University of Padova

Department of Information Engineering

Master Thesis in ICT for internet and multimedia - Telecommunications

A robust GNSS tracking enhancement for hostile environments

Supervisor
Prof. Oscar Pozzobon
University of Padova

Co-supervisor
Prof. Nicola Laurenti
University of Padova

Master Candidate
Pietro Targa

Student ID
2055516

Academic Year
2022-2023

Graduation date
October 12, 2023

Abstract

In urban environments any GNSS receiver is subjected to frequent sudden losses of the line-of-sight (LOS) signal and to the multipath phenomenon, which drastically reduce the accuracy, robustness and availability of the GNSS service. This thesis evaluates the implementation of a tracking loop for robust tracking of GNSS signals in hostile scenarios, developed in collaboration with Qascom and addressed to their software defined GNSS receiver, the QN400.
After an initial analysis of the state of the art in robustness enhancement techniques, it was decided, together with Qascom's Advanced Navigation team, to integrate a Kalman filter inside the common tracking loop structure. More specifically, the proposed tracking loop integrates a fourth-order Kalman filter and an outage detection algorithm into the standard structure, with the overall goal of improving tracking performance in terms of robustness to multipath effects and signal's interruptions. The proposed design was extensively tested with Qascom's semi-analytical simulator in Matlab, both with simulated scenarios, based on the DLR land mobile multipath channel model, and more realistic ones based on live recordings of a GNSS receiver mounted on a vehicle moving in Bassano del Grappa. The proposed solution has shown great efficacy in all designed test environments. Specifically, it has demonstrated superior resilience in resisting signal outages when compared to the standard tracking loop.

Contents

Listing of figures

Listing of tables

Listing of acronyms

ACF	Autocorrelation Function
ASI	Italian Space Agency
BOC	Binary Offset Carrier
CAN	Context Aware Navigation
CDMA	Code Division Multiple Access
DLL	Delay Locked Loop
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DOP	Dilution of Precision
DSSS	Direct Sequence Spread Spectrum
ESA	European Space Agency
FLL	Frequency Locked Loop
GEO	Geosynchronous equatorial orbit
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IF	Intermediate Frequency
IMU	Inertial Measurement Unit
LOS	Line Of Sight
LuGRE	Lunar GNSS Receiver Experiment
MEO	Medium Earth Orbit
NASA	National Aeronautics and Space Administration
NCO	Numerically Controlled Oscillator

NLOS	Non Line Of Sight
PLL	Phase Locked Loop
PRN	Pseudorandom Noise
PVT	Position Velocity and Time
RF	Radio Frequency
RHCP	Right Hand Circularly Polarized
SPS	Standard Positioning Service
VT	Vector Tracking

1 Introduction

This thesis was developed in collaboration with Qascom, an engineering company that offers security solutions in satellite navigation and space cybersecurity. One of its products, the QN400, is a software defined radio based receiver which provides an high degree of flexibility making it capable of different applications for space and Earth. For instance, in the context of space missions, the QN400 has been chosen for the Lunar GNSS Receiver Experiment (LuGRE) project in collaboration with ASI and NASA [14].

On Earth, the application scenario is completely different. In urban environments, such as city centres, any GNSS receiver is subjected to frequent sudden losses of the line-of-sight (LOS) signal and to the multipath phenomenon: the reception of multiple delayed and attenuated replicas of the original signal that interfere with each other at the receiver. These phenomena significantly impact the quality of service in terms of accuracy, robustness, and availability.

This thesis work specifically addresses the need to enhance the robustness and reliability of the QN400-Space GNSS receiver in urban scenarios. In particular the main goal was to make the receiver capable of resisting the frequent signal outages that occur in urban canyons, which cause the standard receiver to loose the lock of the tracking loops and thus to perform frequently the time-demanding re-acquisition process. In order to fulfill this need, a Kalman filter aided by an outage detection algorithm will be described and widely tested in this dissertation.

The next chapters of this document are organised as follows. Chapter 2 presents a general description of a GNSS architecture, Chapter 3 focuses more on the receiver, the signal structure and on standard acquisition and tracking procedures. Chapter 4 presents an overview of the state of the art techniques for robust tracking in hostile scenarios, Chapter 5 describes the proposed tracking loop, the Kalman filter structure and the outage detection algorithm. Finally, Chapter 6 reports the simulation performed and the results obtained, while Chapter 7 discusses the conclusions and future developments of this work.

2

Global Navigation Satellite System

2.1 Global positioning

Navigation can be defined as the strategy of getting something from one place to another; to do so a proper knowledge of the current position is needed and Global Navigation Satellite Systems (GNSS) were born to fulfill this need.
To get a simple idea of how the position computation is carried out, imagine a sailor with his ship lost in the middle of the sea [15]. Suppose that a lighthouse, in a fixed and known position and whose clock is perfectly synchronized with the mariner's one, emits a light pulse at a certain predetermined time instant. Now, by measuring the time delay in which he receives the pulse the mariner is able to calculate his distance from the lighthouse; this measure is defined as a pseudorange.

The mariner, having a single pseudorange and knowing the exact position of the lighthouse, has a circle of possible places where he could be.

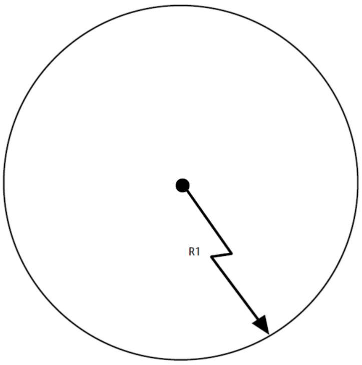


Figure 2.1: Single pseudorange measure. *Illustration by [1]*

In order to get a precise position in a 2D scenario three pseudoranges coming from three different lighthouses are required.

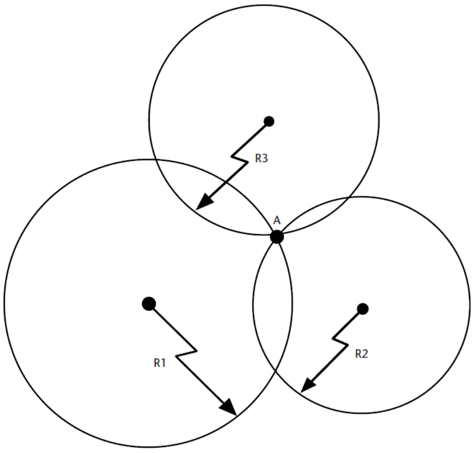


Figure 2.2: Ideal positioning exploiting three pseudoranges. *Illustration by [1]*

In a more realistic formulation, each calculated pseudorange will be affected by some uncertainty; the result of the position calculation will be an area (2D space) or a volume (3D space).

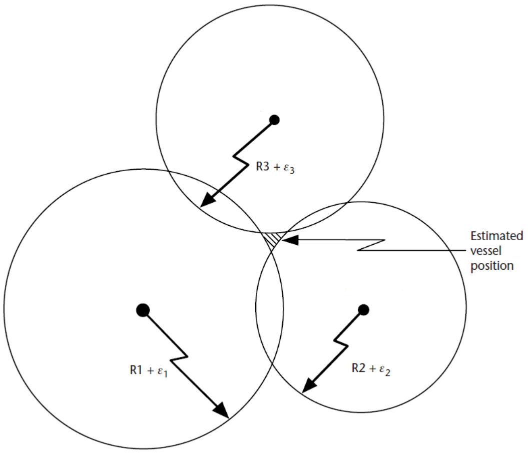


Figure 2.3: Uncertain positioning exploiting three pseudoranges. *Illustration by [1]*

The shape and size of this area depends not only on the number of pseudoranges but also on the position of the lighthouses; this effect is called Dilution of Precision (DOP). A bad distribution of the lighthouses results in a higher uncertainty space and consequently a higher DOP.

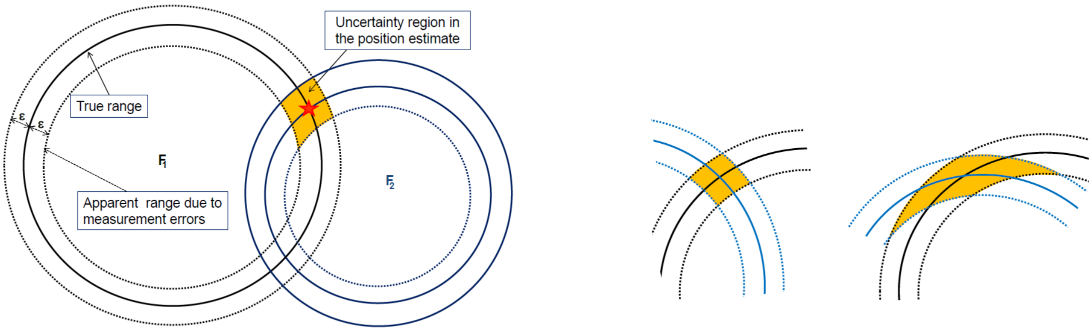

Uncertainty region in the position estimate
True range
ε
ε
F_1
F_2
Apparent range due to measurement errors

Figure 2.4: Dilution of Precision, on the right an example of low and high DOP. *Illustration by [2]*

From this simple example the generalisation to the more complex GNSS 3D scenario is straightforward.
The previously mentioned lighthouses represent the GNSS satellites and pseudoranges are calculated based on the time taken for the signal transmitted by each satellite to reach the receiver. The assumption that the exact satellite position is known to the receiver is satisfied because the signal sent by each satellite encodes the ephemeris data: a set of parameters that define

the satellite's position and orbit. On the other hand, the assumption that the receiver is perfectly synchronised with the satellite becomes too strong; the receivers are usually equipped with cheap quartz clocks which usually have poor stability. For this reason, a time estimate appears in the system of equations to be solved and thus an additional pseudorange measure is required. Furthermore, in order to improve the DOP and provide a more accurate position, multiple pseudoranges are included in the positioning and time estimation [2].

2.2 GNSS architecture

A Global Navigation Satellite System is a satellite-based system that provides continuous positioning over the globe. Nowadays several GNSSs are operational: the well-known GPS designed by the US, the European Galileo, the Russian Glonass, and China's Beidou.

NAVSTAR GPS
GALILEO
ГЛОБАЛЬНАЯ НАВИГАЦИОННАЯ СПУТНИКОВАЯ СИСТЕМА
ГЛОНАСС
北斗卫星导航系统
BeiDou Navigation Satellite System

Figure 2.5: GNSS logos (GPS, Galileo, Glonass and Beidou).

Each of them shares a common architecture structure which consists of three main segments: a space segment, a control segment and the user segment.

