipath error as a function of time for velocities 0.01 and 0.1 m/s. Note the sharp peaks in the negative part for the 0.01 m/s case. Note also that the horizontal axis denotes the *time* and not the actual geometric delay. After 10 s the geometric delay is 20 cm in the slow case and 2 m in the faster case.

However, it should be mentioned that the scenario used here provokes the largest change in multipath. If the incident angle is not oblique and the receiver is not moving perpendicular to the reflector, the change in geometric multipath is diminished accordingly. Nevertheless, we will still be talking about the same order of magnitude.

The simulation above was repeated for other chip shapes. The impact of band limitation and raised cosine chip shaping is demonstrated in figure (4.18). In figure (4.19) the fading effects for the BOC(2,2) signal are shown for both code phase and carrier phase. The signal-receiver configuration from table (4.5) was used. The signal in the band-limited case was a rectangular chip with the same parameters as the signal above, except that

Rectangular	Raised Cosine	ratio
45 m	31 m	1.45
18 m	10 m	1.80
14.4 m	6.2 m	2.30

Table 4.6.: Maximum multipath error from the simulations in figure (4.18). The last column shows the ratio between the two.

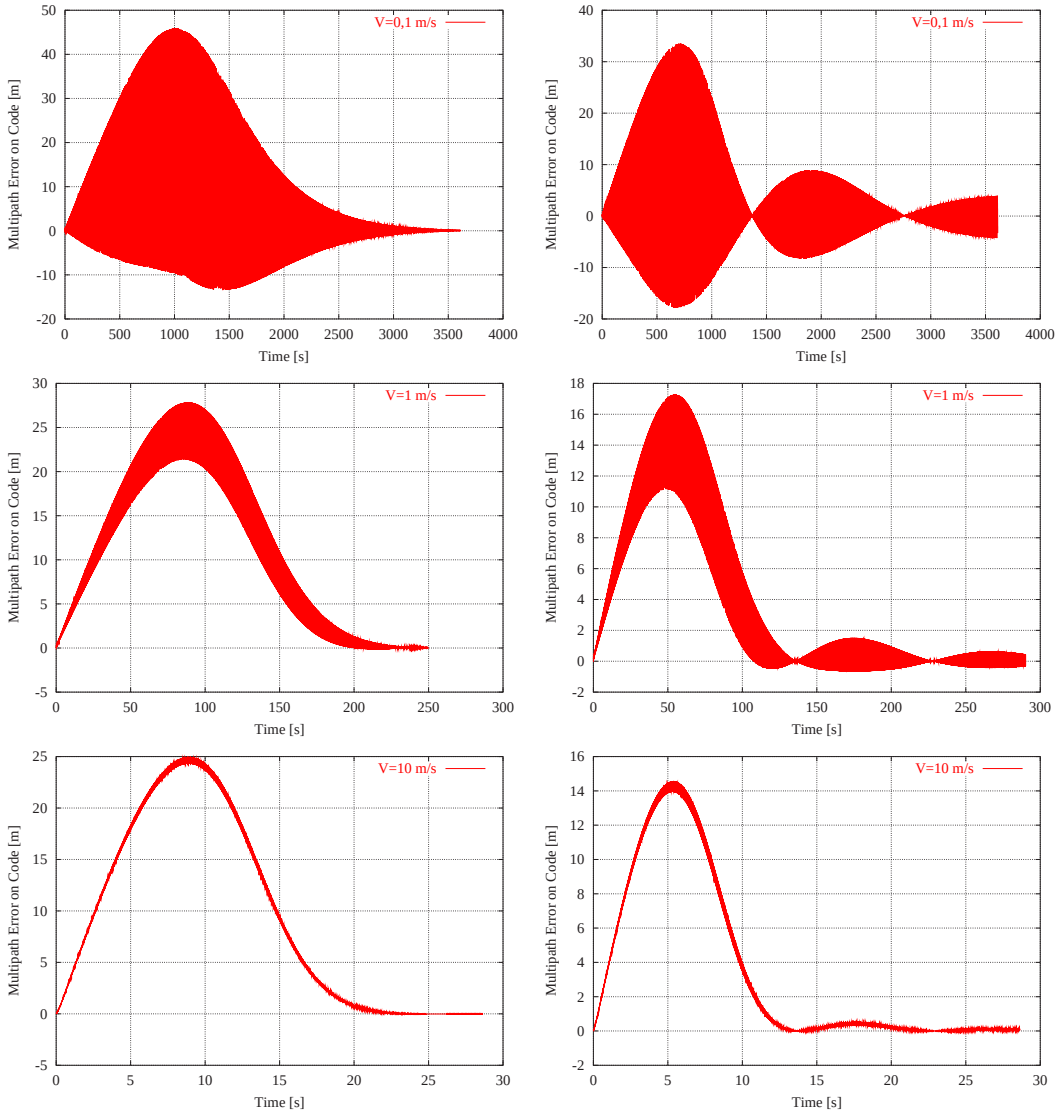

Figure 4.18.: Multipath error as a function of time for the band-limited rectangular and raised cosine chip shape. The geometric setup was identical in all cases except for the velocity. The left plots correspond to the band limited rectangular chip shape (first main lobe only) and the right plots correspond to the raised cosine. Starting from the top, the speed was 0.1 m/s, 1 m/s and 10 m/s for the bottom figure.

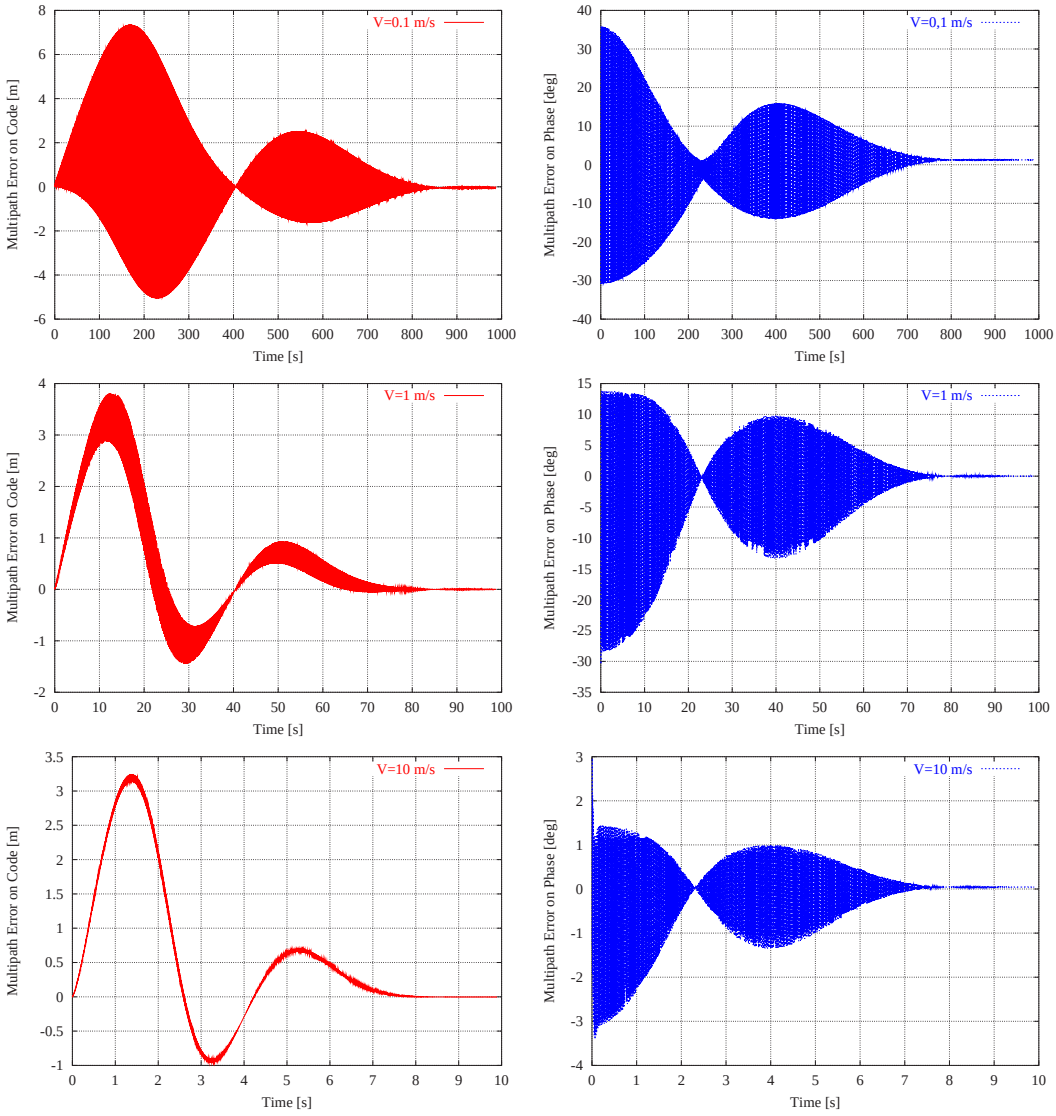

Figure 4.19.: Multipath error for a BOC(2,2) signal. Starting from the top, the velocity was 0.1 m/s, 1 m/s and 10 m/s. The left figures show the error on the code phase and the left figures represent the carrier phase. The geometric setup was identical in all cases except for the velocity.

only the first main lobe was emitted (b = 1 in eq. (2.27)). In order to compare signals occupying similar bandwidths, the chip length for the raised cosine signal was set to $\frac{2}{1+\beta}$ = 183 m. The roll-off parameter was set to β = 0.22. Otherwise the signal is the same as above.

In figure (4.18) the multipath error of the two band limited signals is compared. First we notice that the raised cosine signal shows some side lobes in the multipath error curve. This is caused by the side lobes in the pulse shape. This means that the reflection with a larger delay than $\frac{3T_c}{2}$ will affect the correlation process. Secondly, we see that first zero in the envelope curve occurs earlier than for the band limited rectangular curve. This is due to the higher chipping rate of the raised cosine signal.

The maximum multipath error for the two signals are shown in the table (4.6). As can be seen in table (4.6) and figure (4.18) the multipath error is larger for the rectangular band-limited signal. Further, as the speed is increased (and thus the relative Doppler shift between the direct and reflected signal) the ratio between the two also increases. This suggests that the multipath error is more symmetric for the raised cosine signal than for the rectangular band-limited signal and thus that the multipath fading is more effective in that case.

In figure (4.19) we note the familiar signature of the multiple peaks in the auto-correlation function. The „bubbles" may seem similar for the BOC and the raised cosine signal. However, there are two qualitative differences. The first is that the „bubbles" abruptly vanish when the multipath delay exceeds the 3/2 T_c. Secondly, in contrast to the raised cosine signal the BOC signal structure maintains a negative region for increased fading.

Multipath on the phase measurement In figure (4.20) the phase error due to multipath is shown. The data was taken from the same session as the data above.

As with the code error the first graph approximates the static case very well and the well-known envelopes for the multipath phase error are recognized.

The dynamic behavior of the phase multipath is very dependent on the relative Doppler shift between the direct and the reflected signal both qualitatively and quantitatively. For a slowly changing geometrical multipath, we have a certain fading effect. This fading sets in much later than for the DLL. This is because the loop bandwidth is much larger for the PLL than it is for the DLL.

If we look at the magnitude of the multipath error in figure (4.20) we would expect the PLL to have lost lock a long time ago. In fact, if the boundary conditions for the fourth plot in figure (4.20) are changed slightly, the PLL *does* skip some cycles at the beginning. It should be pointed out that the multipath error enters the loop in a complex way and is *not* comparable to a simple $\theta + N(0, \sigma^2)$ *Ansatz*, which in the linear approximation leads to a loss-of-lock at about 15° or 0.26 radians. The multipath error appears in the PLL equations as a driving force which is proportional to a linear combination of two

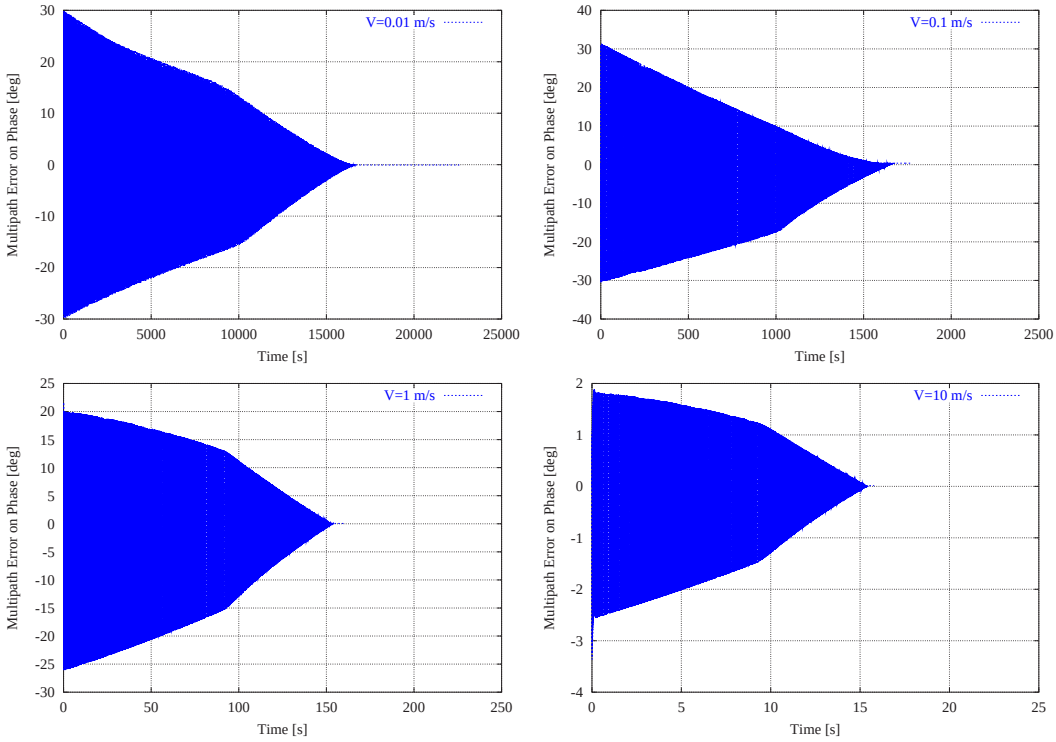

Figure 4.20.: Multipath error on the phase as a function of time. The vertical axis shows the phase error in radians. The geometric setup was identical in all cases except for the velocity. Starting from the top left figure the speed was 0.01 m/s, 0.1 m/s, 1 m/s and 10 m/s for the bottom right figure. The signal structure was the same as in figure (4.16)

harmonic oscillators in a dynamic deterministic way, thus the assumptions leading to the loss-of-lock point at 15° do not necessarily hold in this case.

In figure (4.21) the multipath error on the phase is shown for a rectangular band-limited signal and a raised cosine signal. The signal structures and the experimental set-up are the same as in figure (4.18).

For both signals the PLL looses lock for a velocity of 1 m/s. This shows that the aforementioned classical loss-of-lock condition is not applicable for this particular error source (i.e. multipath). In the low velocity case (v = 0.1 m/s) and in the high velocity case (v = 10 m/s) the PLL maintains lock.

Pre-detection multipath fading As discussed in section 2.5.1 a pre-detection fading of the reflected signal is to be expected. In contrast to the fading caused by the tracking loops, which is essentially a low pass filtering of the *multipath error* itself, the pre-detection multipath fading reduces the *power* of the reflected signal directly.

The plots in figure (4.22) correspond to the same geometric situation as before. Each plot contains the multipath error with and without the pre-detection fading factor. We see

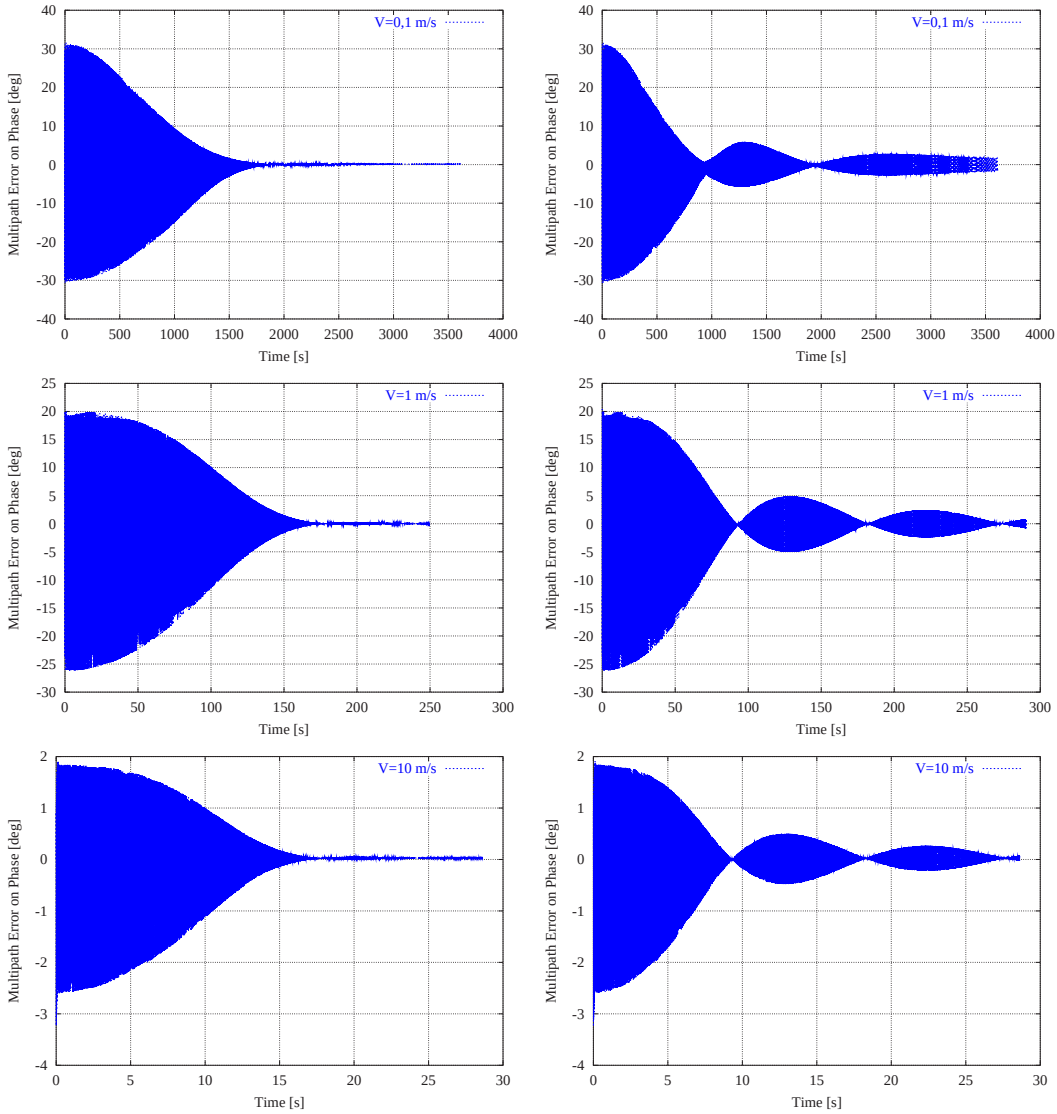

Figure 4.21.: Multipath error on the phase as a function of time for the band-limited rectangular and raised cosine chip shape. The geometric setup was identical in all cases except for the velocity. The left plots correspond to the band limited rectangular chip shape (first main lobe only) and the right plots correspond to the raised cosine. Starting from the top, the speed was 0.1 m/s, 1 m/s and 10 m/s for the bottom figure.

that the pre-detection fading causes further significant fading in the multipath error. The „re-appearance" of the multipath error for the velocity v = 7 m/s is explained by the form of the fading factor ($\mathrm{sinc}(\cdot)$) in eq. (2.134).

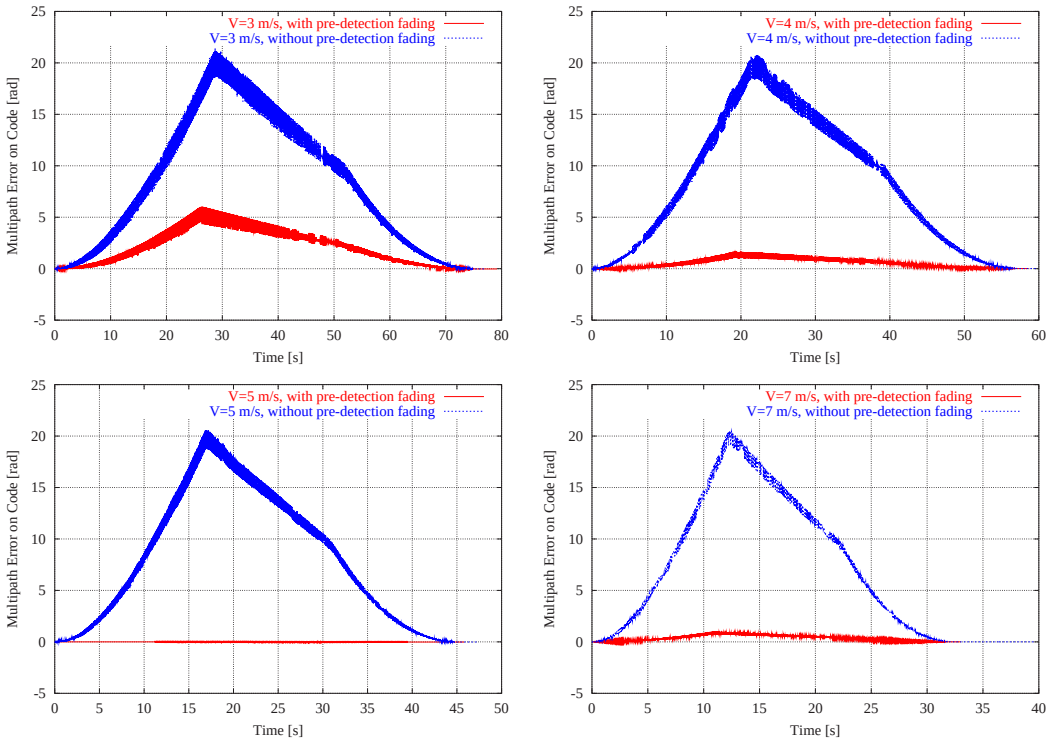

Figure 4.22.: Multipath error as a function of time. Comparing the multipath error with (red curve) and without (blue curve) the pre-detection multipath fading.

As the integration interval of the correlators is usually in the order of 20 ms or less and thus the bandwidth of the correlators (the pre-detection bandwidth) is much larger than the loop filter bandwidth of the DLL, it is clear that the pre-detection fading sets in at a much later point. Therefore one might be led to think that the pre-detection multipath fading is negligible compared to the fading caused in the loops. This would of course be true if the mean value of the multipath error were zero. But as we saw previously this is in general not so, at least not for the non-coherent DLL. The pre-detection fading is of an entirely different nature; it basically generates a fading factor of $\mathrm{sinc}(\Delta\omega T_p/2)$, which for a given relative Doppler shift between the direct and reflected signal can become zero. This is seen in the plot for a vehicle velocity of 5 m/s. This is consistent with the back-of-the-envelop calculation in section 2.5.1. In

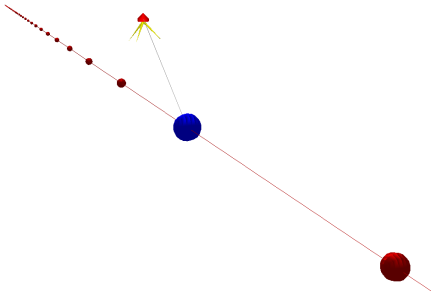

Figure 4.23.: Three dimensional view of the fly-by scenario

any case it generates a substantial fading for velocities above 10 m/s. It should, however, be emphasized once again that the fading effects discussed here depend on the Doppler difference and not directly on the velocity of the vehicle.

4.2.2.1. Transient Tracking Errors

In section 2.5.4 we saw that the transient behavior of the tracking loops is proportional to the time derivative of the line-of-sight. The simple simulation in this section is concerned with this phenomena.

In the chapter on constellation the temporal behavior of the line-of-sight was analyzed. This behavior of the line-of-sight is very important for the dynamic performance of the receiver because the transient properties of the tracking loops is directly proportional to it. In particular this is important for signal sources, which are close to the receiver, such as pseudolites.

In chapter 2.1 a simple geometric setup was analyzed regarding the time derivatives of the line-of-sight. A similar scenario was constructed in the simulator.