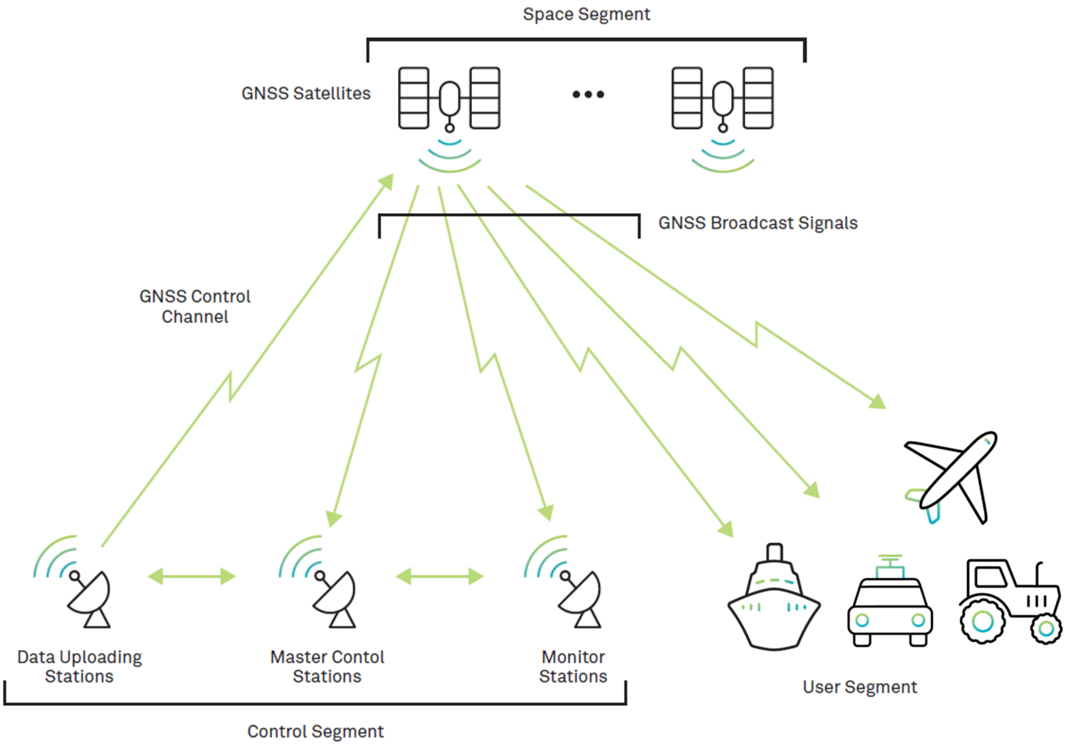


Figure 2.6: GNSS segments. *Illustration by [3]*

2.2.1 Space Segment

The orbital satellites of a GNSS constitute the space segment. Typically positioned in Medium Earth Orbits (MEOs), these satellites are organized into constellations designed to ensure that a user will have at least four satellites in view from any point on Earth.
Figure 2.7 displays how different GNSSs constellations differ in terms of number of satellites, orbits altitudes and inclinations.

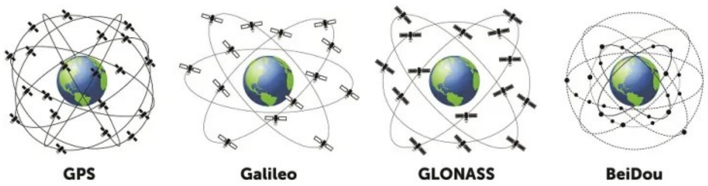


Figure 2.7: GNSS constellations. *Illustration by [4]*

2.2.2 Control Segment

The control segment includes all ground infrastructures responsible for monitoring and maintaining general health and status of the satellites, including their position, solar arrays status, battery level, propellant levels for the orbit adjusting manoeuvres and many more tasks. Furthermore, the control segment defines the ephemeris and almanac data contained in the navigation message and updates each satellite's clock. The ephemeris parameters define precisely each satellite orbit and are esential for user receivers to predict each satellite position. The almanac is a condensed version of these parameters, featuring only 7 of the original 15. These values along with the clock corrections are modified and adjusted each few hours.

2.2.3 User Segment

The user segment consists of all the devices equipped with a GNSS receiver which acquires, tracks and demodulates the signals coming from satellites to obtain pseudoranges and compute the device's position.
These receivers are embedded in many of our everyday devices such as smartphones, smartwatches, cars and many others.

3

Standard GNSS Receiver

3.1 Receiver structure

The GNSS receiver is responsible for acquiring, tracking and demodulating the signals received form the space segment. It then collects the data message, estimates the pseudoranges and computes the position and time.
The general structure of a GNSS receiver is shown in Figure 3.1.

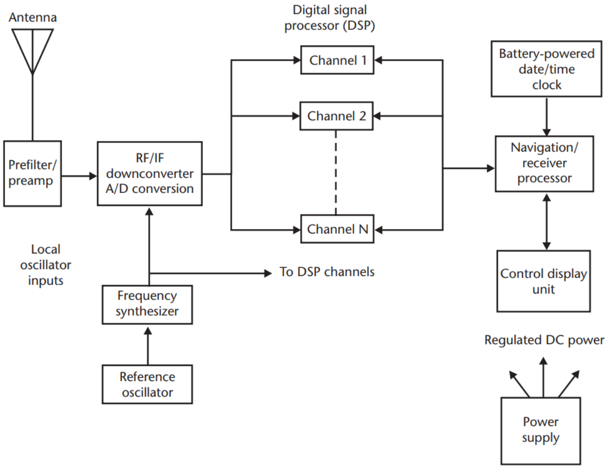


Figure 3.1: Generic GNSS receiver. *Illustration by [1]*

The incoming RF signal, which structure will be described in detail in Section 3.2, is captured by an antenna, filtered, amplified, downconverted and transferred from the analogue to the digital domain. At this point, exploiting the CDMA technique used in the downlink transmission from the satellites to the ground, each satellite's signal is acquired and goes through tracking process while its navigation data is demodulated to obtain the pseudoranges estimations and compute timing and positioning.
For the purposes of this thesis, the focus is moved to the signal structure and to the acquisition and tracking processes performed in each channel belonging to the digital signal processor block showed in Figure 3.1.
For a more detailed description of the receiver please refer to [1], [7] or [16].

3.2 Signal structure

All GNSS satellites transmit their signals in the L-band (from 1 to 2 GHz); the frequency allocation between the different positioning services is shown in Figure 3.2.

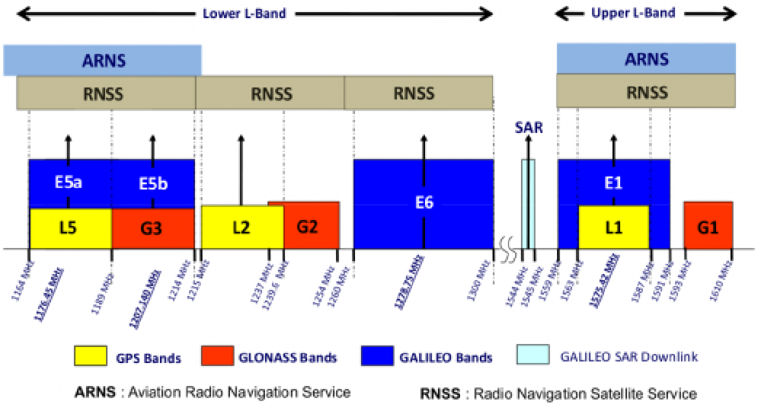


Figure 3.2: Frequency allocation in the L band. *Illustration by [5]*

It's clear that many satellites, both belonging to the same service or not,share the same medium, thus an appropriate Multiple Access technique is required.
Most GNSSs use Direct Sequence Spread Spectrum (DSSS), which adds to the modulation process, BPSK in the case of GPS, the use of a spreading waveform generated from a PRN code, a finite binary sequence completely known to the intended receiver.

A generic signal will then be a combination of the following components:

- **RF carrier**
  A sinusoidal signal at a given frequency.
- **Data waveform**
  A binary encoded message containing the ephemeris and almanac, the clock bias, the satellite status and other complementary information.
- **Spreading waveform**
  A spreading signal generated by the transmitter according to a pseudorandom unique chip sequence: the PRN code. This allows a receiver to acquire and track a given satellite's signal by knowing its PRN code.

The graphical representation of this combination is shown in Figure 3.3. *Tc*, the time interval associated to a PRN code chip and is commonly referred as a chip period, its reciprocal is the chipping rate denoted as *Rc* and the portion of spreading waveform in a chip period is a spreading symbol.
The choice of the spreading symbol differentiates one implementation from the others, in the case of GPS the choice made was pretty simple: a constant amplitude over a chip period. In this case, the complete DSSS process can be referred as BPSK-R [1]. In GALILEO's case a more complex MBOC modulation is utilized, a combination of two different square wave portions (BOC signals)[17] [18].

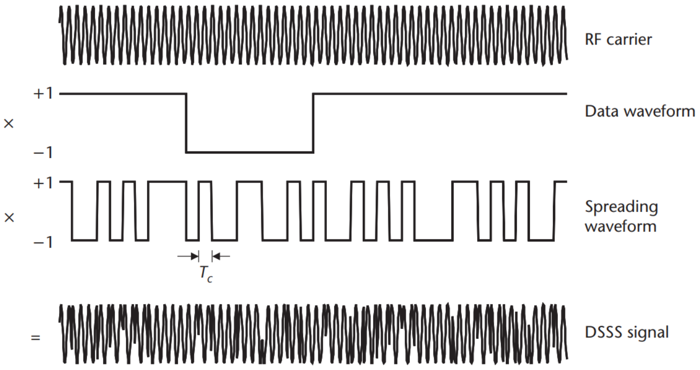


Figure 3.3: DSSS signal generation example. *Illustration by [1]*

The use of DSSS has three main objectives:

- the introduction of frequent phase changes introduced by the PRN waveform enable precise ranging by the receiver thanks to a sharper Autocorrelation Function (ACF) on the peak.
- the use of different PRN codes for different satellites enables the simultaneous transmission at the same frequency.
- it provides significant rejection of narrowband interference.

An important aspect of these DSSS signals is their correlation properties. For simplicity let's consider a baseband signal $r(t)$ with no data encoded and modulated with a BPSK. Its autocorrelation is the result of the correlation between the signal $r(t)$ and a time-shifted replica $r(t-\tau)$ and can be computed as:

$$R(\tau) = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} r^*(t)r(t+\tau)d\tau \tag{3.1}$$

In the case of a perfectly random binary code Equation 3.1 leads to:

$$R(\tau) = \begin{cases} A^2(1-\frac{|\tau|}{T_c}) & \textit{for } |\tau| \leq T_c \\ 0 & \textit{elsewhere} \end{cases} \tag{3.2}$$

The figure below shows a graphical representation of $R(\tau)$.