In figure (4.23) and figure (4.24) a receiver moving past a pseudolite with a constant speed of 10 m/s is shown. The fly-by distance (i.e. the shortest distance between the receiver and the pseudolite) was 15 m.

The thermal noise was turned off and the tracking loop bandwidths were B_L^{pll} = 20 Hz and B_L^{dll} = 1 Hz. No reflectors were installed and thus the tracking error was solely due to the dynamics of the line-of-sight.

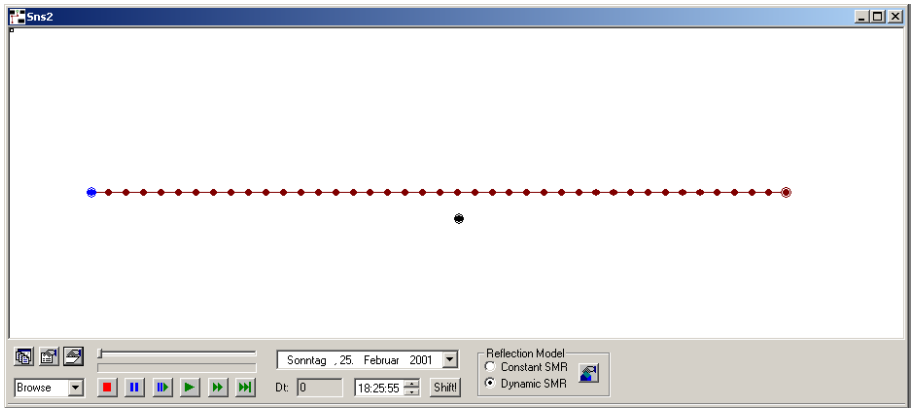

Figure 4.24.: Overview of the simple fly-by scenario. The red dots represent a logging interval of 1s. The receiver is moving with a speed of 10 m/s. The black dot represents the position of the pseudolite. The distance between the red dots is 10 m.

The scenario was run twice. In the first run the carrier aiding of the DLL was turned off and in the second run it was on. The tracking errors of the DLL are shown in figure (4.25) and the behavior of the PLL is shown in figure (4.26).

Comparing the curves in figure (4.25) and the geometric analysis in figure (2.3) in section 2.1.3 we see that the left plot in figure (4.25) does indeed have the same form as the first derivative of the line-of-sight in figure (2.3). Setting B_L^{dll} into eq. (2.182) yields a tracking error of 2.5 m. This is consistent with what is observed in figure (4.25).

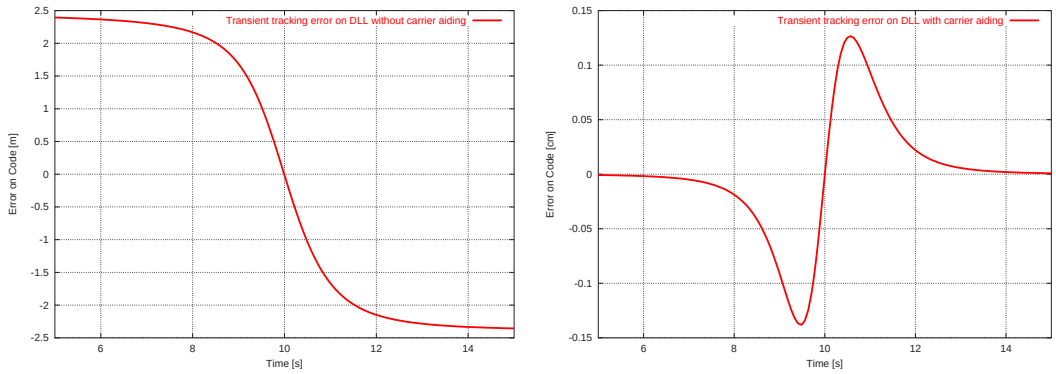

Figure 4.25.: Transient tracking error of the DLL. In the left plot carrier aiding was disabled.

If the fly-by distance was decreased, the speed increased or the loop filter bandwidths decreased, the PLL was observed to loose lock and not to recover immediately. The reason why the tracking loops did not recover and lock onto the signal after the receiver passed the pseudolite was that the violent behavior of the derivatives of the line-of-sight caused the state of the integrator to become filled with such large numbers that no convergence was possible. Thios behavior is surly implementation dependent. If the integrator were re-initialized after a loss-of-lock, recovery was possible. Therefore it is important to implement a resetting of the tracking loops after a loss-of-lock has been detected.

From this we conclude that the transient behavior is very important for signal sources close to the receiver, like pseudolites. In particular for pseudolites at airports it may not be possible to place the pseudolites anywhere. These effects demonstrated here will probably have to be taken into account.

In this simulation the “classical” near-far effect was *not* considered. The simulator adapts ideally to changes in power level and thus things like saturation in the amplifiers and non-linearities etc. were not taken into account. These effects will make the whole situation worse.

4.2.3. Tracking BOC(nx,x) Signals for Large n

Acquiring and maintaining lock of a BOC(nx, x) may become increasingly difficult as n becomes large. A particular example of such a signal would be BOC(14,2) (n = 7). In this section two simulations are made. One where the only error source is thermal noise and

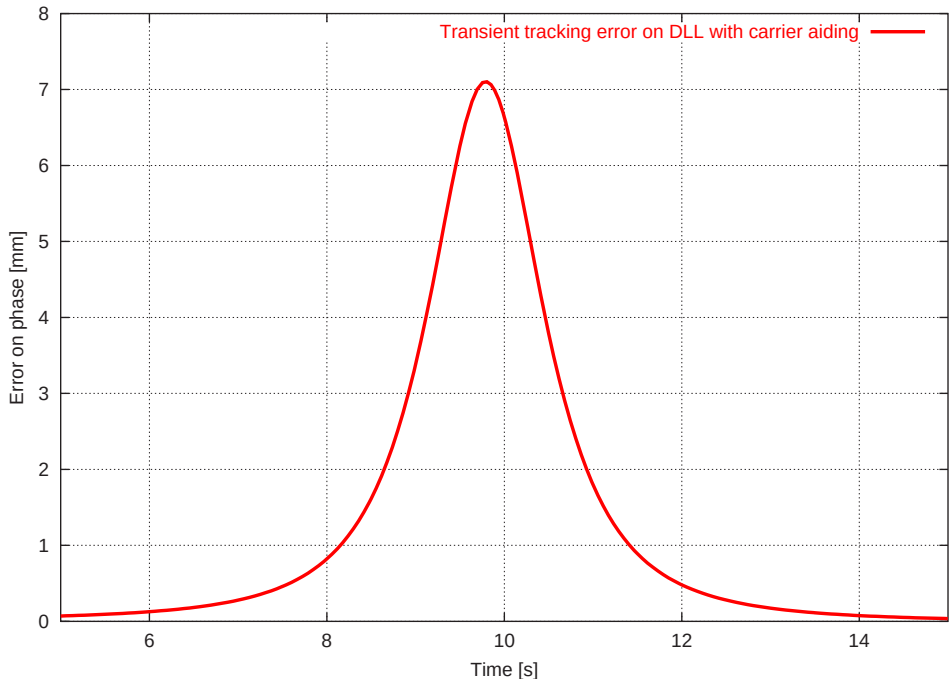

Figure 4.26.: Transient tracking error of the PLL. The transient tracking error of the PLL is independent of carrier aiding.

instabilities of the oscillator. The other simulation is a simple scenario with a single multipath. The loop bandwidths of the DLL and PLL were 1 Hz and 20 Hz respectively. The correlator spacing was 0.05 T_c.

In the first plot of figure (4.28) the DLL error without carrier aiding is shown for a signal-to-noise ratio of 18 dB-Hz. This value of the signal-to-noise ratio was the lowest possible value where the tracking loop was observed to maintain lock on the correct peak of the auto-correlation function for a substantial amount of time (more than 30 min). The phase tracking loop is completely unable to obtain a lock on the signal.

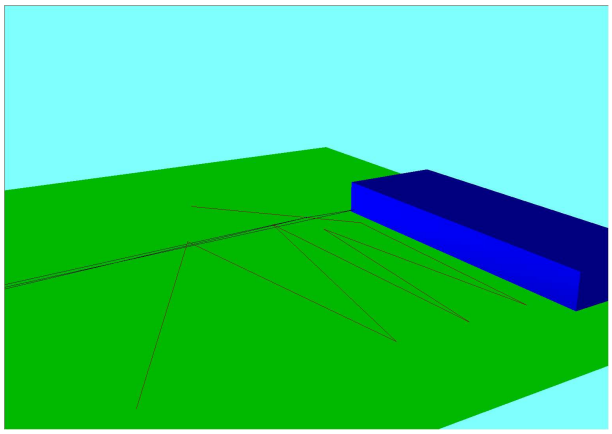

Figure 4.27.: Simple scenario to demonstrate the multipath performance of a BOC(14,2)

For values lower than 18 dB-Hz the DLL missed the auto-correlation function completely. In figure (4.29) the behavior of the DLL is shown for a signal-to-noise ratio of 16 dB-Hz. In the beginning the loop locks onto the correct peak, but looses

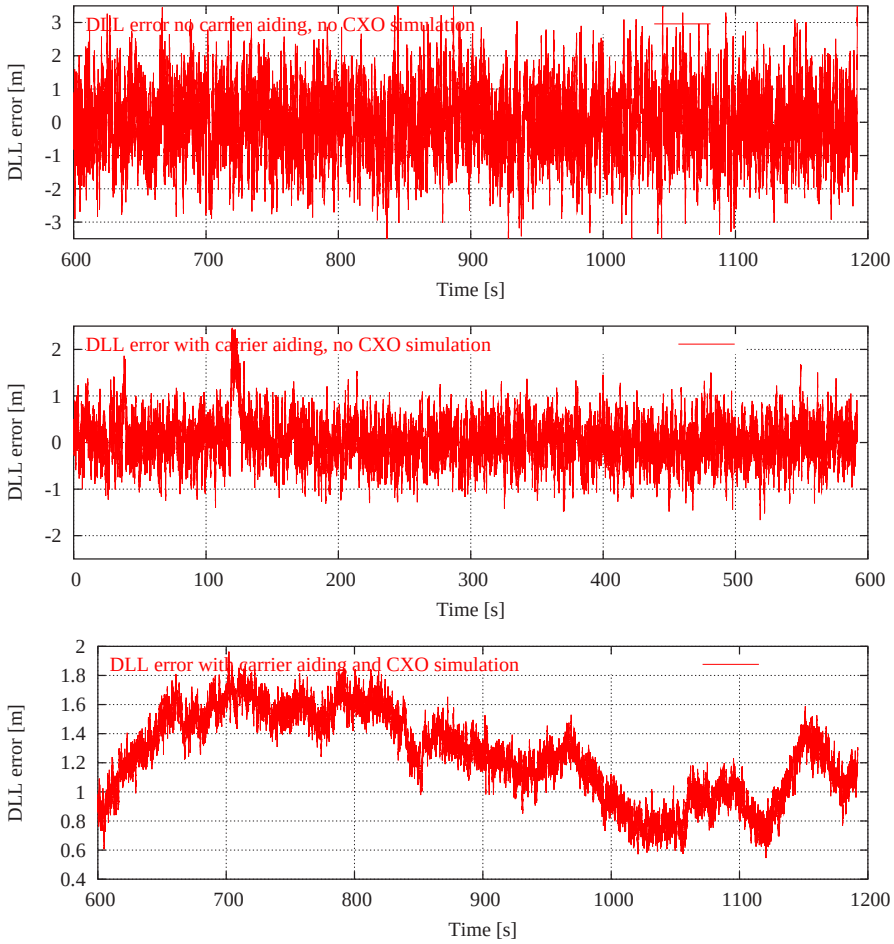

Figure 4.28.: Multipath error on the code for BOC(14,2) under harsh conditions

it after a about 10 seconds. Then there are brief locks onto other maxima of the auto-correlation function as indicated in the figure. The behavior observed in the 100 seconds or so is very similar to that of a PLL that is unable to maintain lock. This analogy is valid until the tracking error is larger than the support of the auto-correlation function.

The phenomenon seen in the last 80 seconds in figure (4.29) is caused by the acquisition algorithm. When the tracking point leaves the auto-correlation function, no signal is detected and the acquisition sets in. This causes the tracking point to be continuously "thrown into" the auto-correlation function again and again, but the tracking loop cannot maintain lock. For the phase-tracking loop the discriminator is periodic and thus there is no acquisition in this sense.

In the second plot of figure (4.28) the DLL error with carrier aiding but no oscillator simulation is shown for a signal-to-noise ratio of 23 dB-Hz, which was the lowest for which the DLL was stable. The DLL maintains lock on the correct peak, but the PLL slips cycles repeatedly as can be seen in figure (4.30).

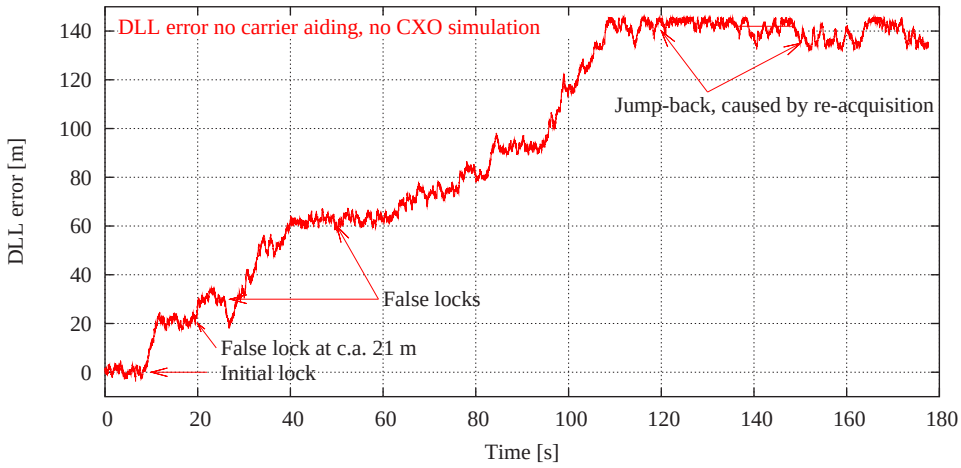

Figure 4.29.: Cycle slipping of BOC(14,2) under harsh conditions

In the third plot of figure (4.28) the DLL error with carrier aiding and oscillator modeling using the standard oscillator from table (2.1) is shown. The lowest tolerable signal-to-noise ratio in this case was 36 dB-Hz. The PLL slips cycles as before, but the behavior is not as violent.

Note the slow oscillation in the code error. This is due to the accumulated noise of the oscillator entering the DLL via the carrier aiding. Here it is seen that the error of the tracking loops, when all things are considered, is a complex stochastic process. In the third plot of figure (4.28) everything comes together: The combination of simulating the oscillator error and feeding this result into the tracking loops, the non-linearities of the tracking loops (PLL and DLL) and the simulation of the thermal noise.

To see if continuous multipath will cause the DLL to loose lock a simple scenario as shown in figure (4.27) was used. The signal and correlator spacings were as before. A rather high signal-to-multipath ratio of 0.5 was used. The signal source is a pseudolite, placed far enough away from the receiver so that the signal dynamics are kept low. The receiver moves with a velocity of 5 m/s along the path shown as red line-segments in the figure. The signal-to-noise ratio varied from 36 to 40 dB-Hz.

The result is shown in figure (4.31). We recognize the section from the multipath envelopes. The dynamics were kept relatively low to provoke high multipath errors. Increased dynamics lead to a better averaging of the errors and thus less danger for the DLL to slip a cycle. The simulation shows that the DLL stays on the right correlation peak.

From these simulation we conclude that neither the code-noise, oscillator noise nor continuous multipath are a serious threat to the DLL tracking of a signal like BOC(14,2). The phase tracking breaks down much earlier. However, using carrier aiding when the PLL slips cycles should be avoided (as always) and very bad reference oscillators can in fact have a bad influence on the code tracking.

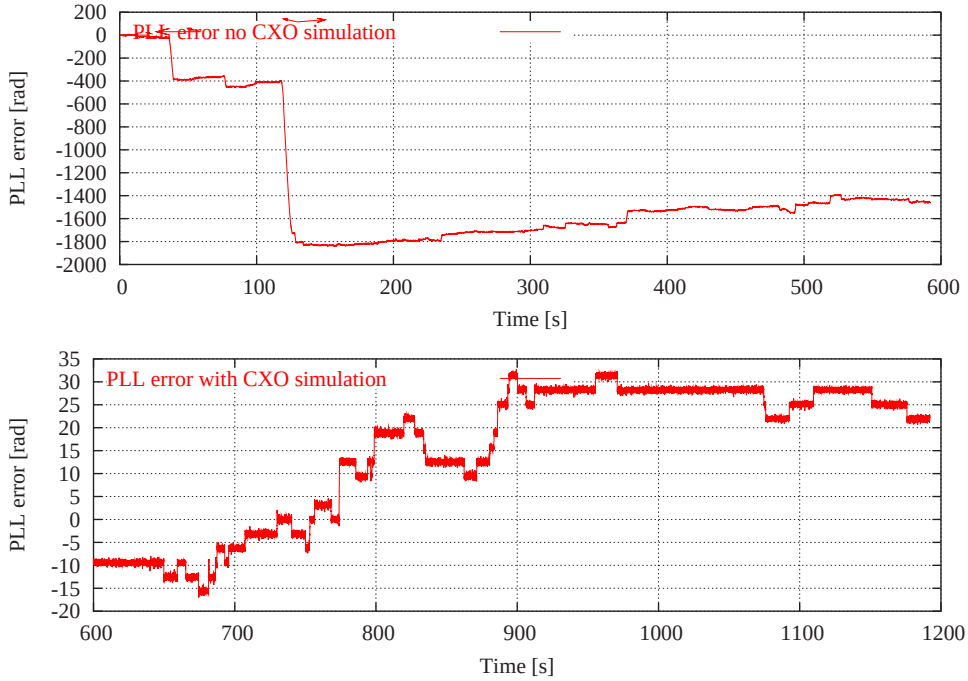

Figure 4.30.: Multipath error on the phase for BOC(14,2) under harsh conditions. The upper plot corresponds to a signal-to-noise ratio of 23 dB-Hz, but without oscillator instabilities and the lower to 36 dB-Hz using the CXO from table 2.1

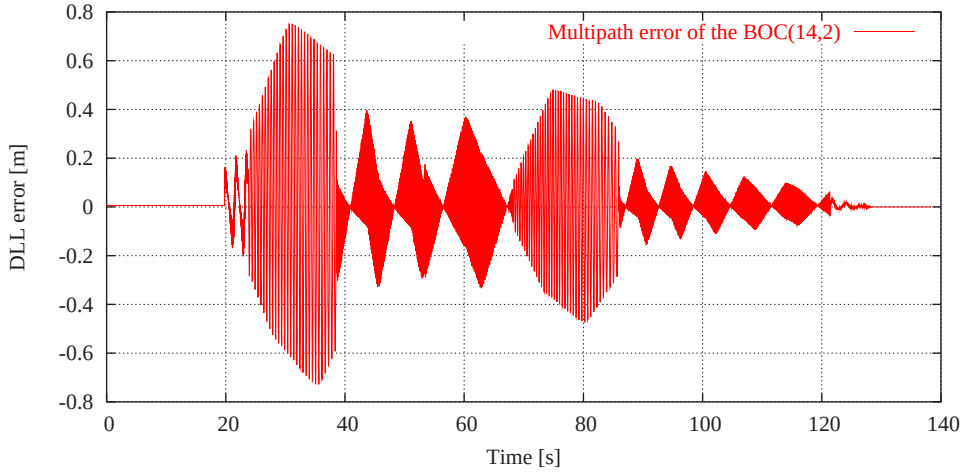

Figure 4.31.: Multipath error on the code for BOC(14,2) under continuous, dynamic multipath conditions.

The issue of wrong acquisition has come up in the discussion of the BOC(14,2) signal. It may therefore come as a surprise that this signal seems, in fact, to be very robust concerning falling out of step. It can be made plausible when the analogy to carrier tracking is considered. When the BOC(14,2) code is being tracked as is done here, it is comparable to tracking a carrier phase with a wave-length of approximately 10 m (see figure (4.41), where the discriminator curve for this example is shown), compared to 10 cm (Costas-loop) when tracking the carrier phase in L-band. Therefore one would expect a very stable lock onto the tracking point. As we shall see in the next section there are still issues with this signal in a more complex environment.

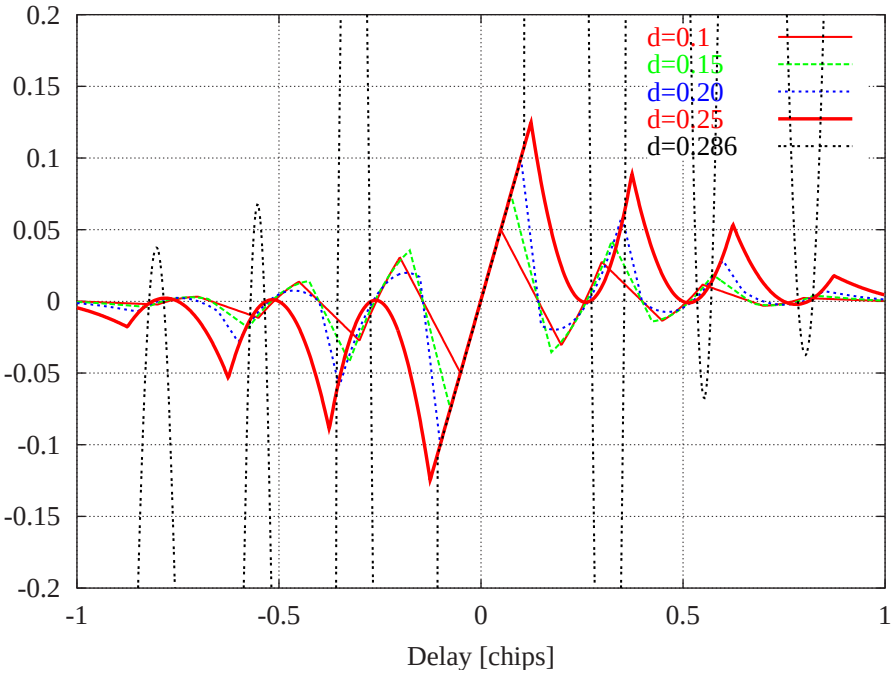

Figure 4.32.: Discriminator for a BOC(2x,x) signal for various correlator spacings

4.2.4. Correlator Spacing for BOC Signals

The choice of correlator spacing for BOC signals seems to be quite critical. If the correlator spacing is increased, the slope of the discriminator will eventually change sign [Bet00b]. The form of the side-lobes is also very dependent on the correlator spacing. Here we examine the impact of correlator spacings for BOC signals.

In figure (4.32) the normalized discriminator of a BOC(2x,x) signal in infinite bandwidth is shown for various correlator spacings. The S-curve is normalized such that the slope $S'(0) = 0$. Therefore the S-curve seemingly behaves normally around zero, but the side-lobes diverge when the slope becomes small.

There are two important points to notice in figure (4.32). The first is that as the correlator spacing d is increased to $0.285T_c$, the slope of the discriminator eventually becomes zero (divergence of the side-lobes). A further increase (not shown in the figure) will reverse the sign.

The second important point about figure (4.32) is the form of the side-lobes. Generally, there are multiple zeros with a positive slope, which lead to a potential danger that the tracking loop will lock onto those zeros. However, for a correlator spacing of $d = 0.25$ roots of the discriminator become second order, i.e. the slopes at the roots are also zero. Although the slope at zero for the $d = 0.25$ is quite low, it is important that there are no stable tracking points, except for zero delay. Thus, false locks during tracking should be less likely or even impossible. This is particularly important for BOC(nx, x) for large n, like the BOC(14,2).

In figure (4.41) the S-curves for the infinite bandwidth BOC(14,2) signal for various correlator spacings are shown. As in the BOC(2x,x) case the zeros of the S-curve for $d = T_c/2n = 0.071428T_c$ seem to be of second order. The same tendency is seen in figure (4.41) as in figure (4.32) that as the correlator spacing goes beyond $d = T_c/2n$ the slope of the S-curve at zero decreases and consequently, the side-lobes diverge.

In figure (4.32) and figure (4.41) the auto-correlation function in infinite bandwidth was used. As was already pointed out in section 2.2.8, the zeros of the auto-correlation function shift slightly as the bandwidth is decreased. Therefore it is not obvious that an adjustment of the correlator spacing may lead to similar results in the case of finite bandwidth. To analyze this the S-curve of a BOC(14,2) signal limited to a bandwidth of 32.736 MHz is shown in figure (4.33) for various values of the correlator spacing.