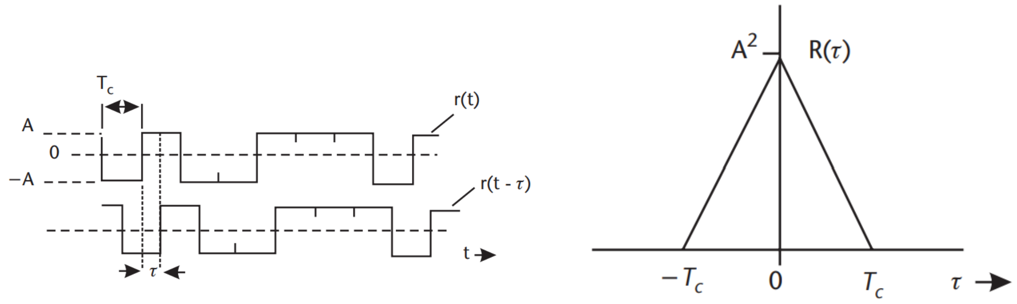


Figure 3.4: Autocorrelation between two copies of r(t). *Illustration by [1]*

Figure 3.5 displays how different modulations result in different ACFs, the higher slope around the main peak provided by the MBOC modulation improves the tracking performance in respect to the simpler BPSK-R [19].

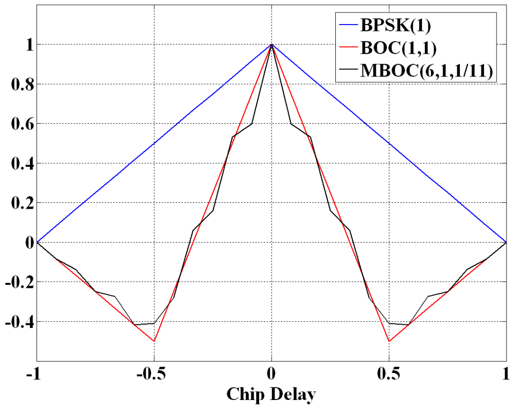


Figure 3.5: Autocorrelation functions of a GPS signal, a simple BOC modulated signal and a GALILEO signal exploiting a MBOC modulation. *Illustration by [5]*

The shape of $R(\tau)$ is the key to the acquisition and tracking process performed by the receiver; by maximising the correlation between a local replica of the PRN waveform and the incoming signal, it can estimate the delay, phase and frequency offsets required to demodulate correctly the navigation data and to compute the pseudoranges.
Finally, the orthogonality between signals transmitted at the same frequency is ensured, since the cross-correlation between two different random sequences of sufficient length is close to zero. This is necessary to share the common medium without inter-signal interference.

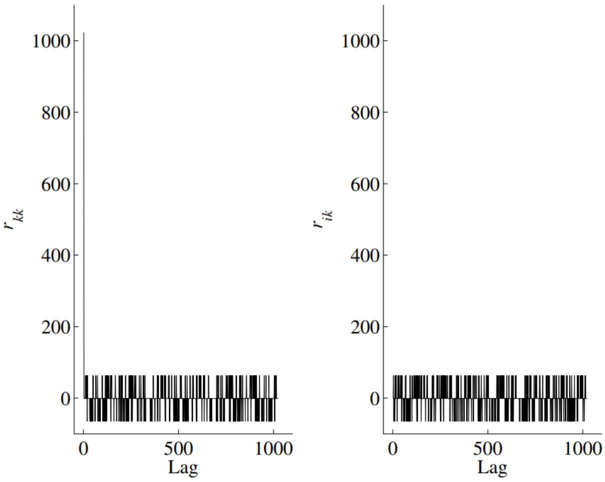


Figure 3.6: Autocorrelation of a 1023 chip long PRN sequence and cross correlation between two different PRN sequences. *Illustration by [6]*

3.3 Acquisition

The purpose of the acquisition process is to determine approximate values of the code delay and carrier frequency of a signal emitted by a specified satellite exploiting its unique and known PRN sequence. The code delay estimate is required to precisely align in time the local replica of the PRN waveform, so that it can be properly removed from the acquired signal. Meanwhile the frequency of the local replica generated by the receiver's oscillator has to match the actual frequency of the incoming signal to downconvert it correctly. In fact, even if the transmitting frequency of a given satellite is precisely known, it will deviate from its original value because of the Doppler effect caused by the mutual velocity between the receiver and the satellite, which shifts the signal's frequency of values up to $\pm 10kHz$.

Recalling the correlation properties introduced in Section 3.2, the acquisition process is a two-variable maximization problem; the correct frequency and code delay values will be those that maximize the correlation between the incoming signal and the local replica. This optimisation problem can be performed by computing the correlation several times, adjusting the two parameters (serial search algorithm) or by using more advanced techniques such as the parallel frequency space search acquisition or the parallel code delay search acquisition [16].

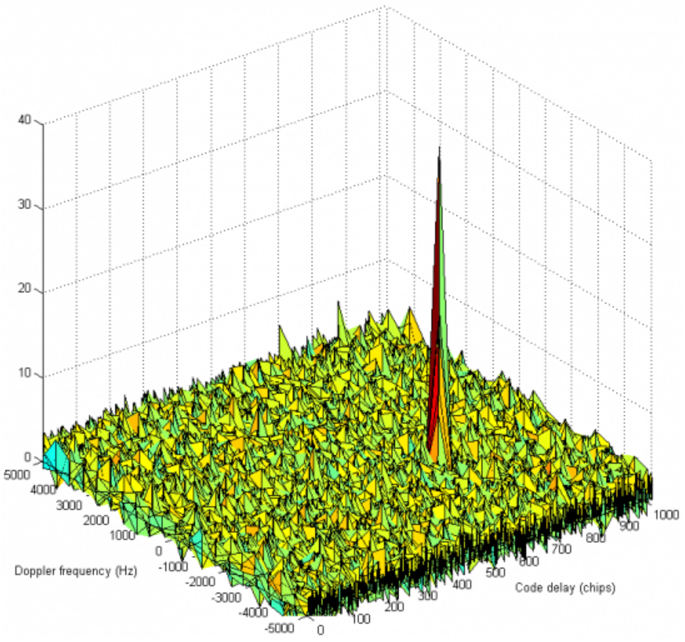


Figure 3.7: Example of acquisition process result where the desired satellite is found to be present. *Illustration by [5]*

3.4 Tracking

The coarse code phase and frequency estimates obtained from the acquisition process must be refined and constantly updated in order to keep up with the changes in the receiver and satellite dynamics. This ensures a correct removal of the PRN sequence, an accurate computation of the pseudorange and precise decoding of the navigation message; this is done by the tracking loop. The structure of a single channel signal processing block responsible of the tracking part is displayed in Figure 3.8. The upper loop is denoted as the code tracking loop or delay locked loop (DLL), while the lower one is the carrier tracking loop which can be ether a phase locked loop (PLL) or a frequency locked loop (FLL).

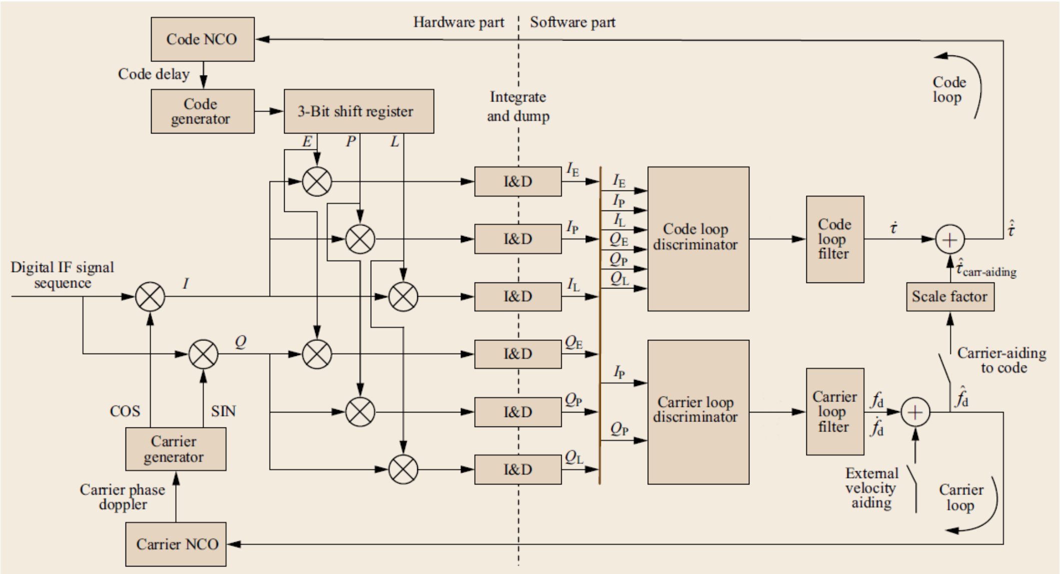


Figure 3.8: Standard GNSS receiver's tracking engine. *Illustration by* [7]

3.4.1 Correlators

The received signal is filtered, amplified and sampled before being separated into its in-phase and quadrature components, I and Q. These two components, obtained exploiting the carrier generated by the receiver at the current estimated carrier frequency, are then multiplied by three distinct replicas of the PRN code: a prompt one with a delay equal to the current estimate, a slightly late one and a slightly early one. These six signals are then integrated by the integrate and dump blocks (I&D) over an integration interval defined as T_{int} typically ranging from $1ms$

to $100ms$. This process is performed by the correlators and results in six signals: I_E, I_P, I_L, Q_E, Q_P and Q_L.

3.4.2 Discriminators

The purpose of the discriminators is to sense and extract, from the correlator outputs, the current amount of error in the local replica frequency and code phase in respect to the incoming signal. Different types of discriminators define different tracking loops.

Carrier loop discriminator

The carrier tracking loop can be either a carrier-phase tracking loop, if the value being tracked is the carrier's phase, or a carrier-Doppler tracking loop, where instead is the carrier's frequency being tracked.
Carrier-phase tracking loops are mainly of two types, pure PLL and Costas PLL. The latter is insensitive to the presence of the phase inversions due to the data bit modulation while the first one has to care about bit inversions caused by the encoded data message. In both cases the inputs are the I_p and Q_p components computed by the correlators and the output is the carrier phase error which will be used as input to the PLL loop filter to reduce noise.
Table 3.1 and Table 3.2 present the most common PLL discriminators alongside with their main characteristics.

Table 3.1: Common pure PLL discriminators.

Discriminator	*Characteristics*
$atan2(Q_p, I_p)$	Computes the four-quadrant arctangent, corresponds to the maximum likelihood estimator at both high and low SNR. Computationally hard.
$\frac{Q_p}{average(\sqrt{I_p^2 + Q_p^2})}$	Q_p normalized by the average prompt envelope, normalization provides insensitivity at high and low SNR. Low computational burden.

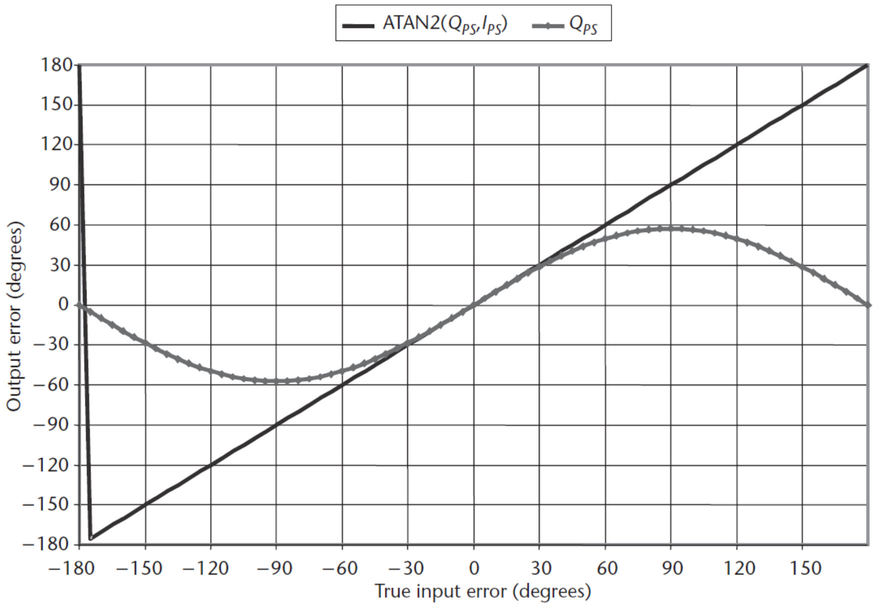


Figure 3.9: Pure PLL discriminators outputs. *Illustration by [1]*

Table 3.2: Common Costas PLL discriminators.

Discriminator	*Characteristics*
$Q_p \times I_p$	Classic Costas analog discriminator, nearly optimal at low SNR, moderate computational needs.
$Q_p \times sign(I_p)$	Decision directed Costas discriminator, nearly optimal at high SNR, least computational burden.
$\frac{Q_p}{I_p}$	Suboptimal but good at both high and low SNR, higher computational burden.
$atan\left(\frac{Q_p}{I_p}\right)$	Computes the two-quadrant arctangent, maximum likelihood estimator solution at high and low SNR but high computational burden.

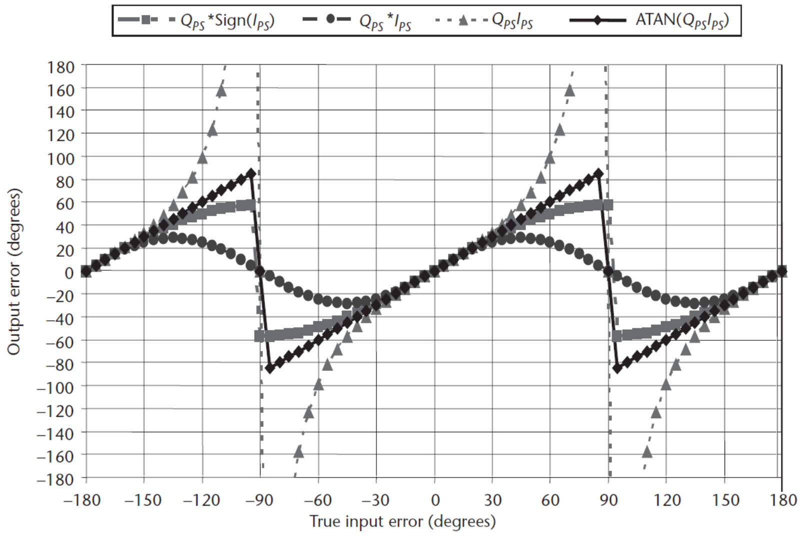


Figure 3.10: Costas PLL discriminators outputs. *Illustration by* [1]

On the other hand the carrier Doppler tracking loop, defined as FLL, produces the carrier Doppler frequency error, which is fed into the FLL loop filter for the same reason described above.

Table 3.3: Common FLL discriminators.

Discriminator	*Characteristics*
$\frac{cross}{\Delta t}$	Classic Costas analog discriminator, nearly optimal at low SNR, moderate computational needs.
$\frac{cross \times sign(dot)}{\Delta t}$	Decision directed Costas discriminator, nearly optimal at high SNR, least computational burden.
$\frac{atan2(dot, cross)}{\Delta t}$	Suboptimal but good at both high and low SNR, higher computational burden.

where: $cross = I_p(t_1) \times Q_p(t_2) - I_p(t_2) \times Q_p(t_1)$, $\quad \Delta t = t_2 - t_1$
and $\quad dot = I_p(t_1) \times I_p(t_2) + Q_p(t_1)$

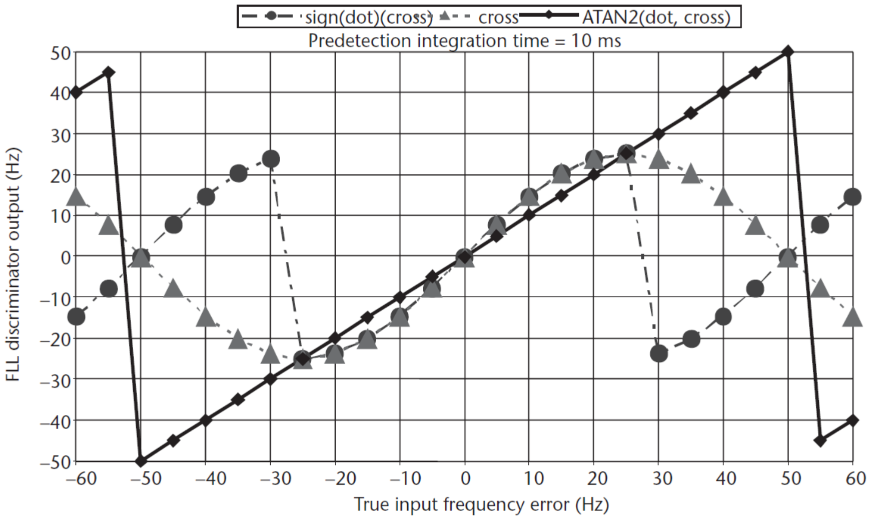


Figure 3.11: FLL discriminators outputs. *Illustration by [1]*

Overall, these three methods aim at achieving the same goal of local carrier adjustment. Their performances, however, are slightly different. The main trade-off lies between robustness to dynamics and accuracy. In general a PLL produces more accurate estimates whereas the FLL is more robust to high dynamics.

The use of a FLL or PLL in the carrier tracking loop is not always exclusive; there exist more complex configurations that utilize both. An example is the FLL assisted PLL architecture [1], which is employed in Qascom's semi-analytic simulator.

Code loop discriminator

The fundamental concept is similar to that of the PLL discriminator; the main difference lies in the fact that the latter is based on the derivatives of the maximum likelihood estimation cost functions, whereas the cost function of the code delay, a triangular shaped correlation function, is not differentiable at the apex, precisely when the code delay error is zero. Therefore the DLL discriminators exploit the use of the early and late correlators to obtain an approximation of the derivative of the cost function.
Figure 3.12 shows some examples of correlators outputs for four different code delay errors. If the replica code is perfectly aligned, the early and late correlators are equal in amplitude and no error correction will be generated by the discriminator. Conversely, if the code local replica is misaligned, the early and late envelopes will differ in amplitude and the discriminator will sense the amount of error and compute a correction.

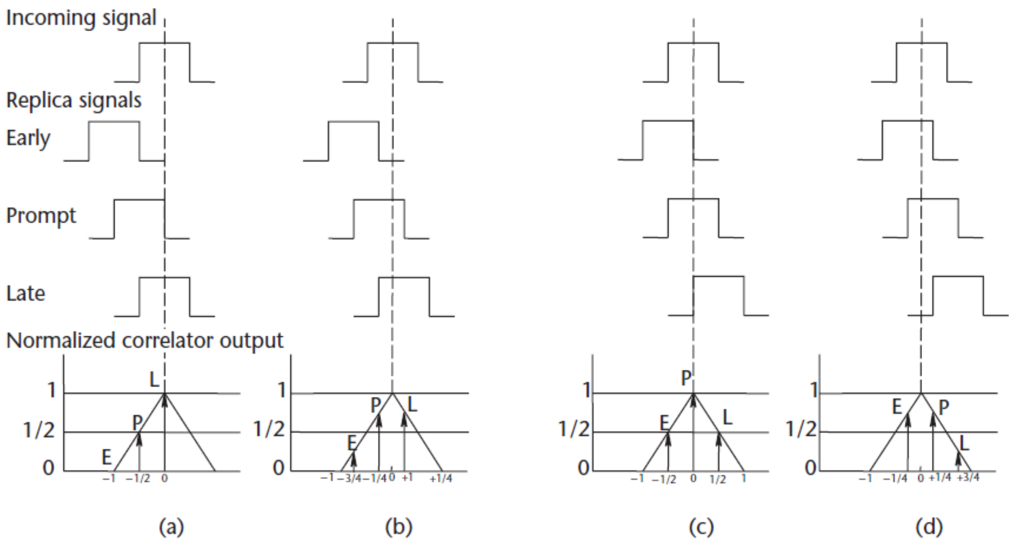


Figure 3.12: Code correlator outputs for different code delay errors. *Illustration by [1]*

The most common types of DLL discriminators are presented alongside with their characteristics in Table 3.4.

Table 3.4: Common DLL discriminators

Discriminator	*Characteristics*
$\frac{1}{2}\frac{E-L}{E+L}$	Normalized early minus late, high computational load.
$\frac{1}{2}(E^2-L^2)$	Noncoherent early minus late power, moderate computational load.
$\frac{1}{2}[(I_E-I_L)I_P+(Q_E-Q_L)Q_P]$	Quasi-coherent dot product power, exploits all three correlators, computationally simple.
$\frac{1}{2}[(I_E-I_L)I_P]$	Coherent dot product, it's the most accurate code discriminator, with a low computational load, but needs the carrier loop to be phase locked.

where: $E=\sqrt{I_E^2+Q_E^2}$ and $L=\sqrt{I_L^2+Q_L^2}$

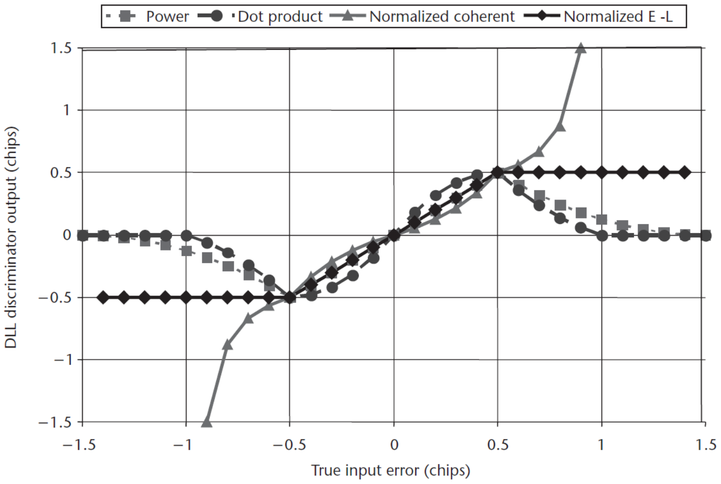


Figure 3.13: DLL discriminators outputs. *Illustration by [1]*

3.4.3 Loop Filter

Both the delay and carrier discriminator outputs are affected by a significant noise component. As a solution a filter is introduced in the loop to mitigate this noise element and generate a more accurate estimate of the code and carrier errors.
The general behaviour of a loop filter is mainly determined by two parameters: the loop filter order and the noise bandwidth. A loop filter can be of first, second or third order and, depending on this, it will be more sensitive to velocity, acceleration or jerk respectively.
The noise bandwidth controls the accuracy of the error estimates as well as the robustness to the dynamics. Higher bandwidths result in higher robustness to dynamics but lower accuracy, lower bandwidths produce more accurate estimates but it will be more affected by dynamics.