To further examine the multipath behavior for the particular value of $d = T_c/2n$, the same simple set-up was used as before (see figure (4.11)). The velocity was 1 m/s and to account for the dynamic averaging of the tracking loops the loop filter bandwidth was set to 1, 5 and 100 Hz. No noise was assumed. The goal was to see whether the interesting correlator spacing of $d = T_c/2n$ could be used and still have an acceptable multipath performance.

In figure (4.34) the results for d=$0.071428T_c$ and d=$0.05T_c$ are compared. For the high loop filter bandwidth case there is no significant difference. For the smaller loop bandwidths the multipath error oscillates in the positive region for the upper figure, while it is more symmetric in the lower figure. From this it can be concluded that concerning the multipath performance there is nothing against using a correlator spacing that reduces the zeros of the side-lobes of the S-curve to second order roots.

The last issue we want to clear here is the thermal noise. To analyze that a receiver was set up next to a pseudolite in the same way as described in section 4.1.1. The signal-to-noise ratio was set to 29 dBHz and a loop filter bandwidth of 1 Hz was used. The DLL error was recorded for 100 s and the one sigma values evaluated.

The result is shown in figure (4.35). The singularity around $0.075T_c$ where the slope of the S-curve is approaching zero is clearly visible in the data. The point of interest is

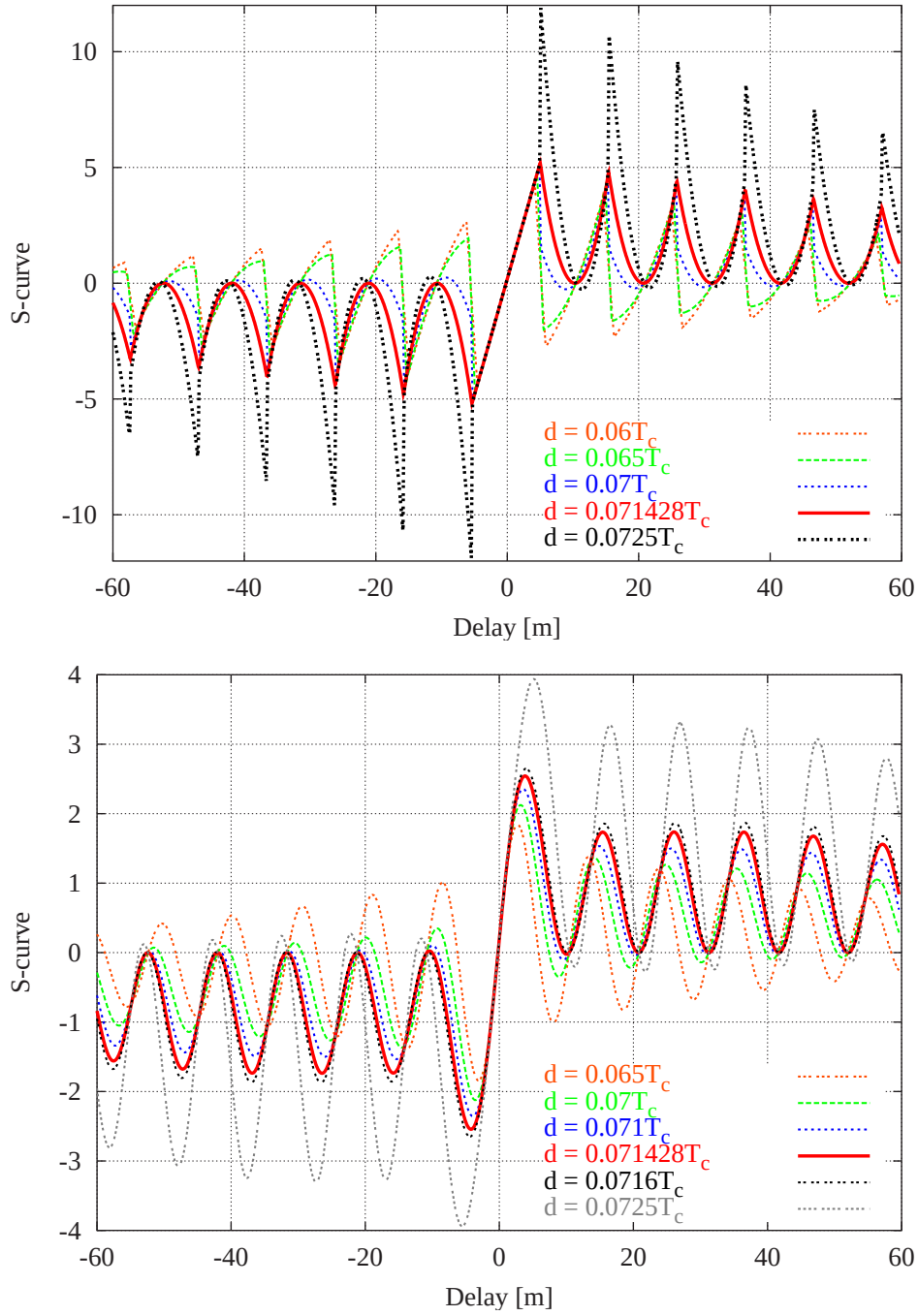

Figure 4.33.: Discriminator for a BOC(14,2) signal for various correlator spacings. The upper figure refers to infinite bandwidth and in the lower figure a bandwidth of 32.736 MHz was used

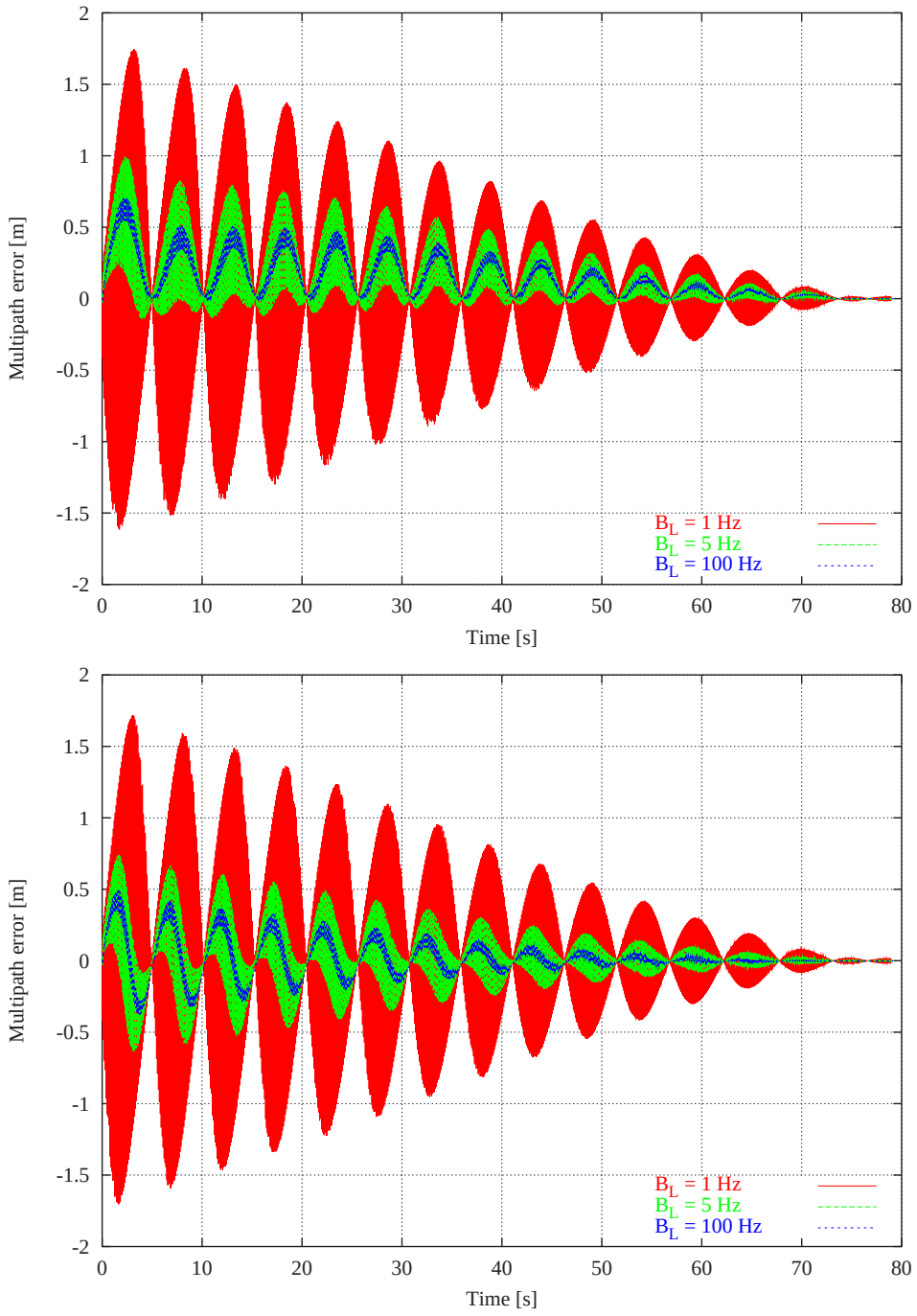

Figure 4.34.: Multipath error for a BOC(14,2) signal bandwidth-limited to 32.736 MHz for three different loop bandwidths ($B_L = 1$, 5 and 100 Hz). In the upper figure the correlator spacing was $d = T_c/2n = 0.071428T_c$ and $d = 0.05T_c$ for the lower one.

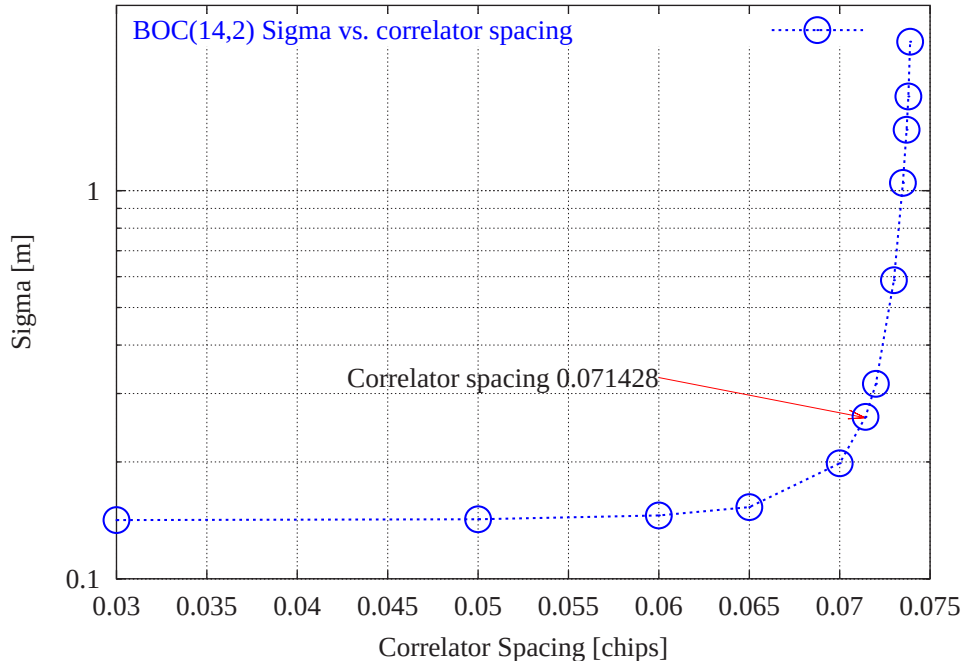

Figure 4.35.: Thermal noise as a function of correlator spacing for a bandwidth limited BOC(14,2) signal.

quite close to the singularity, but not quite in the critical region. With respect to the thermal noise performance one would probably chose a correlator spacing in the flat region, i.e. somewhere between 0.03 and 0.06. This would generate a one sigma level of round 0.14 m. However, if the correlator spacing is chosen as $d = T_c/2n = 0.071428T_c$ then a penalty of a factor 2 must be accepted, i.e. here the one sigma level is 0.26 m.

So, all-in-all one can say that it is possible to choose a correlator spacing such that false locks in the auto-correlation function could potentially be avoided. There is no significant penalty in multipath performance, but the thermal noise performance suffers by a factor of approximately 2. It was also shown that the bandwidth limiting of the signal did not destroy this property of the S-curve.

Although the special feature of the S-curve, which can be achieved with the corresponding choice of correlator spacing may help to avoid false locks in the signal tracking, special arrangements will be necessary for the signal acquisition. The problem is that the acquisition procedure *will* detect signal power outside the pull-in region. In that case if the receiver switches to tracking, it will soon loose lock because of the instability in the S-curve. Even if the danger of false locks could be diminished a special acquisition procedure will still be necessary.

4.3. Complex Scenarios

After having looked at smaller simple scenarios we now turn to scenarios where the interaction of the signal with the environment plays a significant role. For the local environment we have two 3-D city models at our disposal: A 3-D city model of Oedekoven, which is a small suburb near Bonn in Germany. This model was obtained by feature extraction by the Institute for Photogrammetry of the University Bonn. The data set contains terrain data as well as buildings. The other model is also a 3-D city model of Stuttgart and was provided by the German Telecom. The models used for the vehicles were downloaded from the Internet.

4.3.1. Moderate Density (Suburb)

In figure (4.36) an overview of the simulation set-up is shown. The terrain is color-coded, ranging from green for the lowest to brown for the highest point. The buildings are shown in blue. The reference path of the vehicle is shown as a red line.

The material properties of the buildings were set to those of concrete, $\varepsilon = 3$ and $\sigma = 1{\cdot}10^{-4}$℧/m.

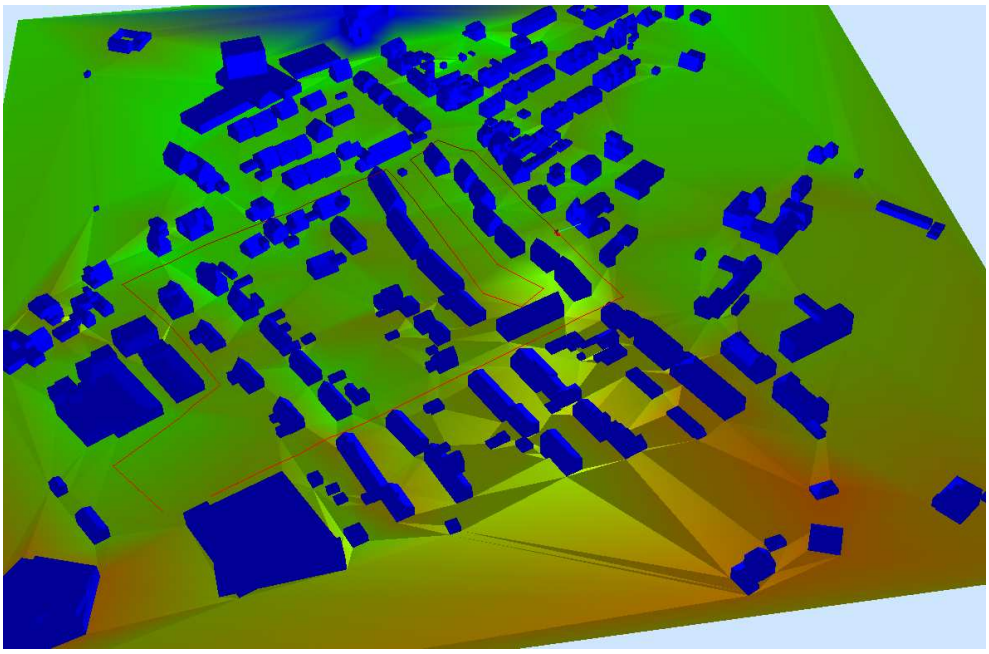

Figure 4.36.: Overview of Oedekoven, near Bonn

The antenna was mounted on the roof of a vehicle model (Ford Explorer). A screen-shot from the SNSS simulation session in figure (4.37) shows a close-up of the initial position.

The antenna of the current position is shown blue. The lines entering the antenna represent the geometric propagation path of the signals.

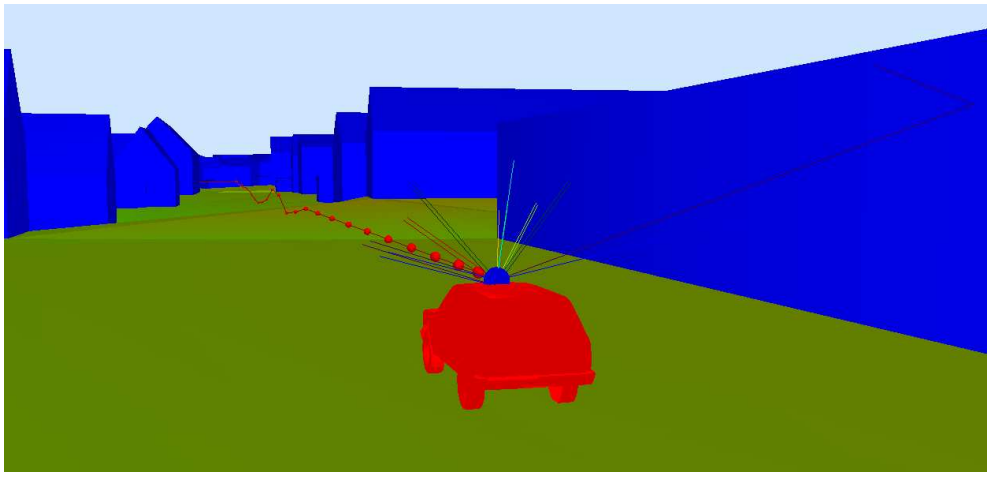

Figure 4.37.: Close-up of the Ford Explorer in Oedekoven.

The satellite constellation used is a GPS almanac from 8. of April 2002, obtained from the Internet (Yuma-format). Three signals were configured on L1 (carrier frequency of 1.575 GHz). A rectangular, infinite bandwidth signal with chip-length of 293 m, RECT(1). This signal represents the GPS C/A code. The second signal is rectangular with a chip-length of 29.3 m (RECT(10)) and bandwidth restricted to the first main lobe, i.e. 20.46 MHz. This corresponds to the Galileo signals on E5a and E5b and the GPS P-code. The third signal is a BOC(14,2) signal, with a bandwidth of 32.736 MHz and is a signal option discussed for Galileo.

The receiver uses the non-coherent early minus late delay lock loop (DLL) for code-tracking and a Costas phase locked loop (PLL) for phase tracking in both cases. The correlator spacing was set to 1 chip length for the C/A and P-code, but the correlator spacing for the BOC(14,2) signal was set to $0.05T_c = 7.3$ m.

Two runs were performed. One with and one without noise modeling. This way a comparison between the error caused by multipath and the error caused by thermal noise is obtained.

The power-level of the satellites was adjusted such that at an elevation angle of 10° the received power level corresponds to the minimum received power-level specified in the GPS ICD [ARI93].

Code Tracking In figure (4.38) the error on the range for the channels tracking the RECT(1) signal on satellite PRN 3 and 21 are shown. The green (dotted) curves show the error caused by multipath only and the red curves is the error caused by thermal noise and multipath. As can be seen the multipath error for this signal is not the dominating

factor in the error budget in this environment. The signal-to-noise ratios achieved were in the order of 46 dB-Hz. Thus, compared to the raw thermal noise on the RECT(1) range measurements the multipath error may not be as critical as usually assumed. However, the case were the receiver locks onto a signal which is *only* received trough a reflection (i.e. no direct line of sight), the resulting error will become substantial.

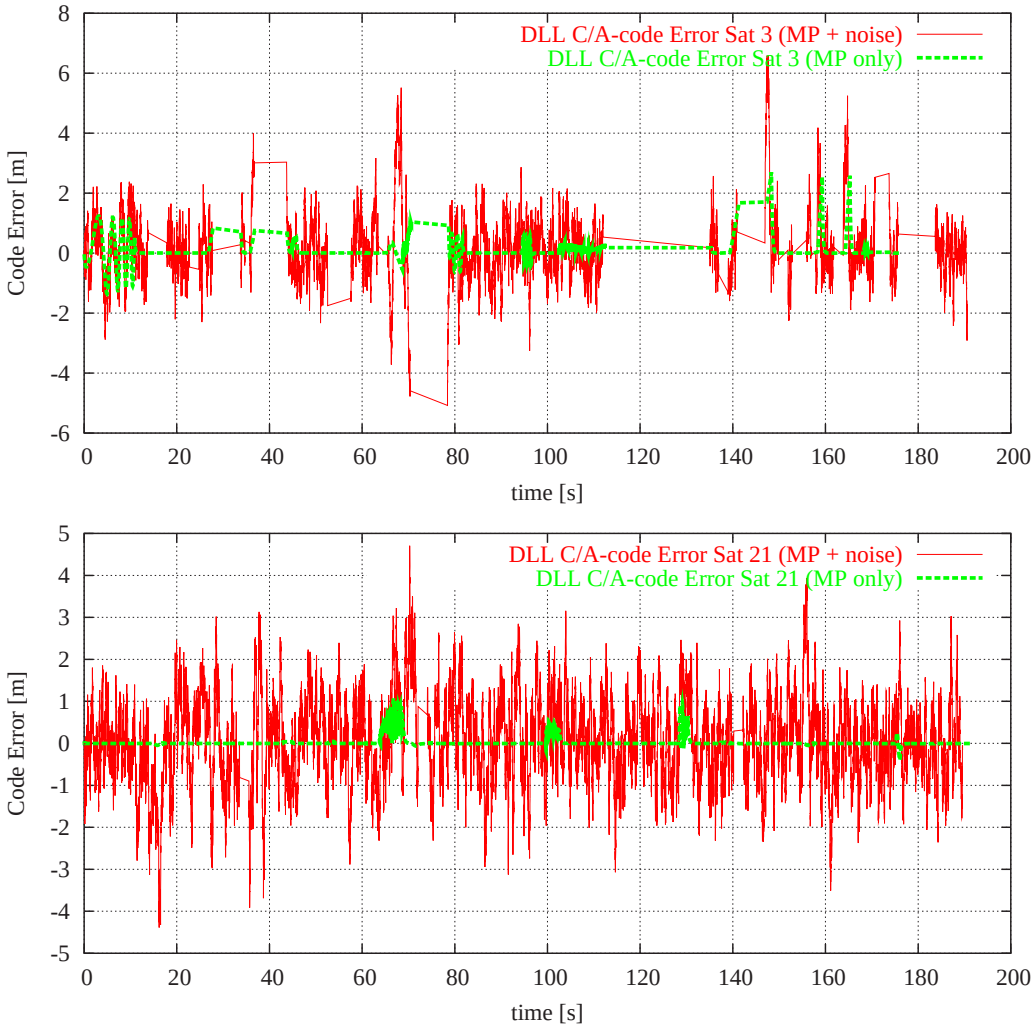

Figure 4.38.: Range error on the RECT(1) signal for satellites PRN 3 and PRN 21.

In figure (4.39) the same situation is shown for the RECT(10) signal. Here the situation is entirely different. The multipath error is an order of degree larger in magnitude than the thermal noise.

As indicated in figure (4.39) for PRN 3 the receiver locks onto an indirect signal, because the direct line-of-sight is blocked. The resulting error is out of the range of the plot; the error for those brief periods of time are 23 m and 32 m. This causes serious damage to the position solution.

The entire geometric situation is identical for the RECT(1) and RECT(10) simulation. Therefore one might expect to see the same peaks in figure (4.38). The absence of the peaks for the RECT(1) signal is no error and has to do with the acquisition process as implemented in SNSS . The pull-in region of the RECT(1) signal is about 300 m. Thus, if the direct line-of-sight is blocked and the resulting indirect signal differs in range by less than the pull-in region, no loss-of-lock is declared and the DLL loop starts to converge to the indirect signal. For the RECT(10) signal on the other hand the DLL looses the signal completely and starts re-acquisition until the signal is found again, only this time it is the indirect signal. Therefore the DLL tracking the RECT(10) signal gains a "head start" on the DLL tracking the RECT(10) signal. Because the duration of the signal blockage is so short, neither of the tracking loops manage to converge properly onto the indirect signal.

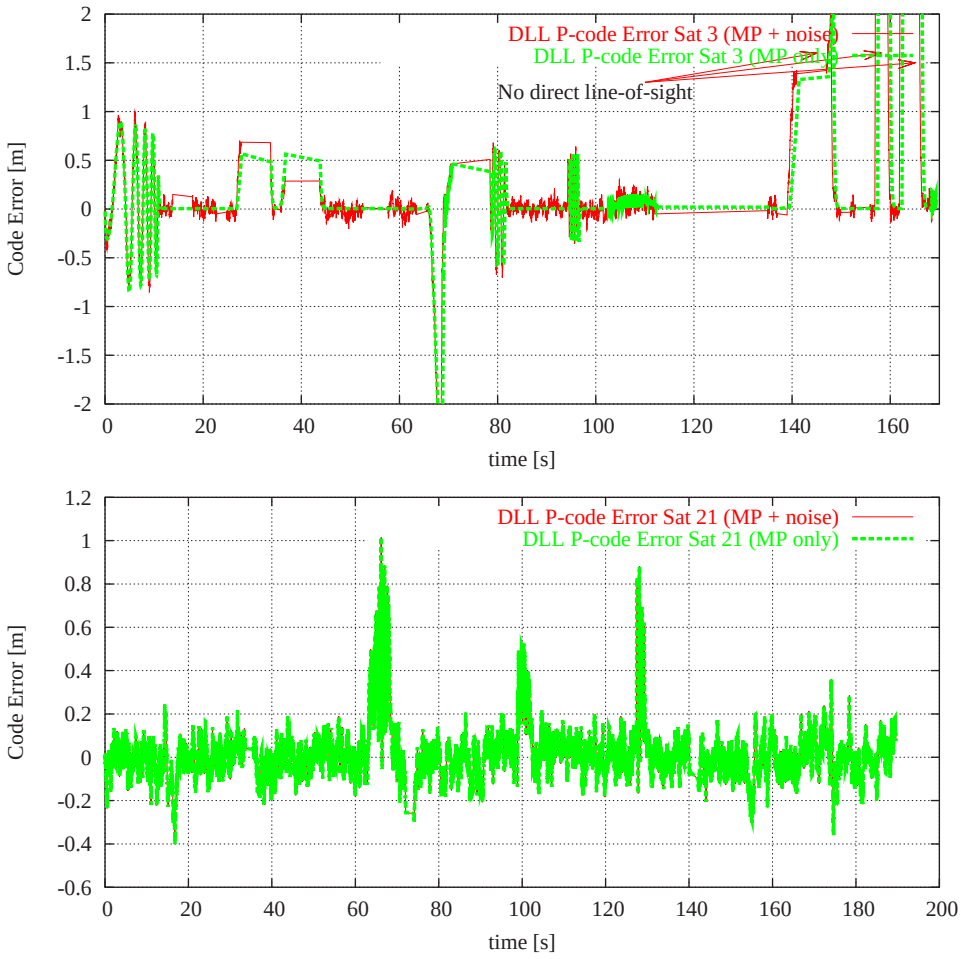

Figure 4.39.: Range error on the RECT(10) signal for satellites PRN 3 and PRN 21.

The noise performance of the BOC(14,2) signal is, as to be expected, extremely good. But the multiple peaks in the auto-correlation function lead to problems concerning the acquisition. As was demonstrated in the section on simple scenarios the multiple peaks of

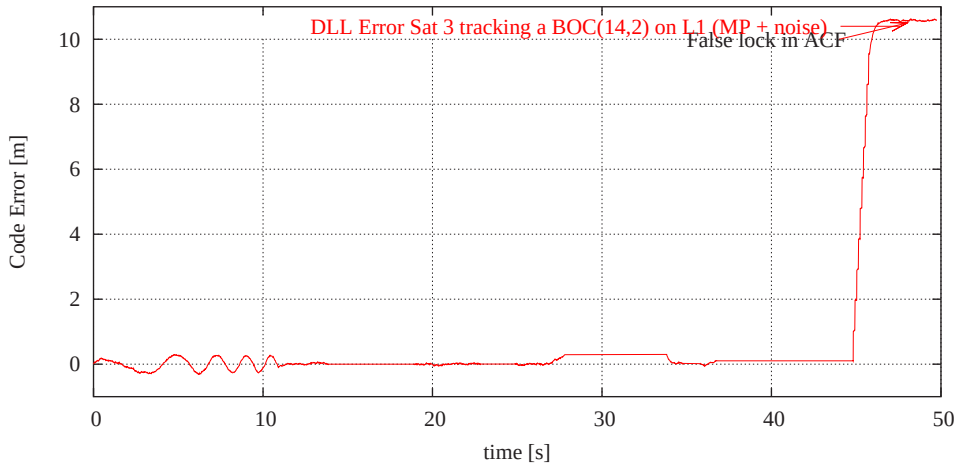

Figure 4.40.: Range error on the BOC(14,2) on L1 for satellites PRN 3

the ACF did not cause serious problems. In this more realistic scenario false locks of the receiver were observed. In figure (4.40) this is seen at around the 45^{th} second. This kind of false lock does not or rarely happens when the change in multipath delay is smooth and the tracking loop has time to follow it. However, in this scenario the false lock is explained as follows: First the direct line of sight is blocked. As a consequence the DLL tries to lock onto the indirect signal. Suddenly, the direct line of sight is visible again and this is where the false lock happens: the tracking loop converges to the next available zero of the discriminator. We notice that the DLL converges to an error of approximately 10.4 m.

In figure (4.41) the discriminator curve of the BOC(14,2) signal is shown. The point of false lock is indicated and it is seen that it is consistent with the 10.4 m tracking offset observed in figure (4.40).

It should be mentioned that the acquisition algorithm used in SNSS basically corresponds to the kind of re-acquisition used in normal receivers. If lock is lost, then the acquisition algorithm starts to "jump" to the left and right of the old tracking point with ever increasing steps, looking for the signal. If a certain signal level is detected, tracking starts again. For a reliable re-acquisition of the auto-correlation function of a BOC(14,2) signal, something more elaborate is called for. This could be in form of some scheme involving a large number of correlators. Another alternative could be to track the side-band of the BOC(14,2) in parallel and continuously compare this to the tracking of the auto-correlation function. If a discrepancy is detected re-acquisition can be initiated. These are interesting developments, however, they will not be pursued further in this thesis. It suffices to have shown that such a scheme is necessary and suggestions have been made as to how this could be done.

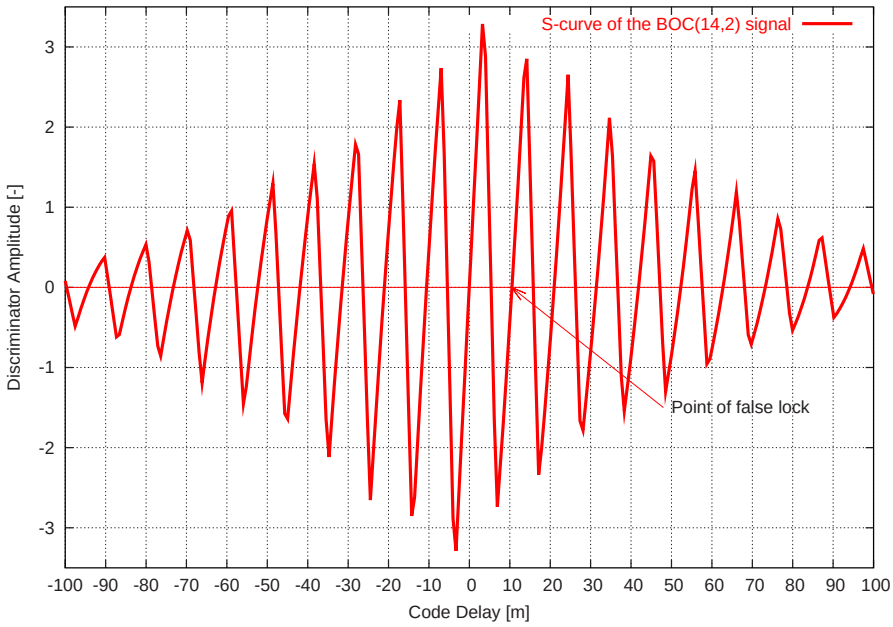

Figure 4.41.: Discriminator curve of the BOC(14,2) signal. Correlator spacing $0.05T_c$ = 7.3 m

Phase Tracking It can be argued that a modern GPS receiver will use the phase measurement to smooth the code measurements and thus improve the code phase error considerably. But for that to be possible the phase measurements must be available and in particular the phase-locked loop should not slip cycles, at least not very often.

In figure (4.42) the result of the phase tracking simulation is shown. During the brief simulation period of 180 seconds at least 6 cycle slips occurred.

A Hatch-filter (typically used for carrier smoothing) usually has a filter constant in the order of 100 seconds or more and therefore a cycle-slip rate as seen in this simulation can be a problem. However, it is possible to detect and repair cycle slips, but it is a rather fragile process and the reliability when they occur as often as in this simulation may suffer.

The lessons learned from those simulations can be summarized as follows

- In a moderate urban area the multipath error is dominated by short range delays. Therefore in such an environment the short-range multipath behavior is very important. Interestingly, there is no significant difference in the multipath performance between the RECT(1) and the RECT(10) signal in this environment.
- For a signal like the RECT(1) signal the multipath performance is much less relevant than the code-noise (see figure (4.38)). However, using a narrower correlator spacing will reduce the code noise.

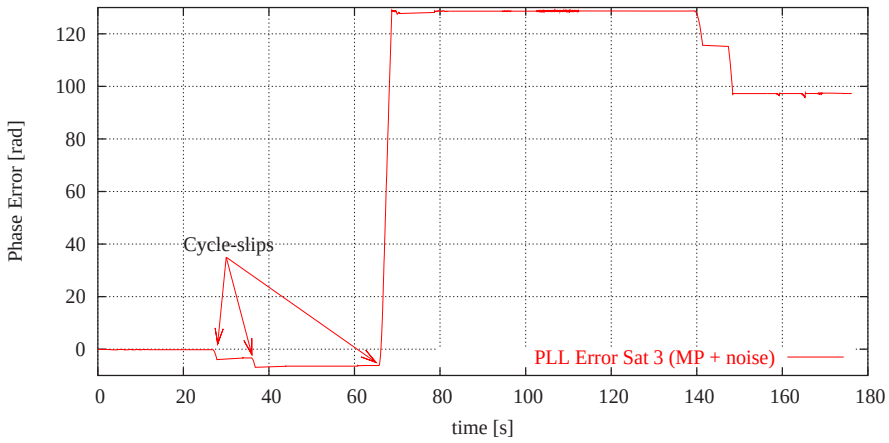

Figure 4.42.: Phase error of the RECT(1) (or RECT(10)) signal for satellite PRN 3. Note that in this environment the phase observation is almost useless due to cycle slipping.

- In a moderate urban area a high rate of cycle slips are to be expected (see figure (4.42)). Thus, using carrier-phase measurements for code-smoothing may not be very reliable.

4.3.2. High Density (Urban Canyon)

For completion a simulation was performed using a model of down-town Stuttgart. The 3-D city model of Stuttgart is shown in figure (4.44). The buildings are blue an the terrain is color coded between green and brown. As in the Oedekoven simulation the signal used was a RECT(1) signal and a BOC(14,2). The route chosen for the simulation is shown in figure (4.43).

The two critical issues addressed in this simulation is the availability of measurements, i.e. the number of satellites directly visible and the tracking of the BOC(14,2) signal.

To assess the availability[2] issue the simulation was performed for a GPS only constellation and a combined GPS and a hypothetical Galileo constellation, consisting of a Walker constellation with the parameters 30/3/1 and an orbit radius of 30,000 km, which roughly represents a Galileo constellation. The GPS constellation used was the GPS almanac from 8. of April 2001.

In figure (4.45) the environment from the users perspective is shown. Comparing this view to the one from Oedekoven (figure (4.37)) the more limited view of the sky is evident. This is also reflected in the decreased availability. For the Stuttgart simulation using the GPS constellation a positioning was not possible for 41 instances of 243, yielding an

[2]The term “availability” as used here simply refers to how often at least 4 lines-of-sight are actually seen by the receiver. It does not necessarily have anything to do with availability in the context of integrity.

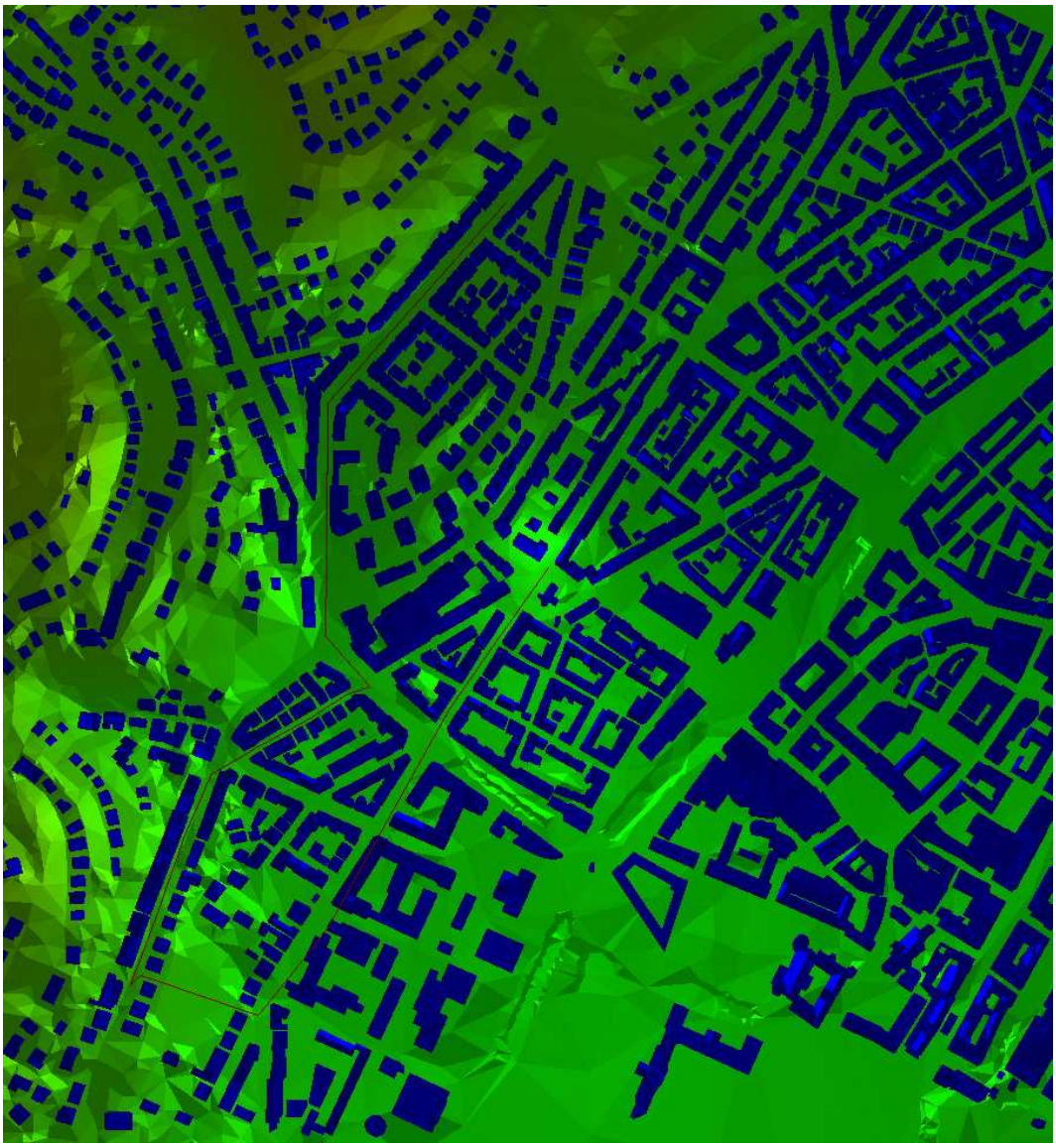

Figure 4.43.: Aerial view of Stuttgart as seen within SNSS. The route chosen for the simulation is indicated by the thin red line.

availability of only 83%. For the combined constellation the availability was 99% (3 out of 243 instances).

Constellation and signal	Position error (RMS)	Availability
GPS/Galileo constellation C/A code	5.1 m	99%
GPS constellation BOC(14,2)	28.6 m	83%
GPS constellation C/A code	5.7 m	83%

Table 4.7.: Total positioning error in the Stuttgart simulation

The root-mean-square (RMS) of the total position error is shown in 4.7. The false locks of the BOC(14,2) signal seriously degrade the performance. The increased number of satellites in the combined constellation improve the position error only marginally.

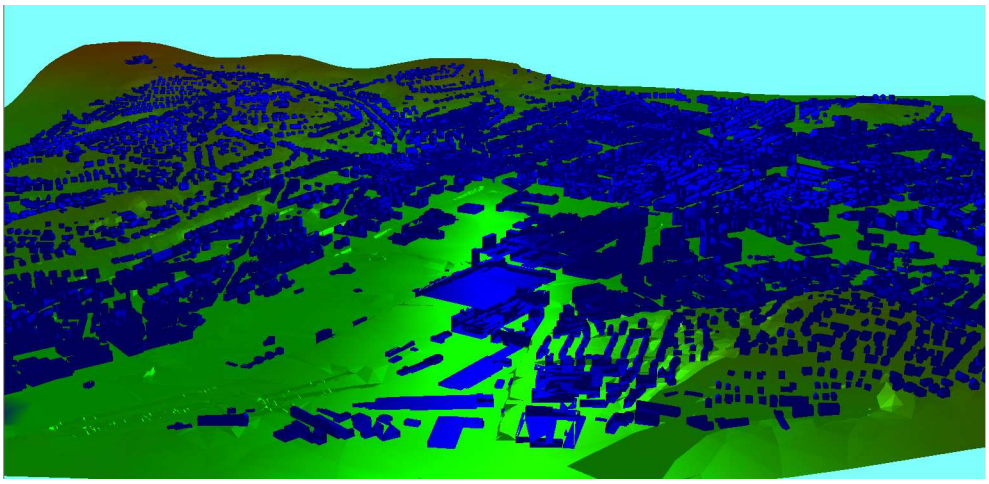

Figure 4.44.: Overview of Stuttgart

It should be noted that these figures refer to the particular simulation performed here and should not be interpreted as any kind of statistical average over all high density urban areas. However, these results imply that the most important issue in an urban environment is the number of satellites in orbit. Thus, the main advantage for car navigation or in general for applications, which operate under harsh conditions, is not necessarily the high quality of the emitted signal, but rather the high availability a *combined* usage of GPS *and* Galileo provide.

In figure (4.46) the total positioning error as a function of time is shown. The total positioning error is the length of the vector between the true reference position and the position generated by the least square calculation. In the figure the error for the GPS only constellation using the RECT(1) and a BOC(14,2) signal is shown as well as the combined GPS/Galileo result.

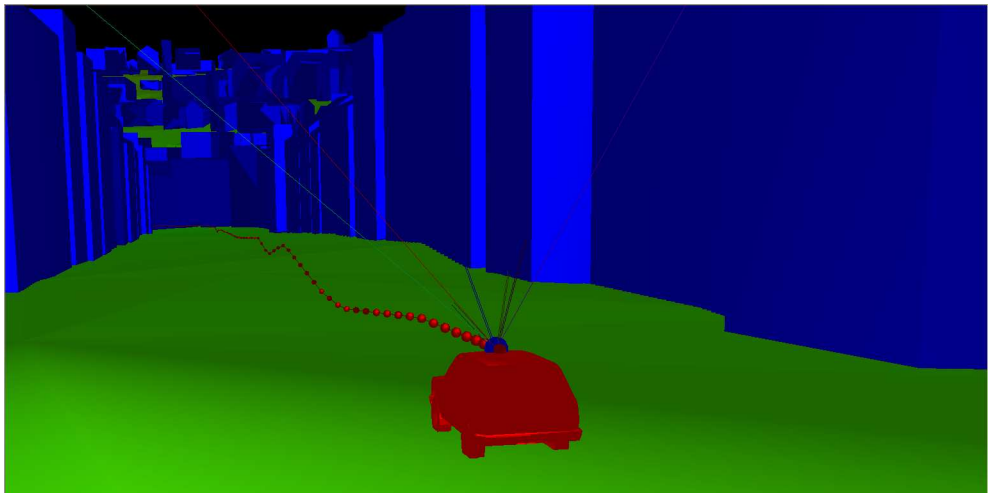

Figure 4.45.: Close-up of the vehicle at the starting point for the simulation in Stuttgart

The total position error for the BOC(14,2) signal shows extreme deviations. This is due to false locks in the auto-correlation function. This can be seen in figure (4.47), where the DLL tracking errors for the RECT(1) and the BOC(14,2) signal are shown for several satellites.

False locks in the BOC(14,2) signal were also observed in the Oedekoven simulation, but they were rather rare. Here in the more dense urban environment these false locks appear more often.

The results here show the BOC(14,2) signal in a very bad light and therefore it is important to interpret the results carefully. The BOC(14,2) signal was tracked using a straight forward, three correlator, non-coherent DLL. This is basically an un-modified GPS receiver architecture with modifications only accounting for the narrower correlator spacing and the higher bandwidth. While it was shown in the simpler scenarios that the lock of the DLL on the BOC(14,2) signal was in fact quite stable, given *continuous* boundary conditions, the more realistic dynamic scenario generated by a complex environment such as an urban area obviously call for a more elaborate receiver implementation.

4.4. Topographic Availability Analysis

In this section the analysis is focused more on the geometry of the topography and the signal propagation and less on the receiver itself. The material in the following section was produced under a EC contract called PARAMOUNT using a slightly modified version of the SNSS simulator.

The analysis is focused on the availability of services (GNSS and terrestrial communication) taking the topography of the environment into account. The ability of SNSS to

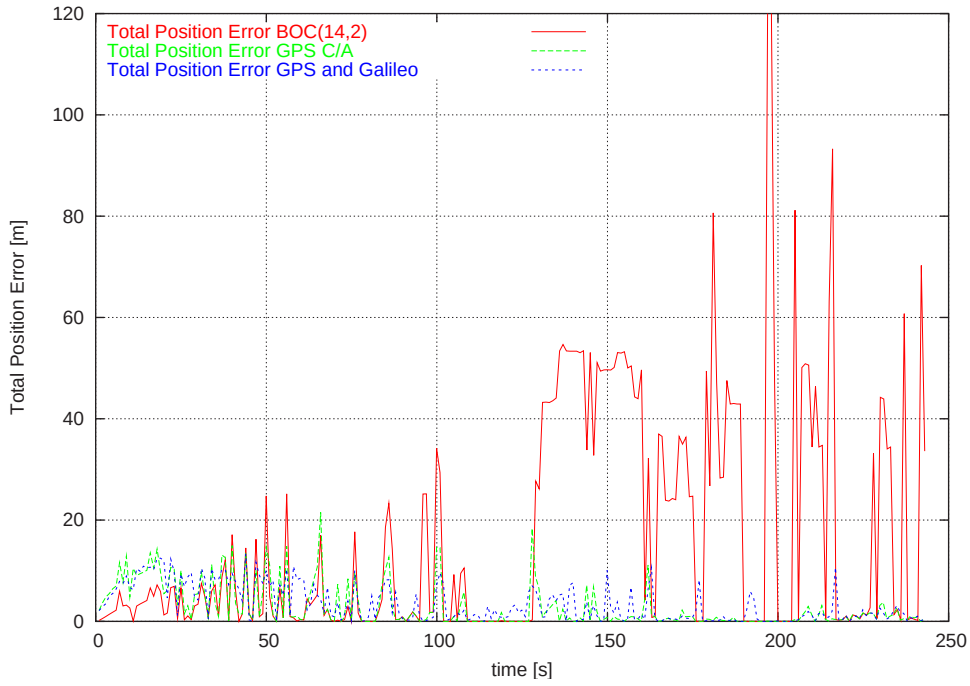

Figure 4.46.: The total error of the code tracking during simulation

generate polygons from digital elevation models (DEM), as described in section 3.3.2 together with the ray-tracing algorithm are key features for this analysis. The areas treated here are the Alps and the Pyrenees.

4.4.1. Large Scale Topographic Availability Analysis

For the modeling of the topographic surface of the Alps, the digital terrain model data DTED0, which is freely available from the US National Imagery and Mapping Agency (NIMA), was used. This elevation data has a grid resolution of 30 arc seconds, which corresponds to about 900 meters at a latitude of 48°. Thus, the DTED0 data of the relevant alpine area includes more than 1 million grid points.

The network overlay of the computation grid points, i.e. the positions of the receiver, was set to a resolution of 0.02°, which corresponds to approximately 2 km. The grid is generated by placing a way-point with a specified density in east and north direction at a given height above the terrain.