3.4.4 NCO and replica generator

The numerically controlled oscillator (NCO) paired with the carrier and code generators produce a local replica of the PRN code and of the carrier wave according to the current estimates computed by the loop filter. These replicas will be the ones used in the next integration interval by the correlators.

Having described the main components of a tracking loop, a condensed view of a tracking loop is displayed in Figure 3.14.

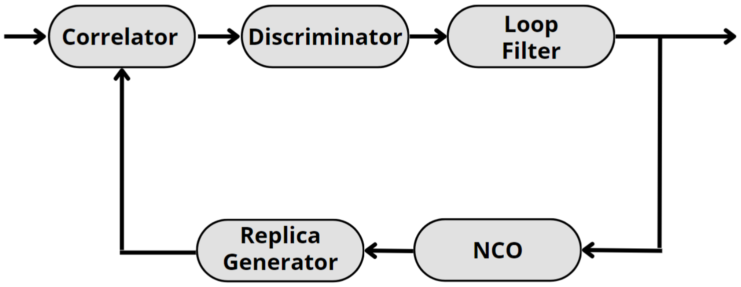


Figure 3.14: Tracking loop structure.

4

GNSS signal degradations and possible solutions

Any GNSS signal encounters many factors that contribute to its degradation on its journey from transmitter to receiver. Firstly, the electromagnetic wave as it travels through the atmosphere is effected by radio frequency interference, whether caused by someone deliberately threatening the service's reliability, e.g. spoofing [20], or caused by other transmission occurring at nearby frequencies [21]. In addition, the ionosphere introduces frequency-dependent delays in the signals, as well as fading and scattering phenomena commonly referred to as ionospheric scintillations which can result in loss of signal power and the subsequent loss of lock on tracking loops [22].

Things get even more complicated if the receiver is placed in an urban scenario where buildings as long as trees, lampposts and other urban infrastructures strongly affect the signal integrity[23] [21]. As displays Figure 4.1, only a few LOS signals will be available to the receiver; many will be completely blocked and many delayed and attenuated copies of the original signal will be captured by the receiver due to multipath effect.

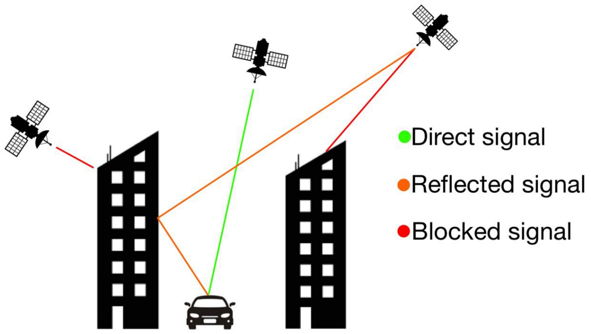


Figure 4.1: Urban canyon.

Figure 4.2 displays the effect of multipath in the received signal's ACF. The distortion observed depends not only on the attenuation and delay of the signal's echo, but also on weather the two signals interfere instructively or desctructively. In both cases the distortion causes a general loss of accuracy in the pseudoranges estimates and robustness in the code and carrier phase tracking.

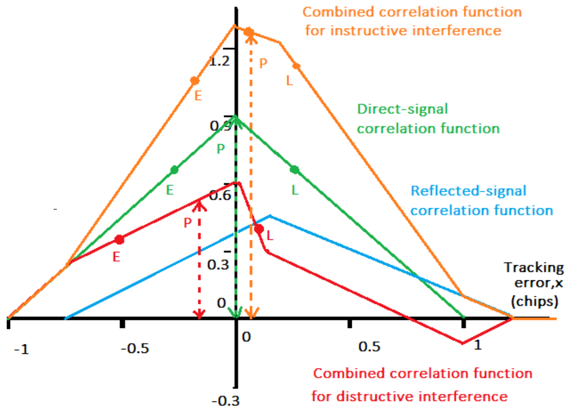


Figure 4.2: Effect of multipath on the signal's ACF. *Illustration by [8]*

4.1 Mitigation solutions

Over the years, many strategies have been developed to mitigate the effects of signal degradation in urban scenarios. These may involve the structure of the receiver, but also the design of the antenna and external sensors belonging to the device on which the receiver is mounted. This chapter presents some of the most interesting state-of-the-art solutions.

Antennas designs

Particular antenna designs can have a key role in multipath effects mitigation effects; a proper antenna pattern could reduce the NLOS components which usually come to the receiver from low elevation angles. Article [24] analyses the performance of a choke ring antenna, but in recent years more complex designs have been developed with the general aim of reducing antenna dimensions. Authors of [25], [26] and [27] considered different antenna arrays, whereas in [28] and [29] performance improvements are observed by designing different ground planes. Another approach is to exploit the change in polarisation that occurs upon reflection of a signal. In particular the GNSS Right Hand Circular Polarized (RHCP) signals, after a single reflection, becomes Left Hand Circular Polarized (LHCP). This difference in polarization has been shown to be a good feature to exploit [30].
Lastly interesting results are presented in [31], where the use of an antenna array is combined with an independent component analysis algorithm which filters noise and reduces multipath interference.

Context Aware Navigation

A common simple approach is Context Aware Navigation. The basic idea is simple: define different working environment categories and properly adjust the receiver parameters to maximize its performance in each scenario. Different metrics and features can be exploited to define and detect the channel status, and different parameters can be adjusted on the receiver. Examples of this approach can be found in [32] and [9] where standard, degradated and highly degradated working environments are defined taking into account the number of satellites in view, their evolution over time and the DOP of the calculated position by the receiver. The most common parameters adjusted by the receiver are the loop filters bandwidths and the Carrier-to-Noise Ratio mask which defines the minimum $C/N0$ of a signal required to be included in the Positioning Navigation and Timing (PNT) estimations.
Figure 4.3, taken from [9], presents an example where the detection of the environment condition is based on a change factor linked to the evolution of the satellite in view.

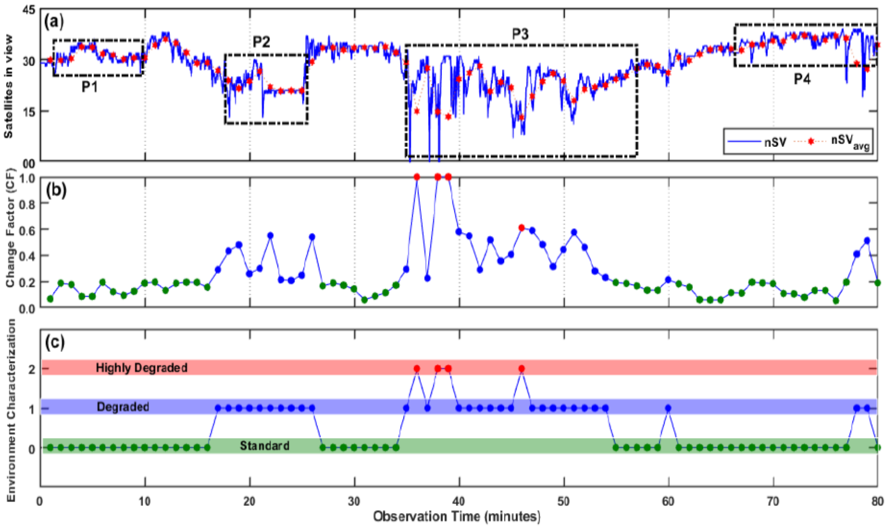


Figure 4.3: Example of CAN algorithm output. *Illustration by [9]*

Tracking loop enhancements

Other strategies that focus on mitigating the effects of multipath act directly on the tracking loop, for example by defining new discriminators. Article [33] suggests a general methodology for designing a multi-correlator discriminator for the code tracking loop, based on a heuristic minimisation of a cost function, the mean multipath error. The same authors in [34] propose a code discriminator based on the steepest descent algorithm.

Other approaches are required to improve the robustness of the receiver against signal power losses, a common solution is the use of Kalman Filters. For instance, [35] and [22]describe the application of such filter to mitigate ionospheric scintillations, but a similar approach proved to be effective in urban scenarios where a signal's $C/N0$ often drops due to the presence of trees, lampposts and buildings that interrupt the LOS signal [36]. An advanced approach involves the use of extended Kalman Filters and vector tracking (VT): instead of using separate tracking loops for each channel, VT processes all channels at the same time with and Extended Kalman Filter for both tracking and PNT estimations [37] [38].

External aidings

Another interesting approach involves the use of external aids, such as the vision-aided receiver presented in [10] which employs an omnidirectional camera to capture images of the sky above the receiver. After some processing the satellites that are supposed to be in view are projected on these images and, by recognizing which satellites are behind buildings, only the LOS signals are retained.

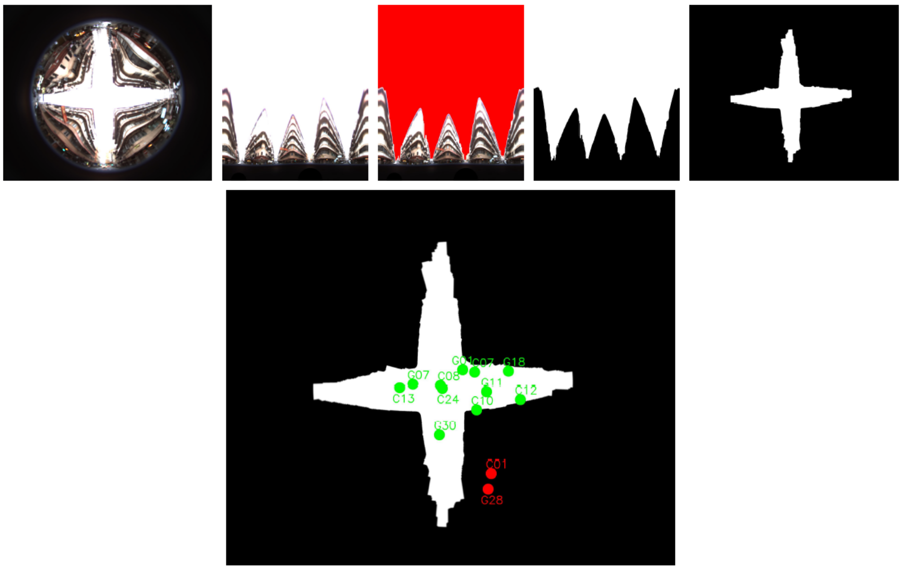


Figure 4.4: The image processing procedure and below the resulting image with projected satellites. *Illustration by [10]*

Machine learning

Finally, there has been a recent shift towards utilizing Machine Learning methods. Examples of Multipath detection algorithms can be found in the literature, including the use of convolution neural networks [39] [11], support vector machines [40], and gated recurrent units [41]. Figure 4.5 displays the structure of convolutional neural network that uses as input a representation of the IQ samples as a 2D image to detect heavy multipath components.