The simulation was performed for 4. April 2002 12:00 to 5. April 2002 12:00 with sampling time steps of 1 hour (thus 24 epochs were processed). The user elevation mask was set to 10°.

Of course a higher grid resolution than the selected 0.02° as well as a higher time sampling rate would have been preferred for a more detailed analysis of the signal shading,

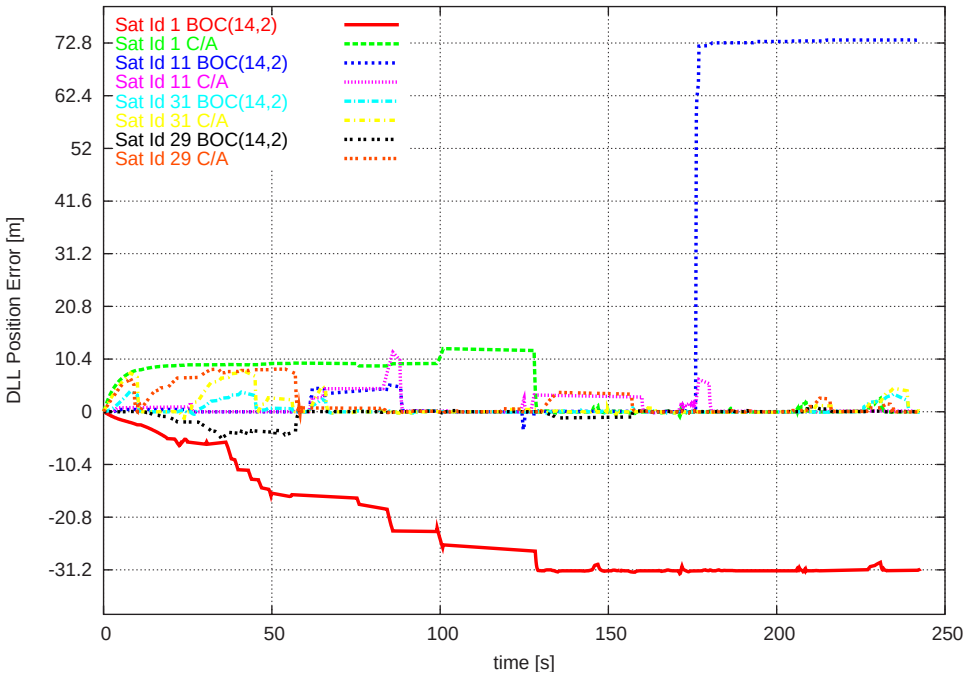

Figure 4.47.: The code tracking error for several satellites during the Stuttgart simulation

but with regard to the large extent of the alpine region a compromise had to be made between a reasonable amount of data for processing on the one hand and an acceptable degree of accuracy of the computation results on the other hand. After performing several tests with different grid resolutions it turned out that the computation of a grid with a resolution higher than 0.02° is not feasible for the whole area of interest due to the necessary calculating time. Further it does not make much sense to generate a grid that is much denser than the resolution of the topological data.

To compare and verify the results the same simulation was performed using the GIMAST analysis tool. This is a service volume simulator developed at the Institute of Geodesy and Navigation, University FAF in Munich. For this simulation no topological data was used, so basically an elliptic Earth was assumed for the area. The result is shown in figure (4.49). The green color corresponds to a minimum of six visible satellites and yellow corresponds to five.

The total accumulated computation time of the GPS availability simulation for the Alps and Pyrenees was about 400 hours, using several PCs/workstations with CPU power up to 2×2 GHz. In consequence of the constraints described above the significance of the resulting availability map is limited.

For the GNSS analysis the two areas of interest were split up into several DTM tiles, each one with an extension of 2°×2° (overlapping each other by 1°) resulting in 32 tiles – with the extension of 1°×1° – for the Alps and 13 tiles for the Pyrenees. For every

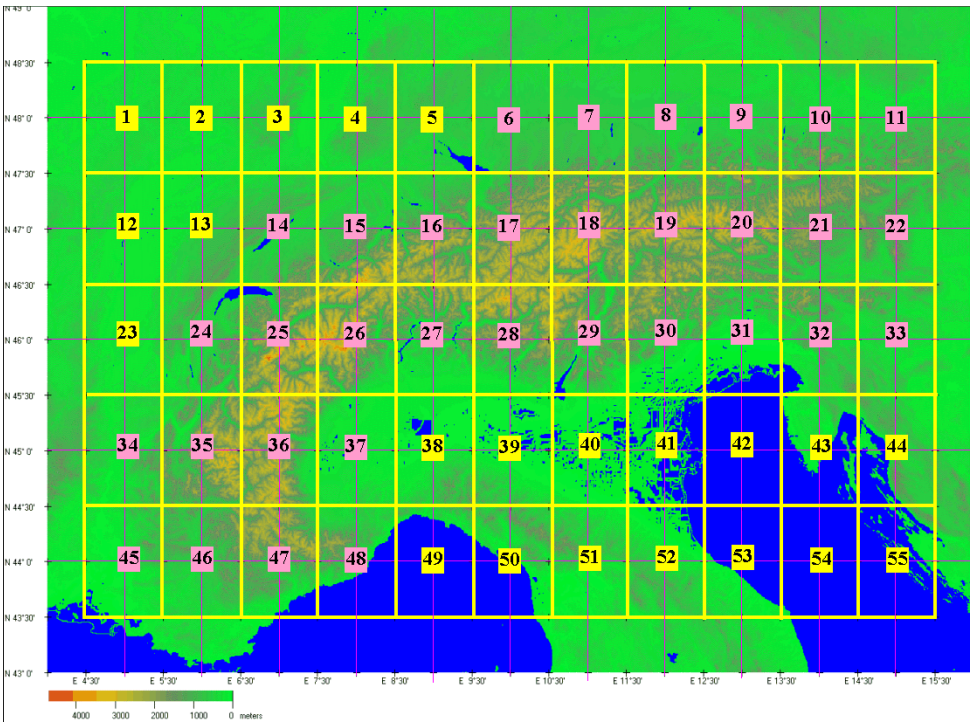

Figure 4.48.: Height encoded map of the Alpine region analyzed

tile the simulation was carried out separately, in order to keep the amount of data on a manageable level. The single map tiles resulting from the simulation were merged to one image afterwords. figure (4.48) shows the terrain of the Alpine area based on the DTED0 data together with an overlay of the grid of computation-tiles. The yellow rectangles correspond to the way-point grid and the red rectangles are the tiles containing the topographic data. For each simulation four red tiles were combined and a grid the size of one tile was generated in the middle of the combined topographic tile (the yellow rectangles). Thus, the area containing way-points is 1/4 of the topographic data for each tile.

Table 4.8 summarizes the relevant input data for the GNSS simulation for the whole areas of the Alps and Pyrenees.

Resolution of terrain grid	0.0083 degrees
Resolution of computation grid	0.0200 degrees
Sampling time steps	1 hour
Simulation period	Noon 4. April 2002 to Noon 5. April 2002
User elevation mask	10°

Table 4.8.: Input parameters for GPS availability simulation (with DTM)

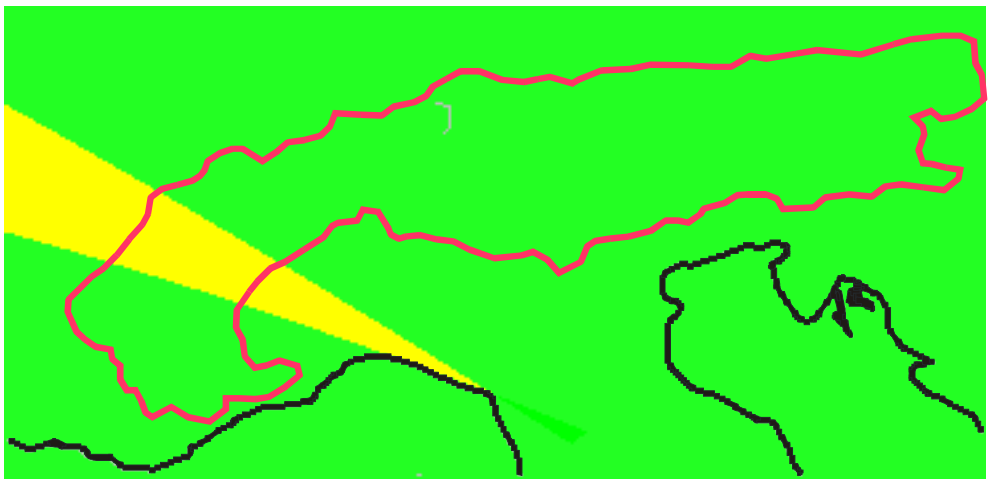

Figure 4.49.: Minimum number of visible satellites. Topography not considered.

Based on the minimum number of satellites visible during the computed epochs, the absolute minimum numbers during the period of 24 hours is displayed for each grid point in the resulting maps (figure (4.50) and figure (4.51)).

Even at first glance figure (4.50) clearly indicates a general decrease in the minimum number of visible satellites compared to figure (4.49) for wide areas in the Alps, caused by the shading of the mountainous terrain. Outside the area highlighted by the contour line, which indicates the Alpine region, the minimum number of visible GPS satellites is nearly constantly six (green marked area). This fully corresponds with the results from the simulation performed using the elliptical Earth model. The area of insufficient GPS availability of three satellites minimum (marked red) is quite small, compared to the total area. This is mainly due to the low resolution of the digital terrain model used, which causes a smoothing of steep and rugged slopes.

The yellow wedge-shaped area in the left part of the image represents a region in which the satellite visibility was reduced to 5 satellites (without regard to additional terrain effects) some time during the 24 hours simulation period. This is due to a worse satellite constellation in the western part of the Alps (see figure (4.49)). The clear-cut shape of this area is related to the GPS satellite footprints and the elevation mask chosen.

From the simulation results for the Pyrenees region given in figure (4.51) it can be easily seen, that in general the influence of the terrain on the visibility of the satellites is not as strong as in the Alps. From the simulation results for the Pyrenees region depicted in figure (4.51) it can be clearly seen, that in general the influence of the terrain on the visibility of the GPS satellites is not as strong as in the Alps. The minimum number of visible satellites in the near-plane area around the mountains is six. Also inside the area enclosed by the contour line, which highlights the mountainous region, large parts are marked green. This means that the satellite visibility is not reduced there by signal shading, as far as it can be simulated using the low resolution data, as it was the case in

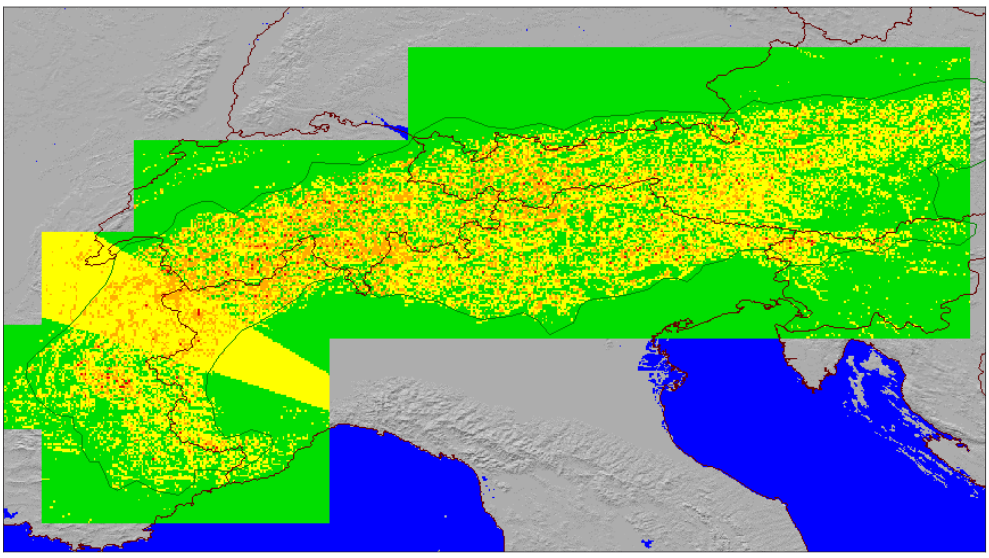

Figure 4.50.: GPS availability in the Alps

the underlying computation. Only in very few parts, mainly on the northern side of the Pyrenees' ridge, areas with a minimum of four or less visible satellites can be identified in figure (4.51). This observation is consistent with the fact that, in general, there are more satellites visible from the south than from the north.

This better visibility, compared to the results for the Alps, might mainly be due to the smoother formation of the Pyrenees terrain and the minor differences in elevation.

However, it has to be pointed out once again, that the two images can only give a low-resolution overview of the minimum visibility for the complete areas of interest.

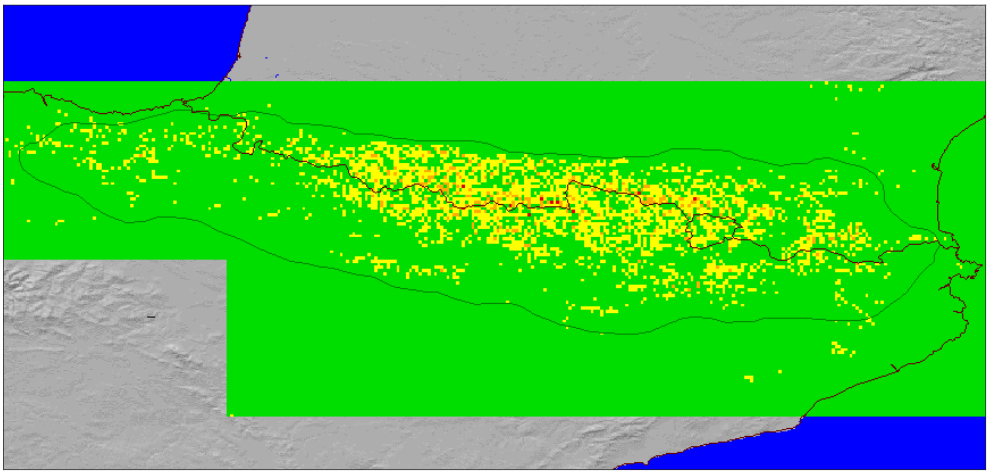

Figure 4.51.: GPS availability in the Pyrenees

4.4.2. Small Scale Topographic Availability Analysis

Because of the high demand on the computation resources needed for a detailed overall analysis for the Alps and Pyrenees, an exemplary availability simulation only for a small sample area (ca. 3×3 km^2) in the Bavarian Alpine region was performed, using a high resolution DTM grid of about 30 m and a corresponding computation grid.

In table 4.9 the input parameters for the simulation are listed. The results are shown in figure (4.52), which shows the minimum numbers of visible satellites for GPS, Galileo and a combined GPS Galileo constellation.

Lon	12°00’58.5" - 12°03’13.5"
Lat	47°39’33.8" - 47°41’8.3"
Resolution of DTM	approx. 30m
Resolution of grid	$0.00025° \times 0.00025°$
Sampling time steps	1 hour
Simulation period	Noon 4. April 2002 to Noon 5. April 2002
User elevation mask	10°

Table 4.9.: Input parameters for GPS availability simulation (with DTM)

Comparing the relief rendering of the 3-D terrain on the left side of figure (4.52) with the corresponding GNSS coverage simulation results, it is obvious that the best minimum GPS availability (up to six satellites minimum) is mainly obtained on the crests as well as on some of the south slopes. The limited visibility of satellites (three or even less) from some of the slopes in the south-west part of the sample area is due to the shading caused by the steepness of the slopes that block the signals from the south and east. In the plane valley of Bayrischzell (left side of the area) the minimum number is continuously four satellites.

Summarizing, it can be stated that, according to the simulation, at least three GPS satellites are available nearly in the whole sample area, which is enough for a two-dimensional positioning.

In figure (4.52) the availability analysis is also shown for Galileo. The structure of the resulting minimum-visibility map bears a strong resemblance to the results of the GPS simulation shown in the same figure. Again the shading of the quite steep slopes in the south-west of the sample area is recognizable. However, one obvious difference is the generally higher number of available satellites: in the critical areas, where only at least three GPS satellites are in sight, according to the simulation results Galileo provides a proper 3-D-positioning with minimum four satellites.

In addition the availability of the combined GPS and Galileo constellation was computed. The minimum-availability results displayed in the picture above show clearly, that sufficient satellite signals are available for the whole sample area, if both GNSS systems are

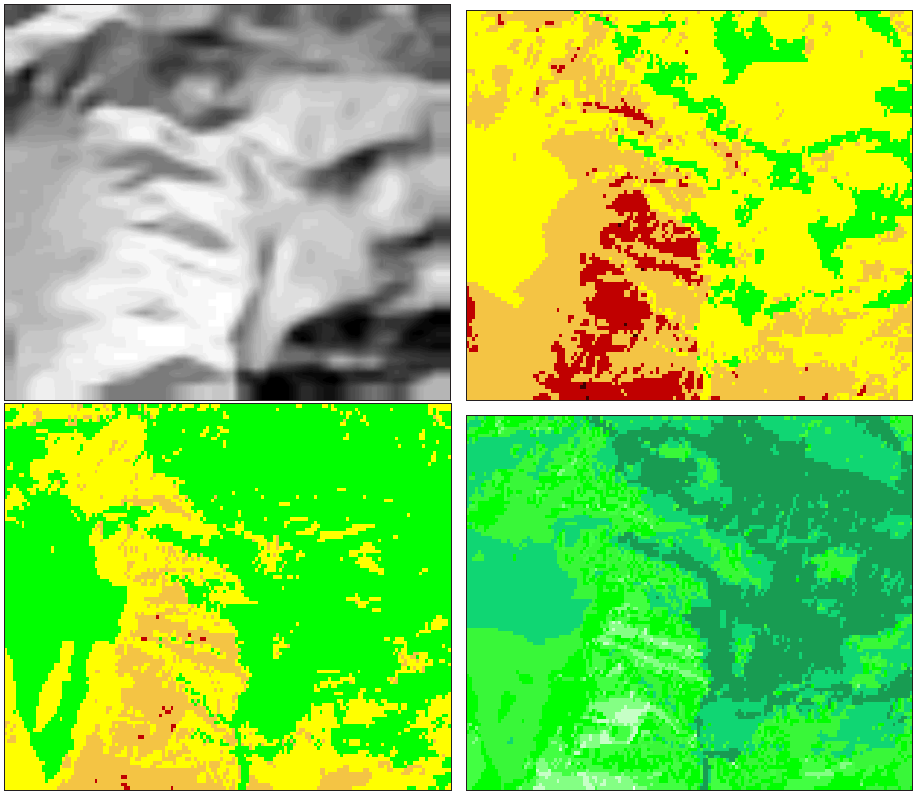

Figure 4.52.: GNSS availability in Sudelfeld. The upper left figure is a relief of the test area. The upper right shows the GPS Availability, the lower left figure shows the Galileo Availability and the lower right shows the combined GPS and Galileo availability.

used together. The absolute minimum value that was obtained from the detailed simulation is 6 satellites, resulting in an availability of 100 %; in many parts even 10 or more satellites (up to 14) are in sight all the time. This is a significant increase compared to the number of visible satellites using solely GPS or Galileo.

This is a very important result. Again we see that only by using GPS and Galileo *together*, a real improvement from the current situation is achieved. This was also observed in the simulations for the dense urban areas. The real benefit lies in the *combined* use of GPS and Galileo.

4.4.3. GSM Coverage Analysis

The following section presents results from GSM coverage analysis. This analysis was also performed with SNSS for the PARAMOUNT project.

As explained earlier, the computational burden for the different kinds of simulation runs can be extremely heavy e.g. about 400 hours for the GPS simulations, due to the large amount of data, which has to be processed.

Therefore, it is essential for a detailed analysis, in which very high resolution data is used, to restrict the simulation to a small area in order not to get in conflict with PC/workstation limitations or to exceed reasonable computation periods. For this reason, the relatively small sample area "Sudelfeld" was chosen, which is located close to the Bavarian town Bayrischzell in the German Alps. The following numbers clarify this.

The Sudelfeld sample area is only about 8 km^2 large compared to the approx. 200.000 km^2 of the whole of the Alps. Computation time for the Sudelfeld simulation (only three base stations) depicted in figure (4.53) was approximately 8 hours on a 1 GHz Personal Computer. Projected onto the whole Alpine area, this would mean a computational burden in the range of 23 years! It has to be noted further that this projection does not once take into account the increased process load due to the increased number of base stations – probably several hundreds– for the whole Alps.

The Sudelfeld sample area is a dedicated recreation area, predestined for skiing in winter and hiking in the summer time. It has to be noted that at the moment the Sudelfeld area was selected only to demonstrate the results from the detailed GPS/Galileo analysis and from the GMTS comparison between the estimated coverage of the provider D2/Vodafone and our own coverage simulations. It has not yet been identified as a candidate for a later test or pilot area. However, this could be the case after the G3-synthesis.

4.5. Summary

To summarize the results of this chapter:

- The implementation of the SNSS simulator was verified with regard to the
  - Thermal noise: Tests were performed for all signal structures currently implemented in SNSS and the resulting code and carrier phase errors compared to theoretical and experimental results. The transfer function of the dynamic system, when white noise was injected into it was also verified.
  - The multipath behavior was demonstrated to show the same behavior as the theoretical results in cases approaching the static limit, which is where these results are valid.
  - The transient behavior was also demonstrated to coincide with the theoretical statements.
- Multipath fading was simulated for various signal structures relevant to Galileo and GPS. The importance of the fading caused by the correlation process for dynamic applications was also shown.

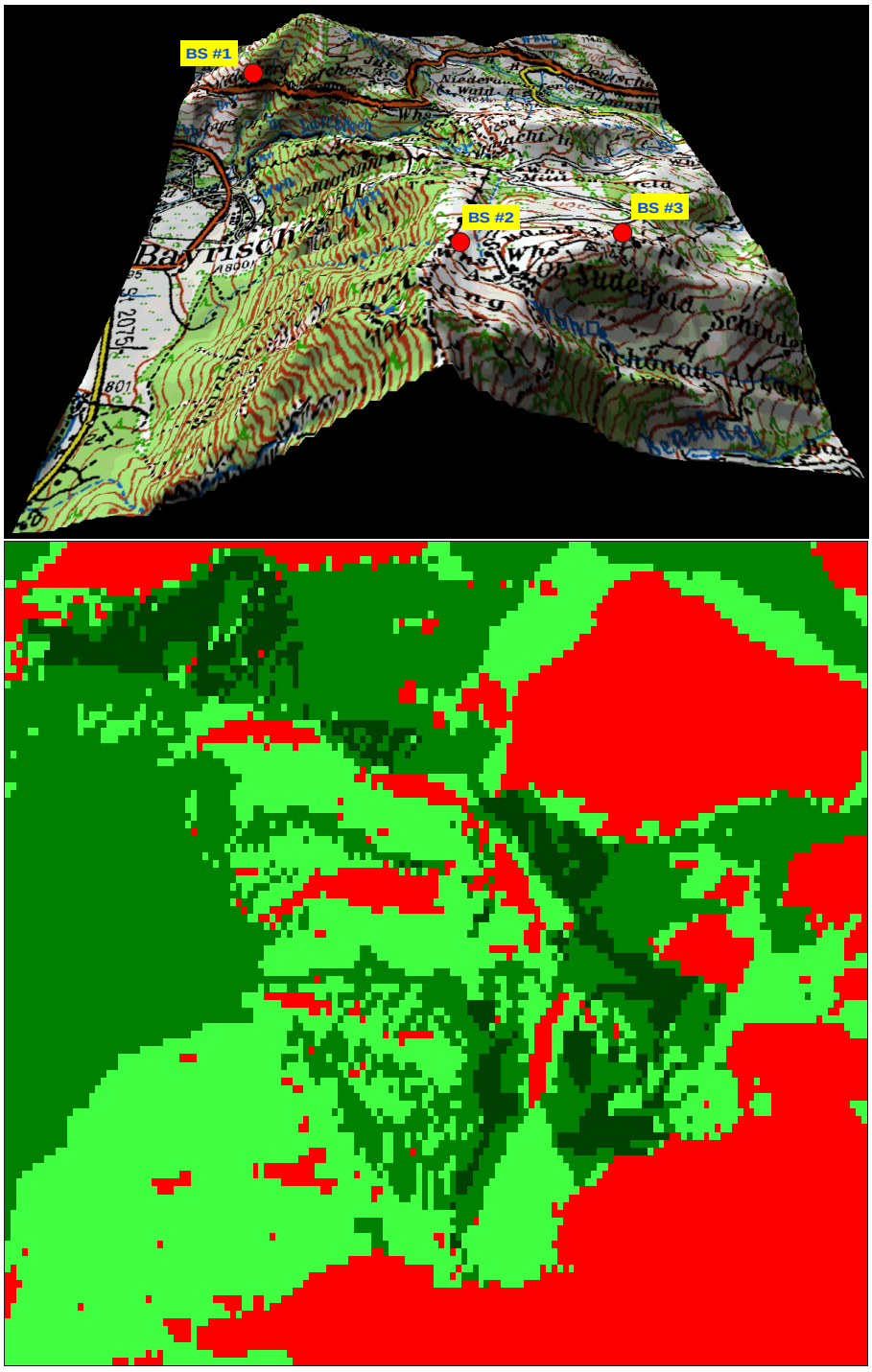

Figure 4.53.: GSM availability in Sudelfeld. The left figure shows the position of the GSM base stations in the test area. To the right the coverage is shown.

- Particular attention was given to the BOC(nx,x) signal structure where n is large. It was shown that under relatively well-defined conditions there is no immediate danger that the tracking loops loose lock of the correct peak. However, in a realistic environment where the boundary conditions become increasingly complex a certain danger exists that the DLL locks onto a false maximum of the auto-correlation function.

- Simulations were performed that demonstrated the behavior of the DLL for various values for the correlator spacing. It was shown that it is possible to choose a correlator spacing such that the zeros of the S-curve become second order and that this value can in fact safely be chosen with regard to the multipath and noise performance.

- Simulations in a moderate urban are were performed and it was seen that for the currently available GPS signals the multipath errors compared to the thermal noise on the code are much smaller. However, the real danger is the blockage of the direct signal while the receiver is able to track the indirect signal. This can cause extremely large errors.

- For the more advanced signal structures (RECT(10), BOC(14,2) etc.) the multipath errors become the dominant error source compared to the thermal noise.

- Simulations in a dense urban area were performed and it was shown that a combined constellation of GPS and Galileo will lead much better availability of a positioning solution. However, the accuracy will not necessarily improve significantly.

- GNSS availability analysis using topographic data was performed for the Alps and Pyrenees. As in the urban area simulations the analysis in for the mountainous regions show that the use of a combined GPS/Galileo constellation dramatically increase the availability.

- Finally, a GSM coverage in Sudelfeld was analyzed using a high-resolution DEM.

5. Theory and Analysis

In this chapter theoretical considerations concerning the various signal structures are contained. These are properties that are not directly of relevance to the simulator.

5.1. Signal Interference

In light of the definition phase of Galileo, in particular the allocation of frequencies and definition of signal structures, the interference between the existing GNSS (i.e. GPS and GLONASS) and several candidate signal structures for Galileo are considered.

5.1.1. Overlay on GPS

5.1.1.1. Interference of Galileo Signals with the M-code

The M code was designed to co-exist with the C/A code and the P(Y) code. Furthermore the M code will be a “stand alone" signal, i.e. it does not rely on the C/A code for acquisition like the P(Y) code does. From a NATO point of view this is an important feature because it allows the operator of the system to jam the C/A code, practically without affecting the M code and thus preventing un-authorized use of the navigation signal. In figure (5.1) the power spectra of the M code, the C/A code and two other codes with 2.046 MHz and 4.096 MHz chipping rate are shown.

It has been shown [Bet00a] that the overlap between C/A code and the M code is small enough to allow intentional jamming of the C/A code. For reasons explained later in this section it may be feasible to overlay Galileo signals with GPS. The chipping rate of the C/A code does not necessarily represent an ultimate upper bound on the chipping rate of a civil signal compatible with the DoD requirements. However analysis analogous to the one performed in [Bet00a] will have to be done if negotiations between the EC and the US Government are to be fruitful.

In this same context the location of the frequencies E1, E2 and E3 are right next to the GPS L1 and L2 bands. A look at figure (5.1) reveals that a spill over from the outside will have a far greater impact on the M code than interference from a code located in the middle. Fratricide will be much more difficult with the civil signals as immediate neighbors in frequency.

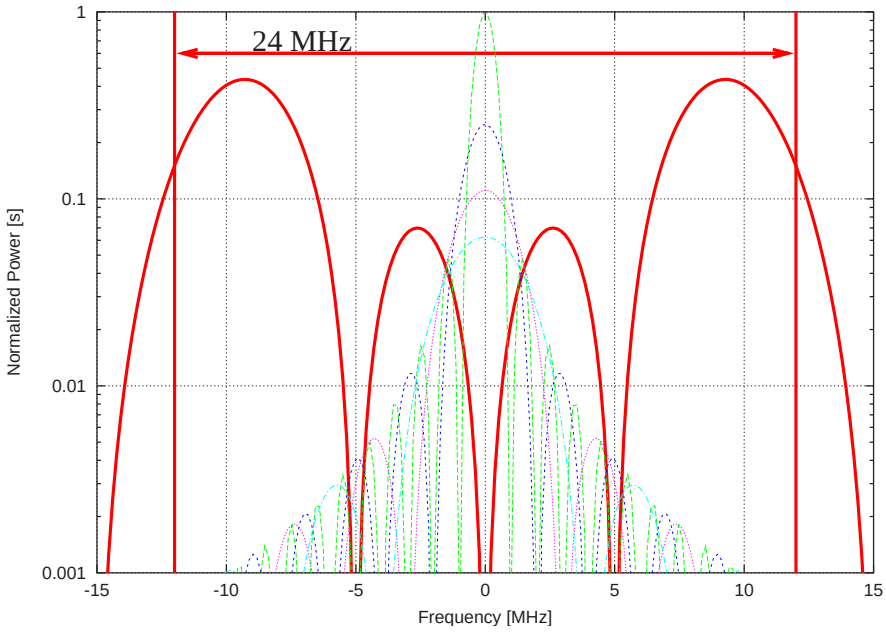

Figure 5.1.: Power spectral densities of the M code, GPS C/A code and codes with GALILEO 2.046 MHz and 4.096 MHz chipping rates

Analogous to the calculation of the interference of the M code with the C/A code[Bet00a] the interference of a potential overlay signal on GPS can be analyzed. The signal-to-noise-plus-interference ratio (SNIR) is given by

$$(C/N_0)_{\text{eff}} = \frac{C_s/N_0 \int_{-\beta_r/2}^{\beta_r/2} G_s(f)df}{\int_{-\beta_r/2}^{\beta_r/2} G_s(f)df + C_i/N_0 \int_{-\beta_r/2}^{\beta_r/2} G_s(f)G_i(f)df} \tag{5.1}$$

where $G_s(f)$ and $G_i(f)$ are the power spectral densities of the M code signal and the interfering signal respectively. β_r is the receiver front-end band-width and C_s and C_i are the received powers of the M code and the interfering signal. Assuming a nominal power-level of -158 dBW for the M code [BBC+00] and a receiver front-end band-width of 24 MHz, the effective signal-to-noise ratio is easily obtained as a function of the power level of the interfering signal. In the figure below the effective signal-to-noise ratio is plotted for a RECT signal with rectangular chip-shape and chipping rates of 1.023, 2.046, 3.069, 4.096, 8.192 and 10.23 MHz.

From figure (5.2) it is clear that an overlay on GPS will not affect the M code at all for all practical power-levels.

In case of jamming, where the J/S ratio is say 60 dB and the jammer has the same spectral characteristics as the overlay code, the M code will be degraded to 27, 24, 22 and 21 dB-Hz for chipping rates 1.023, 2.046, 3.069 and 4.096 MHz, respectively. However, in a situation where the civil codes are jammed, the M code is likely to be employed in spot

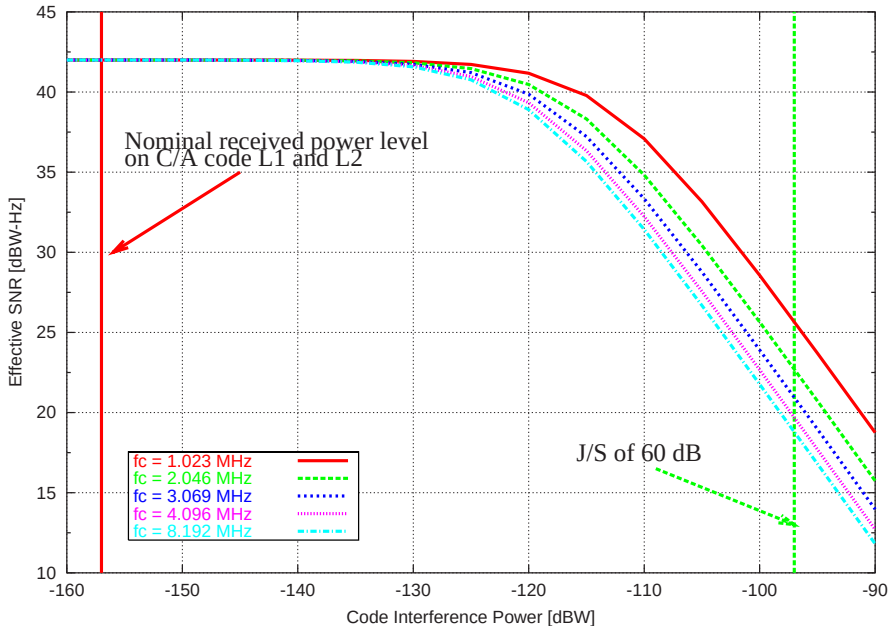

Figure 5.2.: The effective signal-to-noise ratio of a received M code signal as a function of the power-level of the interfering signal. For reference the nominal power-level of the C/A code and a jammer with a 60 dB higher power level is shown

beam mode, which increases its power by 20 dB. In that case the effective signal-to-noise ratio will be above 40dB-Hz in all cases.

5.1.1.2. Interference of Galileo Signals with the C/A code

The interference of the overlay signal with the C/A code is shown in figure (5.3). From the figure it is clear that if the overlay signal has the same nominal power level, it will practically not interfere with the C/A code. In fact, the overlay signal could have a power level almost 10 dB higher than the C/A code, and still not degrade the C/A code significantly, but the planned power levels for Galileo are nowhere near that kind of power. Interestingly, the higher the chipping rate of the overlay signal, the less it will interfere with the C/A code. The highest level of interference is caused by a signal having the same chipping rate as the C/A code, because the overlap term in the denominator in eq. (5.1) is maximized for such a signal.

5.2. Multipath in Combined Measurements

In this section we investigate the spatial correlation in multipath error. The two cases under consideration are correlation of the multipath error in a DGNSS scenario. In this case

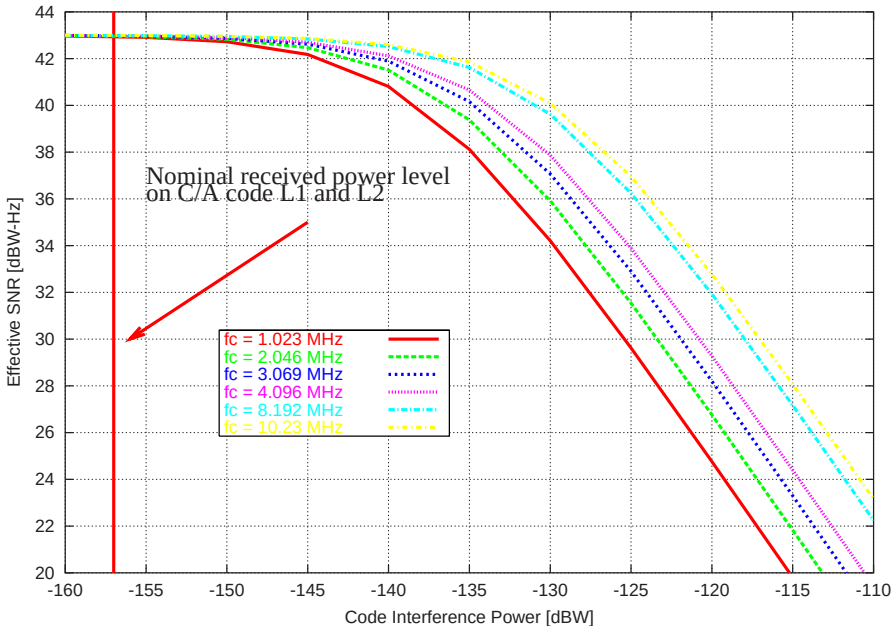

Figure 5.3.: The effective signal-to-noise ratio of the C/A code, caused by the overlay signal. A nominal power level of -157 dBW for the C/A code was assumed

the geometry of the signal propagation is different for different signals and the difference in multipath error is due to the difference in geometry.

The other case is concerned with the dependence between multipath error for two or more signals originating from the same source and received at the same receiver. Here the geometry is identical for the two signals, but the properties of the signals themselves are different. Thus the correlation difference in multipath error comes from the different signal structure.

5.2.1. Correlation of Multipath in DGNSS

In the following we like to answer the question, how multipath errors behave, if we consider a stationary scenario at the reference receiver, but shift the position of the mobile receiver. In our model this corresponds to changing the geometric delay at the mobile receiver, while keeping it constant at the reference station. The geometric delay cannot be identified directly with the spatial separation of the receivers, although they depend on each other. However, this simple model does not require the multipath to originate from the same reflector. For simplicity we assume the SMR to be constant and the same for both reflections.

Although multipath at a local station is a deterministic physical signal, we may consider multipath as being quasi-stochastic, if we look to the multipath propagation in the entire

coverage area of a GNSS satellite. In this coverage area we find a lot of different topography and obstacles with different reflection parameters, roughness etc.. Thus, the idea in our approach is always to discuss multipath on a regional scale, which means on the other hand that multipath is so irregular that it becomes very close to random.

The stationary code multipath error is given in terms of an implicit function, which presents a further complication to the analysis. In addition, the mathematical relations depend on the kind of signal processing implementation in a specific receiver (coherent, non-coherent, decision oriented, squaring, cross product etc.). In order to simplify the number of options we assume that the receiver is using a non-coherent early-minus-late detector for code tracking. This implementation is widely used in commercial GPS receivers. The basic expression that defines multipath error as a function of geometric delay for a non-coherent receiver in the stationary case is found by calculating the root of the S-curve. The result is written as $\Delta\tau = \Delta\tau(\delta, \eta, d)$.

For the phase error we use the Costas discriminator and in the static case we write the multipath phase error $\Delta\theta = \Delta\theta(\Delta\tau, \delta, \eta)$

We are now interested in the auto-correlation function[1] of $\Delta\tau$ and $\Delta\theta$ as functions of the variations in geometric delay. The parameters η and d are auxiliary values describing the shape of the auto-correlation function. After subtracting the mean value from the functions $\Delta\tau(\delta, \eta, d)$ and $\Delta\theta(\Delta\tau, \delta, \eta, d)$ one finds

$$\Phi_{\tau\tau}(\Delta\delta) = \frac{1}{\Xi_\tau} \int_0^{3T_c/2} \Delta\tau(\delta)\Delta\tau(\delta + \Delta\delta)d\delta \tag{5.2}$$

Similarly, we have for the auto-correlation function of the phase multipath error

$$\Phi_{\theta\theta}(\Delta\delta) = \frac{1}{\Xi_\tau} \int_0^{3T_c/2} \Delta\theta(\delta)\Delta\theta(\delta + \Delta\delta)d\delta \tag{5.3}$$

In order to evaluate these expressions, a small program was written. The tasks necessary are first to evaluate the multipath error by finding the root of the S-curve to yield $\Delta\tau$ and $\Delta\theta$ as functions of δ. The resulting look-up tables are then used to calculate the integrals in eq. (5.2) and eq. (5.3) for each value of $\Delta\delta$.

Intuitively, given a certain constant difference in geometric delay between receiver and reference station ($\Delta\delta$), these functions are the normalized integrals over all possible delays. This means, given that constant delay difference, the correlation between the multipath error at the reference and receiver, is the number given by the integral. If it is 1, then for that particular difference in geometric delay, the receiver and reference will have the same error. In a differential scenario, the error would cancel out in that case. If it is -1 the errors have the same magnitude, but different sign. Thus, the multipath error is doubled.

[1]The auto-correlation function we are talking about here should not be confused with the auto-correlation function from previous chapters. That was the auto-correlation function of the signal itself. Here the auto-correlation function of the multipath error is meant.

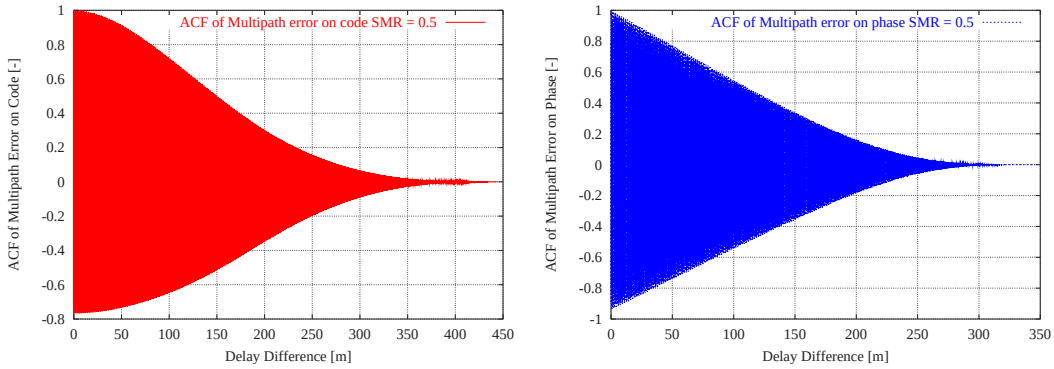

Figure 5.4.: Auto-correlation function of the multipath error for the code (left plot) and phase (right plot). Due to the very rapid oscillations the plots appear as colored regions. The SMR was η =0.5.

In figure (5.4) the auto-correlation functions of the multipath code and carrier phase error are plotted. Note that as with the multipath errors themselves the auto-corelation function oscillates very rapidly.

In figure (5.5) the auto-correlation functions of the multipath code and carrier phase error for the first two meters of relative geometric multipath difference between reference and rover. The curves for three values of the SMR ratio are shown.

If we take a look at the figures above we see that the correlation is not a simple monotonous function of $\Delta\delta$, it has very rapid oscillations with a wave length of 20 cm. As with the multipath error these oscillations originate from the wave length of the carrier.

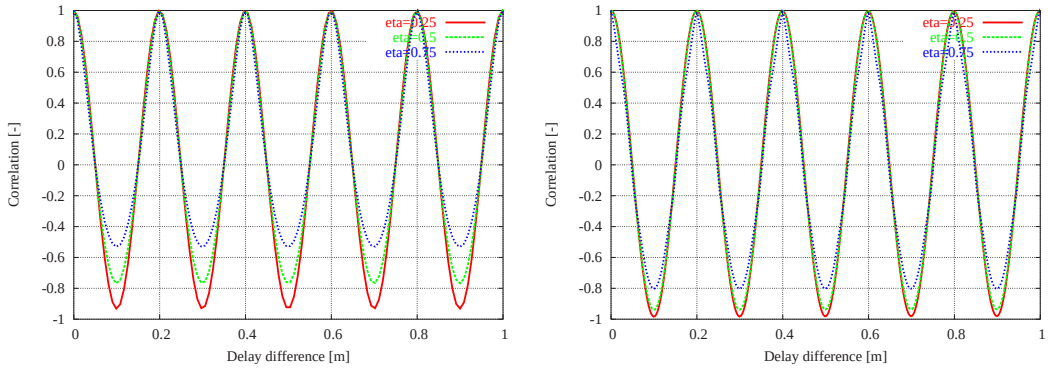

Figure 5.5.: Auto-correlation function of the multipath error for the code (left plot) and phase (right plot) for the first two meters. The SMR ratio was $\eta = 0.25$, 0.5 and 0.75

It may appear strange that the amplitude of the auto correlation function is *larger* for *smaller* values of η. This is due to the normalization: a correlation of -1 simply means that the multipath error at the two locations will have the same magnitude, but opposite

signs. The magnitude of the multipath error will of course decrease with decreasing η. This phenomenon simply reflects the increasing asymmetry of the multipath error curve with increasing η.

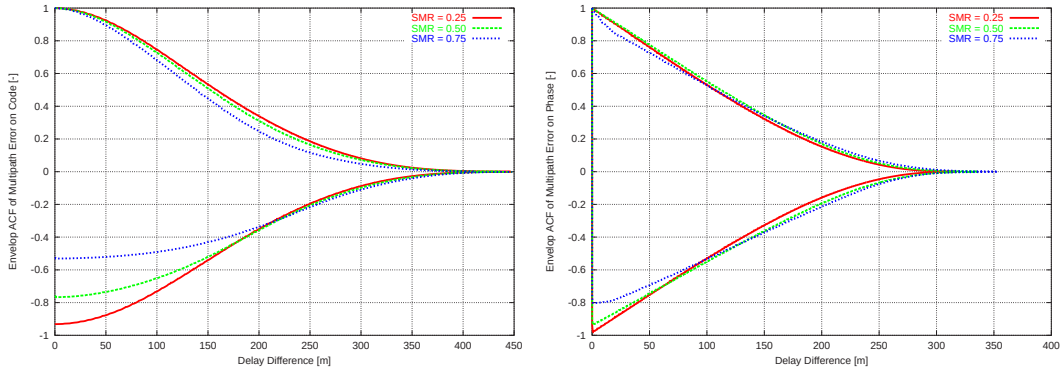

Figure 5.6.: Envelope curves of the auto-correlation function of the multipath error for the code (left plot) and phase (right plot). The SMR was $\eta = 0.25$, 0.5 and 0.75.

Although the auto-correlation function of both DLL and PLL are definitely non-zero, meaning that there is a correlation between the multipath at the receiver and at the reference, this information is not particularly useful for compensating multipath errors in differential mode. The reason is that the auto-correlation function oscillates very rapidly with the difference in geometric multipath (i.e. $\Delta\delta$) and it is impossible for a user to know the value of $\Delta\delta$.

When looking at the auto-correlation function for the DLL and the PLL one might think that the multipath error will tend to average out and become small. This is not the case, because the geometric multipath at the reference will only change due to motion of the satellite. Therefore the duration of a period of these oscillations will be typically of the order of 100 to 1000 seconds. As we have seen already the multipath error changes very rapidly for the moving user.

As soon as the user starts moving (even for low speeds of order of 10 m/s) the multipath error will start oscillating rapidly. Roughly speaking, every time the user traverses one wavelength of the carrier frequency, the multipath error will change sign. If there is multipath at the reference station, it will not change significantly in the time-interval considered and the multipath at the mobile receiver will simply oscillate around the multipath error generated by the reference station.

Bottom-line of these considerations is that DGNSS will not yield a reduced multipath error as compared to the single point case. On the contrary, in differential mode the multipath can only increase.

5.2.2. Correlation of Multipath in Multiple Frequency Systems

Future GNSS systems will definitely have satellites emitting signals on more then one frequency for civilian use. In this context it is to be expected that people will want to construct various linear combinations. The reason for this could be to estimate the ionosphere or to resolve the integer ambiguity of the phase measurements (TCAR).

Obviously, two signals in different frequency bands emitted from the same satellite will travel the same path. So, if there is multipath on one, there is also multipath on the other. Due to the identical geometry, the multipath on the two frequencies will be highly correlated.

To see how this works, the multipath error is calculated for two signals having identical geometrical multipath and identical SMR. The signals are infinite band-width, rectangular signals on the carrier frequencies $\omega_1 = 1.5$ GHz and $\omega_2 = 1.485$ GHz, i.e. the frequency separation was 15 MHz. The chip length was 300 m. Constructing the simple combination

$$\rho_{E_1E_2} := \frac{\rho_{E_1} + \rho_{E_2}}{2} \tag{5.4}$$

we can then calculate the multipath error of $\rho_{E_1E_2}$. The resulting multipath error can be seen in figure (5.7). The figure shows a very dense sampling of the multipath error. The multipath error oscillates very rapidly and thus the plot appears as a filled region.

The enveloping curve (i.e. the convex hull) in figure (5.7) is identical to the one for the classical C/A code case. The main difference is the appearance of two basic frequencies, corresponding to ω_1 and $\omega_1 - \omega_2$. These frequencies when multiplied with the geometric multipath are equivalent to a wave length of 0.2 m and 20 m. A closer look at figure (5.7) reveals this. The curve is basically a superposition of these two frequencies.

Generally, one can say that the multipath error of a combination like the one in eq. (5.4) will suffer the same amount of multipath error as the *worst* measurement in the combination. However, the shape and behavior of the multipath error will be a bit different.