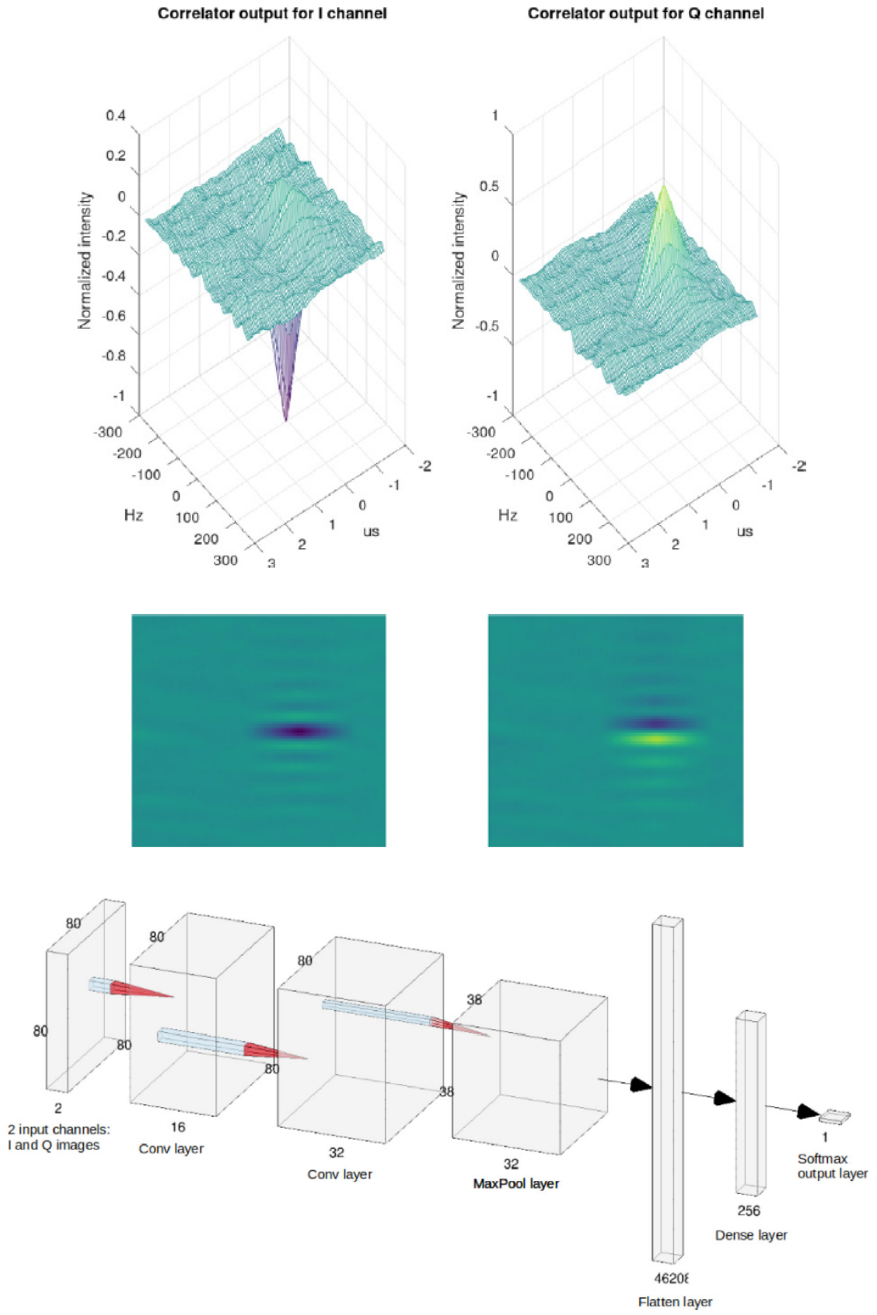


Figure 4.5: 2D image representation of IQ samples and below the convolutional neurale network proposed. *Illustration by [11]*

5
Proposed Tracking Loop Design

Taking into account all possible approaches to mitigate signal degradation in urban scenarios outlined in Section 4.1 and comparing them with the interests and needs of Qascom, it was decided to implement a Kalman filter within the tracking loop. This decision was based on the following reasons:

- The algorithm's scope was decided to be restricted to the tracking loop, with no involvement of the PVT computations or the antenna design, as it is independent of the receiver.
- For sake of simplicity no external aiding methods were considered, such as vision-aiding or inertial measurement units (IMU).
- The current version of Qascom's software defined GNSS receiver, the QN400, proved to be quite unstable when subjected to frequent signal outages rather then to strong multipath components. This forces the receiver to perform the acquisition procedure every time the LOS signal gets interrupted.
- The algorithm's computational demands had to be limited due to limited resources for real-time computations, consequently the more complex VT had to be excluded.

5.1 Overview

An high-level summary of the proposed tracking loop is illustrated in Figure 5.1. In respect to the standard design presented in Figure 3.14, an extra block, placed between the discriminator and the loop filter, have been added which consists in a Kalman filter assisted by an outage detection algorithm.
In the following sections the inner structure and working procedure of these two is presented.

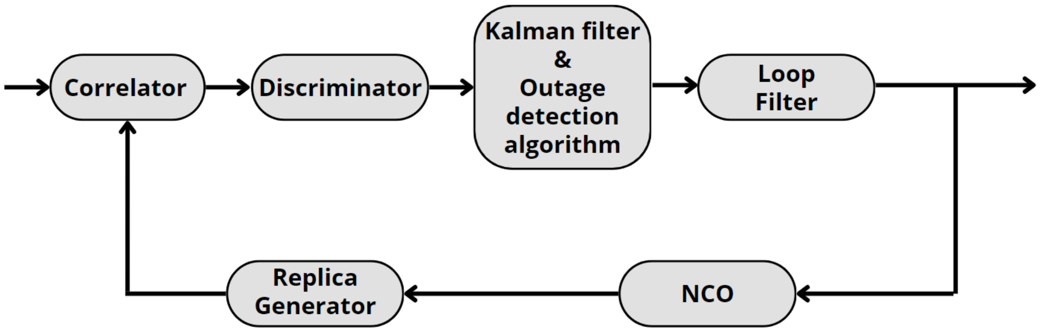


Figure 5.1: Proposed tracking loop

5.2 Kalman Filter

The Kalman filter, first introduced by Rudolph E. Kalman in [42], is widely considered as the optimal solution for tracking and data prediction purposes [43].

The tracked data by the filter is stored in the filter state vector denoted as $\hat{\boldsymbol{x}}_{n,n}$, its dimension is the Kalman filter's order. The working cycle, displayed in Figure 5.2, consists in two phases: "predict" and "correct", based on five main equations.
In the prediction phase performed at time n the expected state for time $n+1$ is computed through the state prediction equation:

$$\hat{\boldsymbol{x}}_{n+1,n} = \boldsymbol{F}\hat{\boldsymbol{x}}_{n,n} + \boldsymbol{G}\boldsymbol{u}_n + \boldsymbol{w}_n \tag{5.1}$$

Where $\boldsymbol{F}$ represents the state transition matrix which projects the current state into a prediction for the future iteration, $\boldsymbol{G}$ is the control matrix which monitors an addition input variable to the Kalman filter $\boldsymbol{u}_n$ and $\boldsymbol{w}_n$ is the unknown process noise.

Alongside the state also its uncertainty $\boldsymbol{P}$ is predicted as a combination of the current state uncertainty and $\boldsymbol{Q}$, the process noise covariance of the dynamic model being tracked.

$$\boldsymbol{P}_{n+1,n} = \boldsymbol{F}\boldsymbol{P}_{n,n}\boldsymbol{F}^T + \boldsymbol{Q} \tag{5.2}$$

During the correction phase, the state update equation evaluates the current state; the previously predicted state $\hat{\boldsymbol{x}}_{n,n-1}$ gets adjusted by a factor which depends on the current measure of the tracked values $\boldsymbol{z}_n$:

$$\hat{\boldsymbol{x}}_{n,n} = \hat{\boldsymbol{x}}_{n,n-1} + \boldsymbol{K}_n(\boldsymbol{z}_n - \boldsymbol{H}\hat{\boldsymbol{x}}_{n,n-1}) \tag{5.3}$$

The observation matrix $\boldsymbol{H}$ extracts the values of interest from the state $\hat{\boldsymbol{x}}_{n,n-1}$ while $\boldsymbol{K}_n$ referrs to the Kalman gain. The term in brackets in Equation 5.3 is commonly referred to as innovation and denoted as $\partial\boldsymbol{z}$. The gain value controls how much the current measurement affects the current state $\hat{\boldsymbol{x}}_{n,n}$ in respect to the prediction. This value is computed at each iteration as:

$$\boldsymbol{K}_n = \boldsymbol{P}_{n,n-1}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}_{n,n-1}\boldsymbol{H}^T + \boldsymbol{R}_n)^{-1} \tag{5.4}$$

Where $\boldsymbol{P}_{n,n-1}$ is the predicted estimate covariance calculated by Equation 5.2 during the previous iteration and $\boldsymbol{R}_n$ denotes the measurement covariance matrix at time n.
Lastly the fifth equation adjusts the estimate covariance matrix:

$$\boldsymbol{P}_{n,n} = (\boldsymbol{I} - \boldsymbol{K}_n\boldsymbol{H})\boldsymbol{P}_{n,n-1}(\boldsymbol{I} - \boldsymbol{K}_n\boldsymbol{H})^T + \boldsymbol{K}_n\boldsymbol{R}_n\boldsymbol{K}_n^T \tag{5.5}$$