To take a less hypothetical example let us consider the European frequencies filed at the WRC2000 in Istanbul and compare it to the GPS III case. With selective availability gone, the ionosphere will present one of the main error sources and thus the so-called ionosphere free combination is of importance. WAAS and EGNOS were motivated by the fact that, before GPS IIF is in place, aviation users are single frequency C/A-code L1 users and ionospheric corrections have to be provided by these augmentation systems. In Galileo it is hoped that ionosphere will not play such a significant role, because most users will have at least two frequencies available.

It is well known that the ionosphere free pseudo-range is obtained, using dual frequency measurements, by the expression

$$\rho^{12}_{\text{iono}} = \frac{\gamma\rho_1 - \rho_2}{\gamma - 1} \qquad \gamma := \frac{f_1^2}{f_2^2} \tag{5.5}$$

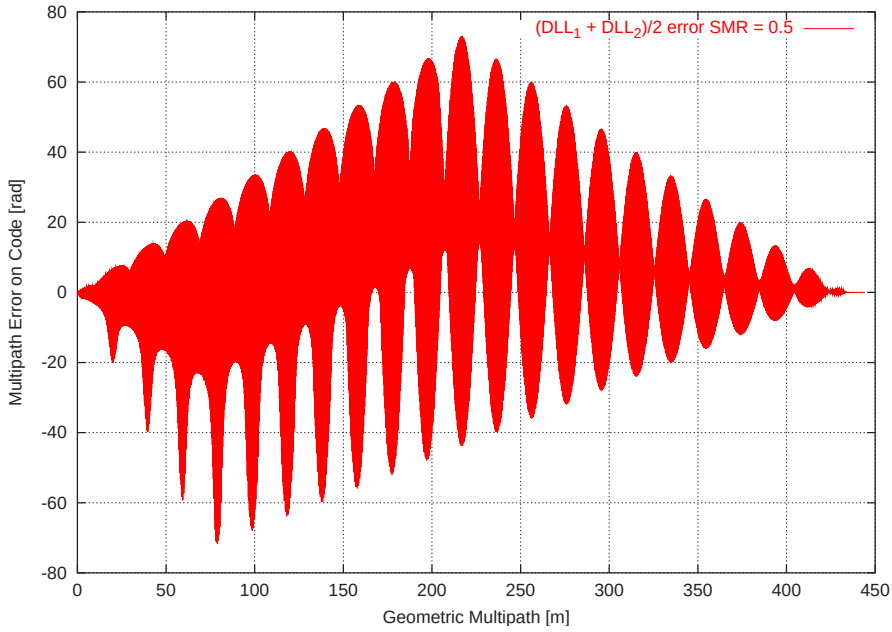

Figure 5.7.: Multipath error for the combination $(\rho_{E_1} + \rho_{E_2})/2$. The sigma was a C/A code with infinite bandwidth and standard correlator spacings. On the horizontal axis the geometric multipath is shown.

To put numbers on this we get using L1 ($f_1 = 1.575$ GHz) and E5a (L5) ($f_2 = 1.174$ GHz):

$$\frac{\gamma}{\gamma - 1} = 2.25 \qquad \text{and} \qquad -\frac{1}{\gamma - 1} = -1.25 \tag{5.6}$$

So in this case the measurement on L1 weighs almost a factor of two more than the L5 measurement.

We note that the corrected pseudo-range ρ_{iono} is a linear combination of the dual frequency measurements ρ_1 and ρ_2 and of course not free of remaining ranging errors on ρ_1 and ρ_2. However, amplification factors above play a key role in the remaining error budget (noise, multipath). As just demonstrated the higher amplification factor is on the higher carrier frequency. This implies that the higher performance signal should also be on the higher carrier frequency.

To demonstrate this we consider three cases in the following. The first case corresponds to the early suggestions for the Galileo signals on E1, E2 and E5. The signals assumed for E1 and E2 are square-root raised cosine signals with roll-off 0.2 and chip length 97 m. The signal on E5 was assumed identical to the P code and the L5 GPS III signal, i.e. rectangular chips, filtered to the first main-lobe. Chip length 29.3 m.

The second case corresponds to the GPS III signals on L1, L2 (GPS C/A code) and L5 (RECT(10) confined to the first main-lobe)

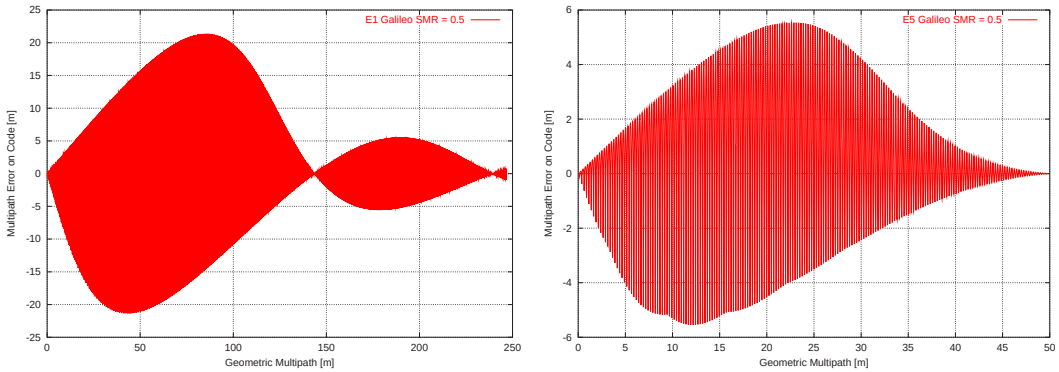

Figure 5.8.: Multipath error for Galileo signals on E1 and E5. On the horizontal axis the geometric multipath is shown.

The third case corresponds to the current open access signal on Galileo on L1 and E5a. The signal on L1 is assumed to be a BOC(2,2) in a bandwidth of 8.184 MHz and the E5a signal to be a RECT(10) confined to the first main-lobe.

Case 1 If there are three frequencies available, we could be lead to assume that two ionospheric free observations could be made. For the thermal noise this may approximately apply when there are two high frequencies and a low one: the main contribution to the thermal noise comes from the higher frequency (higher multiplication factor) so for E1, E2 and E5 it is tempting to try something like

$$\rho_{\text{iono}}^{\text{Galileo}} = \frac{\rho_{\text{iono}}^{15} + \rho_{\text{iono}}^{25}}{2} \tag{5.7}$$

where the two ρ on the right side are iono-free combinations as in eq. (5.5). For independent measurements on E1 and E2 the noise will be reduced by a factor close to $\sqrt{2}$ and this argument can safely be applied to the thermal noise.

For multipath, however, the situation is entirely different, because the multipath error on E1 and E2 are very dependent. To see this we plotted the multipath error for the combination in eq. (5.5).

These signals are shown in figure (5.8). Looking at the vertical axis we see that the multipath performance is very different. The E5 signal has a maximal multipath error of c.a. 5 m while the error on the E1 signal exceeds 20 m.

In figure (5.9) the combinations in eq. (5.7) for all three frequencies and for E1 and E5 alone using eq. (5.5) are shown. We see that there is practically no difference between using only two or all three frequencies what multipath is concerned. The only improvement is that the envelop is superimposed by the beat frequency between E1 and E2.

More importantly, the only visible influence of the E5 measurement is a slight dent in the beginning of the envelope in figure (5.9). So evidently the „good” frequency is suppressed

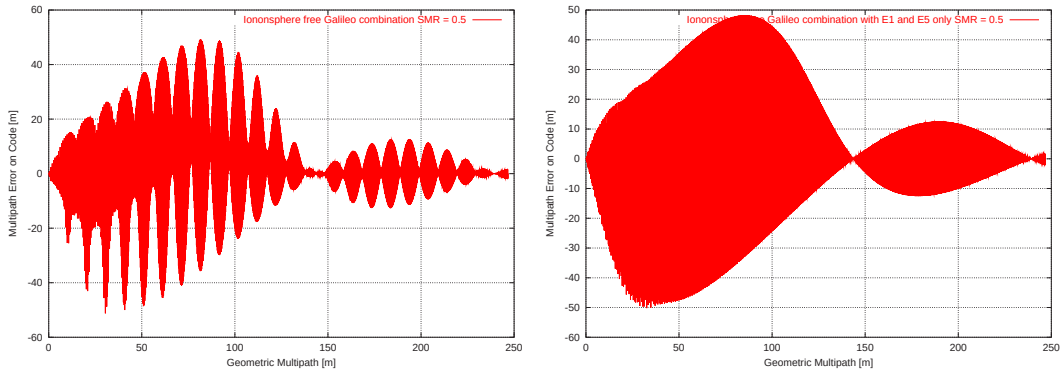

Figure 5.9.: Multipath error for the ionosphere free combinations using the Galileo signals on E1, E2 and E5. The right plot only uses E1 and E5 as described by eq. (5.5). On the horizontal axis the geometric multipath is shown.

and the „bad" is amplified. The situation would be more ideal if the „good" frequency was amplified and the „bad" one suppressed.

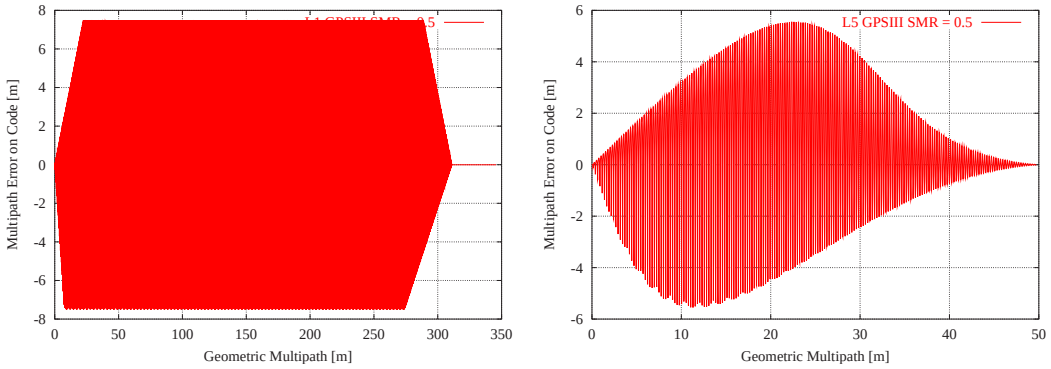

Figure 5.10.: Multipath error for the L1 and L5 signals on GPS III.

Case 2 Now let's compare this to the GPS III case. The signals assumed on L1 and L2 had chip length 293 m, infinite bandwidth and a correlation separation of $0.1T_c$. The L5 signal was identical to the E5 signal above, except for a shift of the center frequency to 1174 MHz.

The multipath envelopes are shown in figure (5.9). Here the maximum multipath errors are comparable, even though the L1 signal is not as good as the L5 signal, the discrepancy is not as bad as with E1 and E5 in Case 1.

In figure (5.11) the ionosphere free combinations using eq. (5.5) and is plotted. The first thing we notice is that the combination eq. (5.7) is not good in the GPS III case, because the frequencies are more evenly distributed. This leads to a higher amplification factor for

example between L2 and L5. However, if we compare figure (5.11) and figure (5.9), the combination eq. (5.5) easily outperforms case 1.

It is important to realize that even though this may seem a drawback here, it is very important in other contexts that the frequencies are properly distributed. This is especially true, when multiple frequencies are to be used for ambiguity resolution. Further, as seen in case 1, the improvement is not significant even if the frequencies are distributed as in the Galileo case.

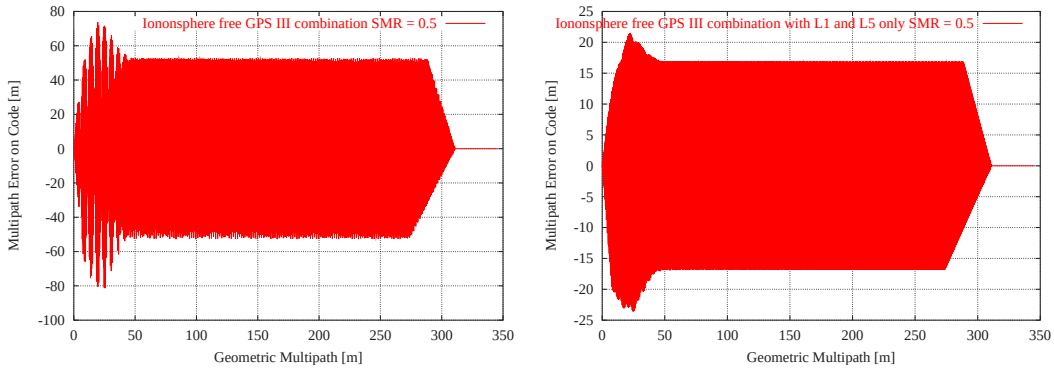

Figure 5.11.: Multipath error for the ionosphere free combinations for GPS III on L1, L2 and L5 using eq. (5.7) to the left and for GPS III L1 and L5 only to the right.

Case 3 The third case represents the current proposal of the EC Signal Task Force. In figure (5.12) the ionospheric free combination according to eq. (5.5) is shown. We notice that the range where the delay produces a multipath error is shorter than in Case 2. For short delays the multipath on the ionospheric free range is slightly larger than in Case 2. This is due to the bandwidth limitation. With infinite bandwidth the maximum value of the multipath errors in Case 2 and 3 would be identical.

Generally, what is learned from all this is that

- A combination of two frequencies ω_1 and ω_2 yields a multipath error which has an envelop that consists of the weighted sum of the individual envelopes.
- The resulting multipath error is superimposed by the beat-frequency $\omega_1 - \omega_2$.

For more than two frequencies the arguments above apply successively. In particular, the properties of independent measurements, which lead to a square root reduction in error (thermal noise), are not valid.

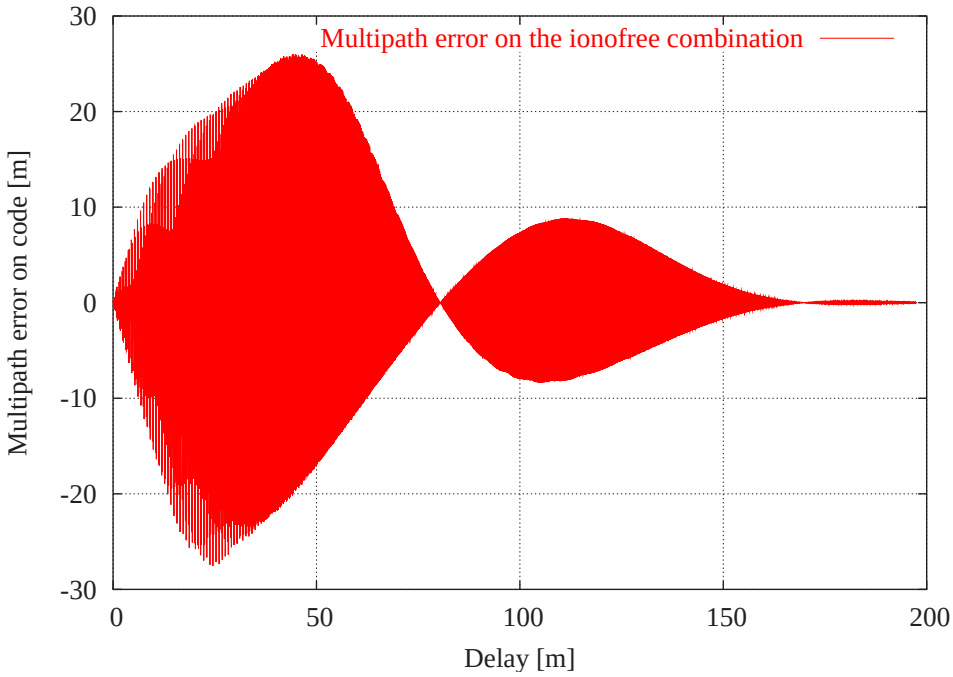

Figure 5.12.: Multipath error for the ionosphere free combinations for Case 3.

6. Summary and Conclusions

Summary In this thesis an end-to-end simulation was implemented encompassing the important effects from the user segments point of view. The modeling and implementation aimed to take all the relevant features into account that have a direct and significant impact on the performance of a GNSS receiver. In particular, emphasis was on the effects that are hard to formulate and treat theoretically, such as non-linearities, stochastic processes and the highly complex boundary conditions generated by the interaction of the signal with the environment.

While it is important to develop a model suitable for implementation as a simulation, the development of the substantial theoretical ground work necessary led to many interesting insights for the better understanding of the nature of a GNSS.

The three most important parts of the model development are the signal model, the signal propagation model and the receiver model.

The signal model is an extension of the well-known signal modeling used to describe GPS signals. The present model was extended to include any sort of BOC-modulation, SRC chip-shaping and a modulation that I have called SOC modulation[1] for “sinusoidal offset carrier" (see section 2.2.7).

It was also shown analytically, that the BOC signal can be tracked in side-band mode and an exact expression for the form of the corresponding correlation function was derived (see section 2.2.8). The side-band correlation function does not have multiple maxima. This property is extremely important for the acquisition and correct tracking of the BOC (and SOC) signal.

A new and more accurate expression for the pre-detection integral[2] was derived (see section 2.2.4.). This was an important step in order to arrive at a valid description of the multipath effect for a moving user.

The most important part of the signal propagation model is essentially a ray-tracing algorithm together with the application of the Fresnel equations. This is a 3-D exact specular ray-tracing, which was derived and implemented during the work accompanying this thesis (see sections 2.3.2 and 2.3.3). The results confirm how important it is to use a deterministic model for the multipath effect as it generates well-defined, realistic dynamics that are responsible for the multipath errors.

[1] During the writing of the thesis, this type of modulation has found its way into the literature under the name “linear offset carrier" (LOC).

[2] The pre-detection integral describes the integrate-and-dump process in the correlators

Beside the signal model the receiver model constitutes a major part of this work. Essentially it is a dynamic modeling of the tracking process (DLL and PLL). As the model is based on continuous calculus it was a challenge to incorporate the effects of the noise processes. However, this was solved by using the Îto calculus to extend the ordinary differential equations to stochastic differential equations (see section 2.5.4).

Another challenge during the development of the receiver model was introducing the effect of the oscillator instabilities. As the oscillator noise in general is a complex stochastic process, it is not realistic to assume a Gauß-ian process. To make matters worse, some part of it (the so-called flicker term) cannot readily be described in the time domain (see section 2.5.3 and references given there). To be able to integrate the oscillator effects realistically into the receiver model, a model was developed that is expressible as a set of stochastic differential equations. The method is based on a tricky expansion of the spectral representation of the oscillator noise process, which is then transformed back into the time domain.

All these models were developed such that they could be integrated into a larger end-to-end simulation afterward. This means that the individual models have the appropriate interfaces and the degree of detail must be compatible.

The implementation was verified by comparing the results to known theoretical expressions and an indirect experimental verification was performed in the sense that some of the theoretical formulas have been compared with experimental data.

Important Results Most analysis of the multipath effect in the literature involve analysis of the so-called envelope curves. The main assumption for these calculations are on the magnitude of the reflected signal, the absence of signal dynamics and the fact that the line-of-sight signal is available.

As shown in this thesis (see section 4.2.2), the static receiver assumption is only valid for extremely low velocities (i.e. when the receiver moves slower than 0.1 to 1 m/s). In addition to the well-known fading caused by the tracking loops, the effect stemming from the pre-detection process (or correlation process) contributes significantly toward reducing the bias resulting from the loop fading[3] (see section 4.2.2). This is a result from the more accurate analysis of the pre-detection integral already mentioned.

Comparing the GPS C/A signal structure (BPSK(1)) with the GPS modernized signals (GPS L5 BPSK(10)) and the Galileo signals (BOC(2,2), BPSK(5), BPSK(10) and BOC-(15,10)) shows an improvement in thermal noise by roughly an order of magnitude. For the multipath error this is generally also true. However, the multipath error caused by short multipath delays (i.e. up to 10 m) will be similar for all signal structures. Interestingly, the short range multipath delays seem to be the most common in an urban situation (see section 4.3). The consequence is that as the thermal noise is reduced for say, a BPSK(10)

[3]The mean value of the multipath error in the static case is *not* zero, thus the filtering due to the tracking loops will not eliminate the multipath error completely

signal compared with a BPSK(1) signal, the multipath effect due to short range delays is not (compare figure (4.38) and figure (4.39) on pages 195 and 196, respectively). In the GPS C/A code (BPSK(1)) case the multipath error is of the same magnitude or even larger than the thermal noise. In the P-code case this is opposite; the multipath effect is the critical effect.

In general, it can be commented that as the signal structures become more advanced and powerful, the more important it is to reduce other error sources in the GNSS system. For example it would be appropriate for Galileo to provide more accurate orbit and clock parameters than the GPS system does to improve the overall positioning accuracy.

Simulations were performed for dynamic situations in realistic urban environments (Oedekoven and Stuttgart, see section 4.3). The most significant errors are mainly generated in situations where the direct line-of-sight is blocked and the receiver is able to track the indirect signal. In theory, this can cause range errors of practically any magnitude. This error is mainly governed by the geometric situation and is relatively independent on the signal structure.

In all the simulations involving complete constellations it was clearly demonstrated that under harsh conditions (urban canyons and mountainous areas), the combined use of Galileo *and* GPS can increase the availability and quality of the services significantly. Thus the interoperability between GPS and Galileo is very important.

A further result from the analysis and the simulations is what I call the dynamic near-far effect. This effect is an error induced in the tracking loops when the receiver passes relatively close to a signal transmitter[4] (see section 4.2.2.1 and section 2.1.3). In this case the higher derivatives of the line-of-sight can become quite large even if the receiver is traveling with constant velocity. This causes range errors in the tracking loops due to their transient response. In some cases it can even cause the receiver to loose lock. When the signal originates from a satellite this effect is not expected to be important. However, for pseudolites this could be important.

Regarding BOC signals it is usually considered a great problem that the BOC auto-correlation function has multiple maxima. This issue was systematically analyzed in simple simulation as well as in full-scale simulations involving 3-D city models and the results can be summarized as follows.

If the tracking loop tracks the correct maximum of the auto-correlation function initially and the line-of-sight is never disrupted then there seems to be little or no danger that the tracking loop will slip onto another (wrong) local maximum of the auto-correlation function. This is the case even if the receiver is subject to extremely low signal-to-noise ratios or extreme multipath conditions (see section 4.2.3). Oscillator instabilities also do not seem to be a problem in this respect.

[4]This effect has nothing to do with the classical near-far effect, which relates to the change in power-level with distance to the transmitter.

However, in more realistic situations where the direct line-of-sight is temporarily blocked, it is possible for the receiver to start tracking the BOC signal on a wrong peak (see section 4.3.1 and figure (4.40)). This can be a problem for stationary as well as for moving receivers.

Two possible approaches to ensure correct BOC tracking were presented in the thesis. The first approach involves using the side-bands to acquire and track the BOC signal and then transfer to the processing of the full auto-correlation function of the BOC signal. The side-band signal is tracked parallel to the full BOC tracking as a monitor. The importance of the results for the side-band tracking is that the side-band can be tracked *without* additional side-band suppression filters, as these filters would introduce an additional delay. Thus the same front-end and the same signal conditioning can be applied for the full auto-correlation function tracking as for the side-band tracking.

Another possibility is to chose the correlator spacing such, that the “wrong" zeros of the S-curve are not stable tracking points anymore (see section 4.2.4). This approach is not optimal as the correlator spacing is limited to this particular value.

The correlation of the multipath effect in a differential GNSS scenario was analyzed for a simple case (see section 5.2.1). The result is that there is a strong correlation between the multipath effect at the reference and at the user. However, due to the rapid oscillations of the multipath effect this correlation varies between ± 1 as the difference in multipath delay changes by half a wavelength (approximately 10 cm). Therefore it seems difficult to use this information to improve the differential solution.

When using multiple frequencies to compensate for the ionospheric effect, measurements are combined in a certain way (ionospheric free combination). As is well-known this leads to an amplification of the noise (see section 5.2.2). In particular, the error on the measurement on the higher frequency is amplified more than the lower frequency measurement. The ionospheric free combination of also amplifies the multipath error. In the GPS modernized case, using L1 (BPSK(1)) and L5 (BPSK(10)) the multipath error on the ionospheric free combination is dominated by the L1 measurement. This is unfortunate as the multipath performance of the L1 measurement is worse than that of L5.