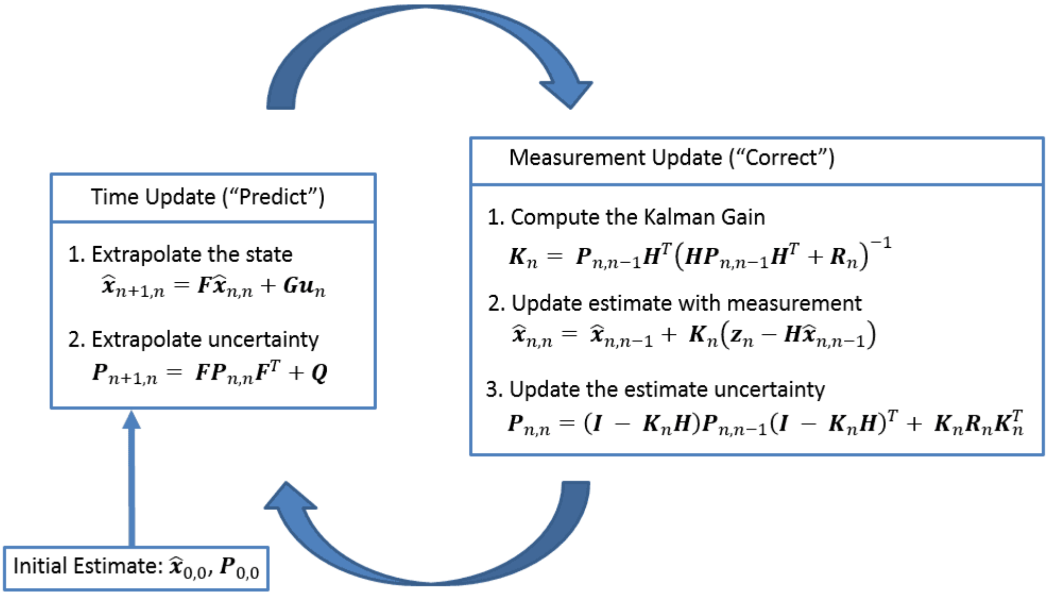

Time Update ("Predict")
1. Extrapolate the state
$\hat{\boldsymbol{x}}_{n+1,n} = \boldsymbol{F}\hat{\boldsymbol{x}}_{n,n} + \boldsymbol{G}\boldsymbol{u}_n$
2. Extrapolate uncertainty
$\boldsymbol{P}_{n+1,n} = \boldsymbol{F}\boldsymbol{P}_{n,n}\boldsymbol{F}^T + \boldsymbol{Q}$
Measurement Update ("Correct")
1. Compute the Kalman Gain
$\boldsymbol{K}_n = \boldsymbol{P}_{n,n-1}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{P}_{n,n-1}\boldsymbol{H}^T + \boldsymbol{R}_n\right)^{-1}$
2. Update estimate with measurement
$\hat{\boldsymbol{x}}_{n,n} = \hat{\boldsymbol{x}}_{n,n-1} + \boldsymbol{K}_n\left(\boldsymbol{z}_n - \boldsymbol{H}\hat{\boldsymbol{x}}_{n,n-1}\right)$
3. Update the estimate uncertainty
$\boldsymbol{P}_{n,n} = (\boldsymbol{I} - \boldsymbol{K}_n\boldsymbol{H})\boldsymbol{P}_{n,n-1}(\boldsymbol{I} - \boldsymbol{K}_n\boldsymbol{H})^T + \boldsymbol{K}_n\boldsymbol{R}_n\boldsymbol{K}_n^T$
Initial Estimate: $\hat{\boldsymbol{x}}_{0,0}$, $\boldsymbol{P}_{0,0}$

Figure 5.2: Kalman filter working cycle. *Illustration by [12]*

This general filter model has to be adapted to the phenomenon being described which is the challenging part of designing a Kalman filter.

These were the key points of a Kalman filter, for a more precise description refer to [12].

5.2.1 Filter design

The design of a Kalman filter starts with the representation of the system model to be described. Following the model derived in [36], a tracking loop, in particular a PAD (PLL assisted DLL) where the information coming from the PLL aids the code tracking loop, can be described with the following differential equation.

$$\underbrace{\begin{bmatrix} \dot{\tau}(t) \\ \dot{\varphi}(t) \\ \dot{f}(t) \\ \dot{a}(t) \end{bmatrix}}_{\dot{\boldsymbol{x}}(t)} = \underbrace{\begin{bmatrix} 0 & 0 & \frac{R_c}{f_r} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} \tau(t) \\ \varphi(t) \\ f(t) \\ a(t) \end{bmatrix}}_{\boldsymbol{x}(t)} + \underbrace{\begin{bmatrix} w_\tau(t) \\ 0 \\ 0 \\ w_a(t) \end{bmatrix}}_{\boldsymbol{w}(t)} \tag{5.6}$$

Since the Kalman state vector has four entries, the Kalman filter is of fourth order, where:

- τ is the difference between the code delay of the local PRN code replica and the authentic one of the incoming signal.
- φ is the difference between the phase of the local carrier replica and the phase of the incoming signal's carrier
- f is the difference between the Doppler shift of the local replica and the Doppler shift of the incoming signal
- a is the difference between the jerk of the local carrier replica and the jerk of the incoming signal

Note that in the first equation we can notice the term responsible for the PAD:

$$\dot{\tau}(t) = \frac{R_c}{f_r} f(t) + w_\tau(t) \tag{5.7}$$

Where R_c is the chipping rate and f_r is carrier frequency of the GNSS signal being tracked. Another aspect worth mentioning is the model employed for the process noise, which assumes Brownian motion for both angular acceleration and code delay.

From Equation 5.6, exploiting the backward Euler transform, the discrete time system model can be derived from the continuous time model:

$$\underbrace{\begin{bmatrix} \tau_{n+1,n} \\ \varphi_{n+1,n} \\ f_{n+1,n} \\ a_{n+1,n} \end{bmatrix}}_{\hat{\boldsymbol{x}}_{n+1,n}} = \underbrace{\begin{bmatrix} 1 & 0 & \frac{R_c}{f_r} T_{int} & \frac{R_c}{f_r} T_{int}^2 \\ 0 & 1 & T_{int} & T_{int}^2 \\ 0 & 0 & 1 & T_{int} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{F}} \underbrace{\begin{bmatrix} \tau_{n,n} \\ \varphi_{n,n} \\ f_{n,n} \\ a_{n,n} \end{bmatrix}}_{\hat{\boldsymbol{x}}_{n,n}} + \underbrace{\begin{bmatrix} T_{int} & 0 & \frac{R_c}{f_r} T_{int}^2 & \frac{R_c}{f_r} T_{int}^3 \\ 0 & T_{int} & T_{int}^2 & T_{int}^3 \\ 0 & 0 & T_{int} & T_{int}^2 \\ 0 & 0 & 0 & T_{int} \end{bmatrix}}_{\boldsymbol{W}} \underbrace{\begin{bmatrix} w_{\tau\ n} \\ 0 \\ 0 \\ w_{a\ n} \end{bmatrix}}_{\boldsymbol{w}_n} \tag{5.8}$$

This is the state prediction equation of the designed Kalman filter presented in Equation 5.1. T_{int} is the integration time during which the correlators calculate their output, so the time interval between two Kalman filter iterations.

The state update equation presented in Equation 5.3 can be written as:

$$\underbrace{\begin{bmatrix}\tau_{n,n}\\ \varphi_{n,n}\\ f_{n,n}\\ a_{n,n}\end{bmatrix}}_{\hat{\boldsymbol{x}}_{n,n}} = \underbrace{\begin{bmatrix}1 & 0 & \frac{R_c}{f_r}T_{int} & \frac{R_c}{f_r}T^2_{int}\\ 0 & 1 & T_{int} & T^2_{int}\\ 0 & 0 & 1 & T_{int}\\ 0 & 0 & 0 & 1\end{bmatrix}}_{\boldsymbol{F}} \underbrace{\begin{bmatrix}\tau_{n,n-1}\\ \varphi_{n,n-1}\\ f_{n,n-1}\\ a_{n,n-1}\end{bmatrix}}_{\hat{\boldsymbol{x}}_{n,n-1}} + T_{int} \underbrace{\begin{bmatrix}v_3 & \alpha_3 & \beta_3\\ v_2 & \alpha_2 & \beta_2\\ v_1 & \alpha_1 & \beta_1\\ v_0 & \alpha_0 & \beta_0\end{bmatrix}}_{\boldsymbol{K_n}} \underbrace{\begin{bmatrix}\delta\tau_n\\ \delta\varphi_n\\ \delta f_n\\ \delta a_n\end{bmatrix}}_{\delta\boldsymbol{z}_n} \tag{5.9}$$

The vector $\boldsymbol{z}_n$ is the actual input of the Kalman filter, corresponding to the delay, phase and frequency discriminator outputs.

The therm $\delta\boldsymbol{z}_n$ is commonly known as innovation; it measures the extent of deviation between the previous prediction and the actual measurements.

$$\delta\boldsymbol{z}_n = \underbrace{\begin{bmatrix}z_{\tau\ n}\\ z_{\varphi\ n}\\ z_{f\ n}\end{bmatrix}}_{\boldsymbol{z}_n} - \underbrace{\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}}_{\boldsymbol{H}} \underbrace{\begin{bmatrix}\tau_{n,n-1}\\ \varphi_{n,n-1}\\ f_{n,n-1}\\ a_{n,n-1}\end{bmatrix}}_{\hat{\boldsymbol{x}}_{n,n-1}} \tag{5.10}$$

Lastly, it is necessary to estimate the twelve terms of the Kalman gain matrix.

Authors of [36] present an interesting approximation that reduces the complexity of runtime computations. Considering Equation 5.4, the inverse of the matrix in brackets can be computationally demanding. For this reason by assuming independent measurements and:

$$\boldsymbol{R}_{i,j} \gg \left(\boldsymbol{H}\boldsymbol{P}_{n,n-1}\boldsymbol{H}^T\right)_{i,j} \qquad \forall i,j = 1,2 \tag{5.11}$$

$$R_\tau \gg R_f\frac{{R_c}^2}{f_r} \gg R_\varphi\frac{{R_c}^2}{f_r} \tag{5.12}$$

Equation 5.4 simplifies to :

$$\boldsymbol{K}_n = \boldsymbol{P}_{n,n-1}\boldsymbol{H}^T\boldsymbol{R}_n^{-1} \tag{5.13}$$

Which is much easier to evaluate since $\boldsymbol{R}_n$ is a diagonal matrix containing the measurements variances: R_τ, R_φ and R_f.