A. Definition of Constants and Functions

This appendix contains a list of commonly used constants and functions. In this thesis the sinc-function is defined as[1]

$$\mathrm{sinc}(x) := \frac{\sin(x)}{x} \tag{A.1}$$

The Heaviside step function is defined by:

$$\theta(x) := \begin{cases} 0 & x < 0 \\ 1 & x > 1 \end{cases} \tag{A.2}$$

The symmetric unit rectangle:

$$\mathrm{Rect}(x) := \theta(x + 1/2) - \theta(x - 1/2) \tag{A.3}$$

The triangular function $\mathrm{Tri}(\cdot)$ is defined as:

$$\mathrm{Tri}(x) := \begin{cases} 0 & \text{for } |x| > 1 \\ 1 - |x| & \text{for } |x| \leq 1 \end{cases} = (1 - |x|)\theta(1 - |x|)\,, \tag{A.4}$$

The sine integral $\mathrm{Si}(x)$ is defined as

$$\mathrm{Si}(t) := \int_0^t \frac{\sin(x)}{x} dx \tag{A.5}$$

[1] Sometimes this function is defined with a factor of π, but we will refrain from that

B. Relations and Identities

A trivial identity related to the complex conjugate of a geometric series:

$$\left(\sum_{k=1}^{N} a^k\right)\left(\sum_{j=1}^{N} a^{-j}\right) = N + \sum_{k=1}^{N-1}(N-k)(a^k + a^{-k}) \tag{B.1}$$

Standard geometric series ($a \neq 1$):

$$\sum_{k=1}^{N} a^{k-1} = \frac{1-a^N}{1-a} \tag{B.2}$$

Integral representation of the delta distribution:

$$2\pi\delta(x-a) = \int_{-\infty}^{\infty} \mathrm{e}^{i(x-a)t}\, dt \tag{B.3}$$

The derivative of the Heaviside function:

$$\theta'(x) = \delta(x) \tag{B.4}$$

The Fourier transformation of $\cos(at)$ and $\sin(at)$:

$$\begin{aligned}
\mathcal{F}\{\cos(at)\} &= \pi\delta(\omega-a) + \pi\delta(\omega+a) \\
\mathcal{F}\{\sin(at)\} &= i\pi\delta(\omega+a) - i\pi\delta(\omega-a) \\
\mathcal{F}^{-1}\{\cos(a\omega)\} &= \frac{1}{2}\delta(\omega-a) + \frac{1}{2}\delta(\omega+a) \\
\mathcal{F}^{-1}\{\sin(a\omega)\} &= \frac{i}{2}\delta(\omega+a) - \frac{i}{2}\delta(\omega-a)
\end{aligned} \tag{B.5}$$

The Fourier transformation of the Triangle function:

$$\mathcal{F}\{\frac{a}{2}\,\mathrm{Tri}(at/2)\} = \mathrm{sinc}^2(\omega/a) \tag{B.6}$$

Representation of a square wave in terms of the Heaviside step function:

$$\mathrm{Sqr}_n(x) = (-1)^n[\theta(x+T_c/2) + \theta(x-T_c/2)] + 2\sum_{k=-n+1}^{n-1} \theta(t+kT/2n) \tag{B.7}$$

where the parameter n counts the number of square-wave periods within a symbol. For $n = 1$ eq. (B.7) reduces to the Manchester case.

Mean value theorem for integrals Assume f and g are continuous on $[a, b]$. If g never changes sign in $[a, b]$ then, for some c in $[a, b]$, we have

$$\int_a^b f(x)g(x)dx = f(c)\int_a^b g(x)dx \tag{B.8}$$

The proof of this theorem can be found in most text books on calculus, e.g. [Apo69]

C. Derivation of the ACF for the PLL

In section (4.1.1) the auto-correlation function of the stochastic process resulting from the PLL was simulated and compared to the theoretical result. In this appendix the auto-correlation function of the PLL is derived.

It can be shown that the auto-correlation function of a Gauß-ian white noise process sent through the linear transfer function $H(\omega)$ is given by [Lin72]

$$R(\tau) = \frac{1}{2\pi} \int H(i\omega) H(-i\omega)\, \mathrm{e}^{\omega\tau}\, d\omega \tag{C.1}$$

Further, it can be shown that the closed loop transfer function of the linearized PLL in eq. (2.192) is given by [Spi73, Eis97]

$$H(\omega) = \frac{\omega_n^2 + i2\xi\omega_n\omega}{\omega_n 2 + i2\xi\omega_n^2 - \omega^2} \tag{C.2}$$

Thus we are confronted with calculating the inverse Fourier transformation of the square of the magnitude of the transfer function $H(\omega)$. This is straight forward, although a bit cumbersome.

The obvious way is to use the residuum theorem. We then have to identify the poles of the integrand in eq. (C.1). The integrand of eq. (C.1) has four poles at $\omega_\pm$ and $\bar{\omega}_\pm$ where

$$\omega_\pm = \omega_n \left(i\xi \pm \sqrt{1-\xi^2} \right) \tag{C.3}$$

and the bar means complex conjugate. The denominator of the integrand in eq. (C.1) can now be written as

$$D(\omega) := (\omega - \omega_+)(\omega - \omega_-)(\omega - \bar{\omega}_-)(\omega - \bar{\omega}_+) \tag{C.4}$$

There are two poles in the upper complex plane of ω and two in the lower. Assuming τ to be positive we can close the integration curve in the upper complex ω-plane. The fact that the auto-correlation function must be symmetric, saves us the trouble of considering the case $\tau < 0$. Using the residuum theorem, we can now write the auto-correlation function

as

$$\begin{aligned} R(\tau) &= \frac{1}{2\pi}\int H(i\omega)H(-i\omega)\,\mathrm{e}^{\omega\tau}\,d\omega \\ &= -i\lim_{\omega\to\omega_+}\frac{(\omega-\omega_+)(\omega_n^4+4\xi^2\omega_n^2\omega^2)}{(\omega-\omega_+)(\omega-\omega_-)(\omega-\bar{\omega}_-)(\omega-\bar{\omega}_+)}\,\mathrm{e}^{\omega\tau} \\ &\quad -i\lim_{\omega\to\omega_-}\frac{(\omega-\omega_-)(\omega_n^4+4\xi^2\omega_n^2\omega^2)}{(\omega-\omega_+)(\omega-\omega_-)(\omega-\bar{\omega}_-)(\omega-\bar{\omega}_+)}\,\mathrm{e}^{\omega\tau} \\ &= -i\frac{(\omega_n^4+4\xi^2\omega_n^2\omega_+^2)}{(\omega_+-\omega_-)(\omega_+-\bar{\omega}_-)(\omega_+-\bar{\omega}_+)}\,\mathrm{e}^{\omega_+\tau} \\ &\quad -i\frac{(\omega_n^4+4\xi^2\omega_n^2\omega_-^2)}{(\omega_--\omega_+)(\omega_--\bar{\omega}_-)(\omega_--\bar{\omega}_+)}\,\mathrm{e}^{\omega_-\tau} \end{aligned} \tag{C.5}$$

We are interested in the normalized auto-correlation function and thus proportionality factors are ignored. Inserting the roots from eq. (C.3) we get

$$\begin{aligned} R(\tau) &\propto \mathrm{e}^{-\xi\omega_n\tau}\left(\left[(1+4\xi^2)\sqrt{1-\xi^2}+i\xi(4\xi^2-1)\right]\mathrm{e}^{i\omega_n\sqrt{1-\xi^2}\tau}\right. \\ &\qquad \left.+\left[(1+4\xi^2)\sqrt{1-\xi^2}-i\xi(4\xi^2-1)\right]\mathrm{e}^{-i\omega_n\sqrt{1-\xi^2}\tau}\right) \\ &\propto \mathrm{e}^{-\xi\omega_n\tau}\left((1+4\xi^2)\sqrt{1-\xi^2}\cos(\omega_n\sqrt{1-\xi^2}\tau)+\xi(4\xi^2-1)\sin(\omega_n\sqrt{1-\xi^2}\tau)\right) \end{aligned} \tag{C.6}$$

For the standard value $\xi=\frac{1}{\sqrt{2}}$ the normalized auto-correlation function becomes:

$$\boxed{\frac{R(\tau)}{R(0)} = \frac{\mathrm{e}^{-\omega_n\tau/\sqrt{2}}}{3}\left(3\cos(\omega_n\tau/\sqrt{2})-\sin(\omega_n\tau/\sqrt{2})\right)} \tag{C.7}$$

D. List of Symbols

The following list contains frequently used symbols:

T_c:	Chip length
S:	Signal
$\tau(t)$:	Transmission delay of the signal
$\phi(t)$:	Phase of the signal
ω:	Carrier frequency
b_k:	State of a code or data bit ± 1
$g(t)$:	Chip shape
g:	Scattering factor for reflections off rough surfaces
T_p:	Integration interval of the correlator
N_c:	Number of bits in a code
t:	Time
d:	Correlator spacing. Defined as the delay between the early and the late correlator
$P(\omega)$:	Power spectrum
$\mathcal{F}$:	Fourier transformation
$\omega' := \omega + \Delta\omega$:	Frequency of the reference oscillator
$\phi' := \phi + \Delta\phi$:	Phase of the reference oscillator
$r(\cdot)$:	Integrand of the auto-correlation function
$\mathrm{Sqr}_n(t)$:	Square wave for BOC modulation, see appendix A
$\theta(\cdot)$:	Heaviside step function
$\delta(\cdot)$:	Dirac delta distribution, see appendix A
n:	Number of square wave periods in a BOC modulated chip
$t_c := T_c/n$:	Duration of a Square wave period
$\star$:	Convolution
B_{BW}:	Bandwidth
b:	Bandwidth parameter

P_{rec}:	Received power
P_{t}:	Transmitted power
λ:	Wave length
N_0:	Spectral power density of the thermal noise
$\hat{n}$:	Normal vector
$n(t)$:	White Gauß-ian noise process
$\xi(t)$:	Noise after integration
$\mathbf{p}$:	Differential operator
h_0:	Allan variance parameter, white frequency noise
h_{-1}:	Allan variance parameter, flicker frequency noise
h_{-2}:	Allan variance parameter, random walk frequency noise
χ_{ocx}:	Oscillator noise
$D(t)$:	Discriminator function of the tracking loops. Also called S-curve
δS_θ:	Variance of the carrier phase discriminator
δS_τ:	Variance of the code phase discriminator
ω_L:	Resonance frequency of the second order loop
ξ:	Critical damping of the second order loop
β:	Roll-off parameter, used for raised cosine signals
$\omega_{\text{max}\pm}$:	Maximum spectral density of BOC and SOC signals
$G(\mathbf{p})$:	Transfer function of the phase tracking loop
$F(\mathbf{p})$:	Transfer function of the code tracking loop
ω_{IF}:	Intermediate frequency
a_j:	Signal amplitude of multipath number j. a_0 corresponds to the direct line-of-sight.
$\eta_j := a_j/a_0$:	Relative signal amplitude between the direct line-of-sight and multipath number j.

E. List of Acronyms

The following list contains frequently used acronyms:

A/D:	Analog to digital conversion
BOC:	Binary offset carrier
BPSK:	Binary phase shift keying
C/A:	Coarse acquisition code
CDMA:	Code division multiple access
CPU:	Central Processing Unit
DLL:	Delay lock loop
DME:	Distance measuring equipment
DTM:	Digital terrain model
E,e:	Early channel
EGNOS:	European geostationary overlay system
ENU:	Local coordinate system with axis pointing in the directions East, North and up
GNSS:	Global Navigation Satellite System
GPS:	Global Positioning System
I:	In-phase component
IF:	Intermediate frequency
L,l:	Late channel
LNA:	Low noise amplifier
MEDLL:	Multipath estimating delay lock loop
MEO:	Medium earth orbit
MFC:	Microsoft foundation classes
MMIC:	Microwave mineaturized integrated circuit
NCO:	Numarical controlled Oscillator
NIMA:	National Imagery and Mapping Agency
NIST:	National institute of standards and technology
ODE:	Ordinary differential equation
P,p:	Punctual channel
PLL:	Phase locked loop
PRS:	Publicly regulated service
Q:	Quadra-phase component
RMS:	Root-mean-square
SDE:	Stochastic differential equation
SMR:	Signal-to-Multipath ratio
SNIR:	signal-to-noise-plus-interference ratio

SNSS:	Simulated Navigation Satellite System
SOC:	Sinusoidal offset carrier
STF:	Signal task force
STL:	Standard template library
TCAR:	Three carrier ambiguity resolution
TDMA:	Time division multiple access
TTFF:	Time to first fix
UTC:	Coordinated universal time
VCO:	Voltage controlled Oscillator
WAAS:	Wide area augmentation system
WGS-84:	World geodetic system
WRC:	World radio conference
XO:	Quarz oscillator

F. Acknowledgment

This thesis was written (mostly) during my years at the Institute for Geodesy and Navigation at the University of the Federal Armed Forces in Munich. Heartfelt thanks goes to the following (written in the corresponding languages)

An allererster Stelle möchte ich mich bei meiner Frau Ingrid bedanken. Danke für Dein Verständnis dafür, dass ich so viel von unserer gemeinsamen Zeit für diese Arbeit aufgewendet habe. Ohne diese Einstellung und Deine generelle Unterstützung, hätte ich es nie geschafft das hier fertig zu schreiben.

Mein Dank gilt Herrn Univ.-Prof. Dr.-Ing. Günter W. Hein, der diese Arbeit möglich gemacht hat, indem er mich in sein Institut aufgenommen hat.

Besonderer Dank geht an Univ.-Prof. Dr.-Ing. Bernd Eissfeller für die Betreuung der Arbeit, die auf Teile seiner Habilitation basiert.

Den Kolleginnen und Kollegen am Institut möchte ich auch ganz herzlich danken. Insbesondere dem "alten Kern" (Bernhard, Oliver, Jürgen, Robert, Wolfgang und Udo). Danke Euch für die vielen fachlichen (und nicht-fachlichen) Diskussionen, Ideen und Anregungen... und natürlich für den freizügigen Austausch von source-code.

Mamma, Amma og Palli: Þakka ykkur kærlega fyrir alla hjálpina og fyrir að reka á eftir mér og segja mér að drullast nú til að klára þetta!

Far, Takk for dine gode råd og at du også skælte mig ud med at jeg nu skulle se at få den afhandling skrevet færdig. Det havde du selfølgelig ret i!

Bibliography

[All66] D. Allan. Statistics of Atomic Frequency Standards. *Proceedings of the IEEE*, 54(2):221–230, 1966.

[Apo69] Tom M. Apostol. *Calculus*, volume II. John Wiley & sons, 1969.

[ARI93] ARINC Research Corporation, 2250 E. Imperial Highway, Suite 450, El Segundo, CA. *Navstar GPS Space Segment/Navigation User Interface*, revision c-004 edition, 1993.

[BBC+00] B. C. Barker, J. W. Betz, J. E. Clark, J. T. Correia, J. T. Gillis, S. Lazar, K. A. Rehborn, and J. R. Straton. Overview of the M-Code Signal. In *Proceedings of the Institute of Navigation, National Technical Meeting,* Anaheim, CA, USA, January 26-28 2000. Institute of Navigation.

[Bec94] D. Becker. Mehrwegeeinfluß bei GPS. *Zeitschrift für Satellitengestützte Positionierung, Navigation und Kommunikation*, 3(4):122–127, 1994.

[Bet99] J. W. Betz. The Offset Carrier Modulation for GPS Modernization. In *Proceedings of the National Technical Meeting*, pages 639–648, San Diego, California, January 25-27 1999. Institute of Navigation.

[Bet00a] J. W. Betz. Analysis of the M code Signal Interference with C/A code receivers. In *Proceedings of the Institute of Navigation, National Technical Meeting*, Anaheim, CA, USA, January 26-28 2000. Institute of Navigation.

[Bet00b] J. W. Betz. Design and Performance of Code Tracking for the GPS M-Code Signal. In *Proceedings of the Institute of Navigation, International Technical Meeting*, Salt Lake City, UT, USA, September 19-22 2000. Institute of Navigation.

[BF00] J. W. Betz and P. B. Fine. Effect of Narrow Correlator Spacing on Code Tracking Accuracy. In *Proceedings of the Institute of Navigation, International Technical Meeting*, Salt Lake City, UT, USA, September 19-22 2000. Institute of Navigation.

[BK00] J. W. Betz and K. R. Kolodziejski. Extended Theory of Early-Late Code Tracking for Bandlimited GPS Receiver. *Navigation: Journal of The Institute of Navigation*, 47(3):211–226, 2000.

[Blo92] S. Bloch. Aufbau und Funktion eines GPS Empfängers. In *DGON Seminar Satellitennavigationssysteme*, pages 31–55, Potsdam, 1992. Deutsche Gesellschaft für Ortung und Navigation (DGON).

[BS87] P. Beckmann and A. Spizzichino. *The Scattering of Electromagnetic Waves from Rough Surfaces*. Artech House Inc, Norwood, MA 02062, 1987.

[CPC+93] C. E. Cohen, B. S. Pervan, H. S. Cobb, D. G. Lawrence, J. D. Powell, and B. W. Parkinson. Real-Time Cycle Ambiguity Resolution Using a Pseudolite for Precision Landing of Aircraft with GPS. In *Proc. DSNS 93*, Amsterdam, The Netherlands, March 30 - April 2 1993.

[CPL+93] C. E. Cohen, B. S. Pervan, D. G. Lawrence, H. S. Cobb, J. D. Powell, and B. W. Parkinson. Real-Time Flight Test Evaluation of the GPS Marker Beacon Concept for Category III Kinematic GPS Precision Landings. In *Proceedings of the International Technical Meeting*, pages 841–849, Salt Lake City, Utah, September 22-24 1993.

[DFF92] A. J. Van Dierendonck, P. Fenton, and T. Ford. Theory and Performance of Narrow Correlating Spacing in a GPS Receiver. *Journal of the Institute of Navigation*, 39(3):265–283, 1992.

[Die92] A. J. Van Dierendonck. Theory and Performance of Narrow Correlator Spacing in a GPS Receiver. *Journal of the Institute of Navigation*, 39(3):265–283, 1992.

[Die94] A. J. Van Dierendonck. Understanding GPS Receiver Technonlgy: A Tutorial on What Those Words Mean. In *Proceedings of International Symposiom on Kinematic Systems in Geodesy, Geomatics and Navigation*, pages 15–24, Banff, Canada, August 1994.

[Die96] A. J. Van Dierendonck. GPS Receivers. In B. W. Parkinson and J. J. Spilker, editors, *Global Positioning System: Theory and Applications*, volume I of *Progress in Astronautics and Aeronautics*, pages 329–407. American Institute of Aeronautics and Astronautics, Inc., 1996.

[DMB84] A. J. Van Dierendonck, J. B. McGraw, and R. G. Brown. Relationsship between Allan Variances and Kalman Filter Parameters. In *Proceedings of the 16-th Annual Precise Time and Time Interval (PTTI) Systems and Applications Meeting*, pages 273–292, Greenbelt, MD, USA, December 1984.

[DPHR99] P. A. Dafesh, A. Parker, J. K. Holmes, and S. H. Raghavan. Simulation of the Code Tracking Performance of Square-Wave Modulated Ranging Codes. In *Proceedings of the Institute of Navigation International Technical Meeting*, pages 633–641, Salt Lake City, UT, USA, September 1999. Institute of Navigation.

[Eis97] B. Eissfeller. *Ein dynamisches Fehlermodell für GPS Autokorrelationsempfänger*. Habilitation, Schriftenreihe der Universität der Bundeswehr, Heft 55, Universität der Bundeswehr, München, 1997.

[FB00] P. M. Fishman and J. W. Betz. Predicting Performance of Direct Acquisition for the M-Code Signal . In *Proceedings of the Institute of Navigation, National Technical Meeting*, Anaheim, CA, USA, January 26-28 2000.

[FFF+91] P. Fenton, B. Falkenberg, T. Ford, N. Keith, and A. J. Van Dierendonck. NOVA-Tel's GPS Receiver, The High Performance GPS Sensor of the Future. In *Proceedings of the Institute of Navigation*, Albuquerque, NM, September 1991. Institute of Navigation.

[Gel74] Arthur Gelb. *Applied Optimal Estimation*. The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts 02142, 1974.

[GLV93] R. De Gaudenzi, M. Luise, and R. Viola. A Digital Chip Timing Recovery Loop for Band-Limited Direct-Sequence Spread-Spectrum Signals. *IEEE Transactions on Communications*, 41(11):1760–1770, November 1993.

[Gol80] H. Goldstein. *Classical Mechanics*. Addison-Wesley Publishing Company, 1980.

[HGI+02] G. W. Hein, J. Godet, J-L. Issler, J-C. Martin, R. Lucas-Rodriguez, and T. Pratt. Status of Galileo Frequency and Signal Design. In *Proceedings of the Institute of Navigation, International Technical Meeting*, Portland, Oregon, USA, September 24-27 2002.

[HRL98] J. K. Holmes, S. H. Raghavan, and S. Lazar. Acquisition and Tracking Performance of NRZ and Square Wave Modulation Symbols for Use in GPS. In *Proceedings of the Institute of Navigation, Annual Meeting*, Denver, CO, USA, June 1-3 1998. Institute of Navigation.

[ICA92] J. L. Issler, B. Cugny, and P. Agnieray. Multipath Effects on GPS Measurements During an Orbital Rendez-vous. In *Proceedings of the Institute of Navigation*, pages 841–851, Albuquerque, NM, September 1992. Institute of Navigation.

[IE02] M. Irsigler and B. Eissfeller. PLL Tracking Performance in the Presence of Oscillator Phase Noise. *GPS Solutions*, 5(4):45–57, 2002.

[Jac82] J. D. Jackson. *Classical Electrodynamics*. Addison-Wesley, second edition, 1982.

[Kas95] N. J. Kasdin. Discrete Simulation of Colored Noise and Stochastic Processes and $1/f^{\alpha}$ Power Law Noise Generation. *Proceedings of the IEEE*, 83(5):802–827, May 1995.

[KC73] J. D. Kraus and K. R. Carver. *Electromagnetics*. McGraw-Hill Electrical and Electronoc Engineering Series, Tokyo, 1973.

[KP95] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*, volume 23 of *Applications of Mathematics*. Springer-Verlag, Heidleberg, 1995.

[KPS97] P. E. Kloeden, E. Platen, and H. Schurz. *Numerical Solution of SDE Through Computer Experiments*. Univeritext. Springer-Verlag, Heidleberg, 1997.

[Lin72] W. C. Lindsey. *Synchronization Systems in Communication and Control*. Prentice Hall Electrical Engineering Series. Prentice Hall, 1972.

[LS73] W. C. Lindsey and M. K. Simon. *Telecommunication Systems Engineering*. Prentice Hall Electrical Engineering Series. Prentice Hall, 1973.

[Man71] B. B. Mandelbrot. A Fast Fractional Gaussian Noise Generator. *Water Resources Res.*, 7(3), June 1971.

[Mat96] P. Mattos. Multipath Elimination For The Low-Cost Consumer GPS. In *Proceedings of the Institute of Navigation, International Technical Meeting*, Kansas City, Missouri, USA, September 17-20 1996. Institute of Navigation.

[MB99] G. A. McGraw and M. S. Braasch. GNSS Multipath Mitigation Using Gated and High Resolution Correlator Concepts. In *Proceedings of the Institute of Navigation, National Technical Meeting*, pages 333–342, San Diego, CA, USA, January 25-27 1999. Institute of Navigation.

[Nee94] R. D. J. Van Nee. Multipath Effects on GPS Code Measurements. *Journal of the Institute of Navigation*, 39(2):177–190, 1994.

[PTVF97] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The art of scientific computing*. Cambridge University Press, Cambridge, New York, Port Chester, Melbourne Sydney, 2 edition, 1997.

[SD99] J. J. Spilker and A. J. Van Dierendonck. Proposed New Civil GPS Signal at 1176.45 MHz. In *Proceedings of the Institute of Navigation International Technical Meeting*, Salt Lake City, UT, USA, September 1999. Institute of Navigation.

[Sen91] J. Sennot. GPS Receiver Operation and Digital Signal Processing. In *Navtech Seminars*, 1991.

[Spi73] J. J. Spilker. *Digital Communications by Satellite*. Prentice Hall Electrical Engineering Series. Prentice Hall, 1973.

[Spi80] J. J. Spilker. GPS Signal Structure and Performance Characteristics. *Navigation*, 1:29–54, 1980.

[War94] Ph. W. Ward. Dual Use of Military Anti-Jam GPS Receiver Design Techniques for Commercial Aviation RF Interference Integrity Monitoring. *Zeitschrift für Satellitengestützte Positionierung, Navigation und Kommunikation*, 3(4):127–138, 1994.