The derivation of the estimate covariance $\boldsymbol{P}$ proposed in [36] is quite complex; based on a steady state analysis in continuous time, it involves the solution of a Continuos-time Riccati Equation

(CARE). Then Equation 5.13 leads to the continuous time Kalman gains which multiplied by the integration time approximates the discrete time ones.

$$\boldsymbol{K} \approx T_{int} \begin{bmatrix} k & \frac{R_c}{f_r} 2\gamma & \frac{R_c}{f_r} 2\gamma^2 \frac{R_\varphi}{R_f} \\ 0 & 2\gamma & 2\gamma^2 \frac{R_\varphi}{R_f} \\ 0 & 2\gamma^2 & 3\gamma^3 \frac{R_\varphi}{R_f} \\ 0 & \gamma^3 & 2\gamma^4 \frac{R_\varphi}{R_f} \end{bmatrix} \tag{5.14}$$

The values γ and k, are equal to:

$$\gamma = \left(\frac{q_a}{R_\varphi}\right)^{1/6} \qquad k = \left(\frac{q_\tau}{R_\tau}\right)^{1/2} \tag{5.15}$$

Where q_a and q_τ are the process noises variances associated to the delay and angular acceleration errors. These two values are unknown and γ and k can be modeled as two adjustable parameters related to the DLL and PLL loop filter bandwidths as follows.

$$\gamma = \frac{6}{5} PLL_{band} \qquad k = 4DLL_{band} \tag{5.16}$$

It's important to notice the match between the behaviour of a loop filter depending on its bandwidths presented in Subsection 3.4.3 and the behaviour of a Kalman filter depending on its gain. Higher bandwidths lead to higher Kalman gains thus in the state update Equation 5.3 the innovation term gets a bigger weight in respect to the prediction, making the Kalman state evolution more sensitive to the discriminator's outputs, so less smooth and more robust to dynamics. On the contrary smaller bandwidths and so smaller Kalman gains reduce the impact of the discriminator's outputs on the state update equation, making the state evolution smoother but less robust to dynamics.

To better understand how the Kalman filter works, Figure 5.3 shows the evolution of the phase discriminator's output in red, the Kalman state phase estimation in blue and the prediction of the phase error estimate computed by the Kalman filter at the previous iteration in green. It can be noted how the estimate stored in the Kalman state is basically the prediction performed at the previous cycle perturbed by the phase discriminator output which is given as input to the Kalman filter.

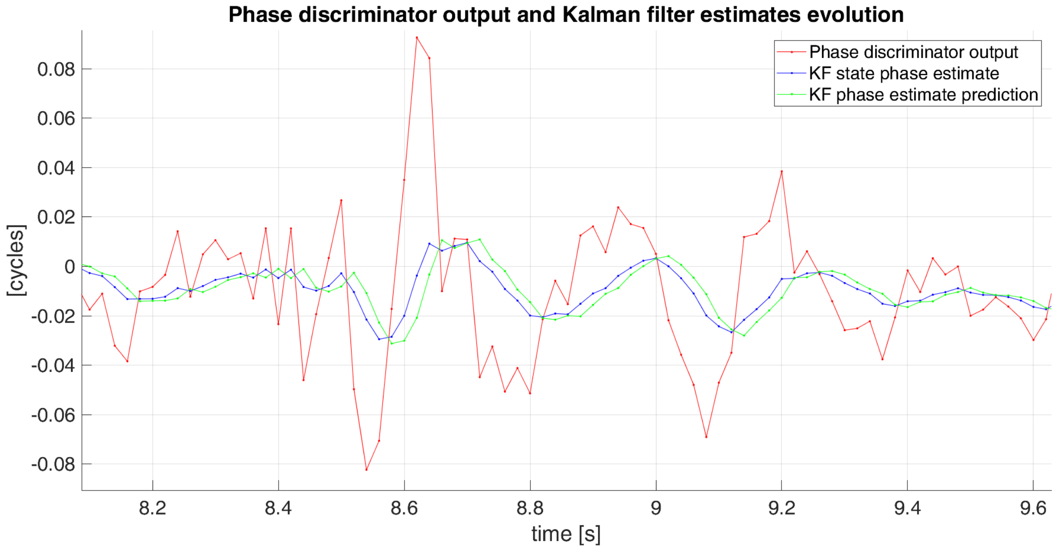


Figure 5.3: Example of Kalman filter evolution of the phase error estimate.

5.3 Outage detection algorithm

When an obstacle interrupts the LOS signal between a satellite and the receiver, an outage occurs. This results in a drop in power of the received signal and consequently the tracking loops start tracking the channel's noise. The discriminators outputs also become noisy directly affecting the code delay, carrier phase and frequency error estimates, resulting in the loss of lock of the signal. At this point the receiver is forced to perform again the acquisition process in order to demodulate the satellite's signal once more.
Adding a Kalman filter to the standard loop structure can alleviate this issue. In case of outage the error estimates of the tracking loop, the first three entries of the state vector, are less affected by the discriminator outputs since there is the contribution of the predicted state which tends to propagate the evolution experienced in the past. Despite that first tests made clear that this wasn't enough. In order to reduce and actually eliminate the noise contribution when there is no LOS signal to track, an outage detection algorithm has been developed with the purpose of temporarily set the discriminator's outputs to zero and so rely only on the predictions of the Kalman filter.

The algorithm working procedure can be separated into two distinct phases: the outage declaration and an outage recovery phase.

5.3.1 Outage Declaration phase

The outage declaration needs to be as fast as possible. The first tempted approach was to rely on the $C/N0$ estimates produced by the receiver; whenever the carrier to noise ratio drops under a certain threshold a new outage is declared. However this approach had a significant drawback. The estimates produced by the receiver are result of a filtration and the transient generated by the filter when the signal's power drops, even if it only lasts for a few hundreds of milliseconds, causes a delay in the outage declaration which is enough to cause the tracking loop to lose the lock of the incoming signal.

A much faster solution based on the frequency innovation values of the KF has been adopted. Each time this value gets larger than an adaptive threshold, the outage is declared and the discriminator's outputs, the $\boldsymbol{z}$ vector entries, are set to zero. Note that the choice to rely only on the frequency innovation was found to be more reliable rather than considering also the delay and phase innovation evolutions.

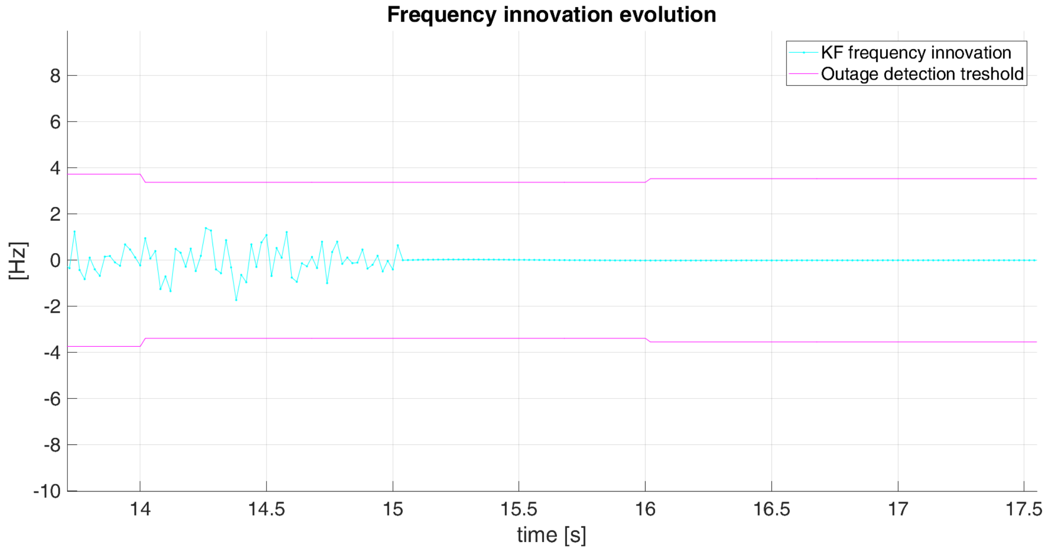


Figure 5.4: Outage declaration example.

In Figure 5.4 is shown an example of outage declaration, precisely at $15.08s$. Please notice how the first innovation sample out of bounds is not displayed since the discriminator's outputs have been already set to zero. Furthermore it can be noticed how at $14s$ and $16s$ the threshold value gets updated. The updating criteria will be explained later in Subsection 5.3.3.

5.3.2 Outage recovery phase

Hopefully, after a certain time, the LOS signal from the satellite will be available again. Assuming that the Kalman filter has propagated correctly, the signal's power will rise back and the tracking loop should restart the normal tracking procedure restoring the discriminator's outputs as the inputs to the Kalman filter.

This switch doesn't need to be instantaneous as it is preferable to tolerate some extra delay in order to restart the tracking loop with clearer samples. Therefore the end of a signal blockage can be declared when the estimated $C/N0$ rises back to a user-defined value. This value has to be carefully set; a too low threshold would risk to reactivate the tracking loop too soon or even during an outage because of an oscillation of the $C/N0$ estimate, on the contrary a high value could let the Kalman filter propagate even if a weak signal is received after a blockage.

Figure 5.5 provides an example of recovery after an outage.

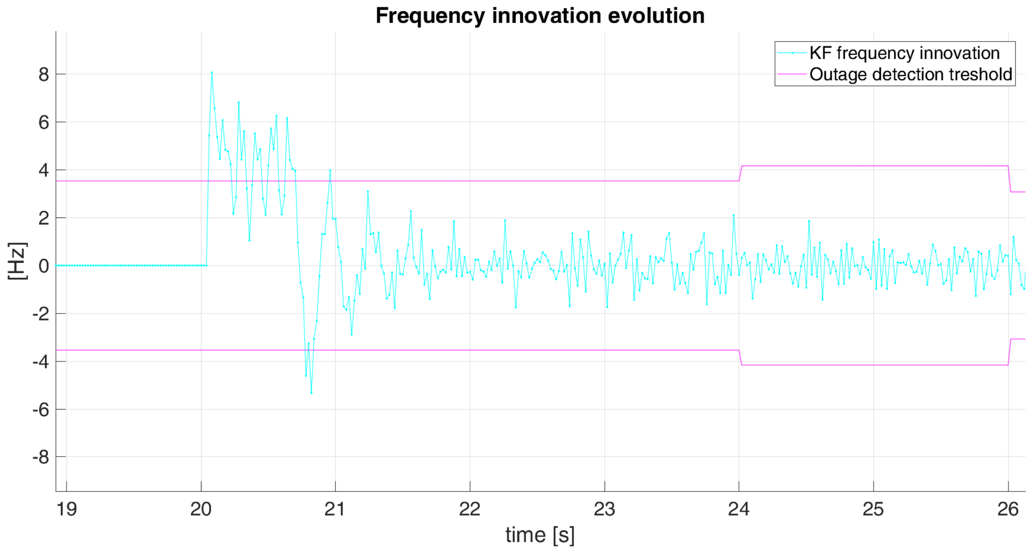


Figure 5.5: Outage recovery example.

From this example, it is clear that a timeout period is necessary to let the tracking loop adjust its estimates after a signal blockage before declaring a new one. With this purpose the algorithm is set ready to declare a new outage only after n innovation samples fall inside the thresholds consecutively. Different values of n have been tested, between 10 and 100, corresponding on $0.2\ s$ and $2\ s$ since $T_{int} = 20ms$.

A full example of the outage detection algorithm's working cycle is illustrated in Figure 5.6.

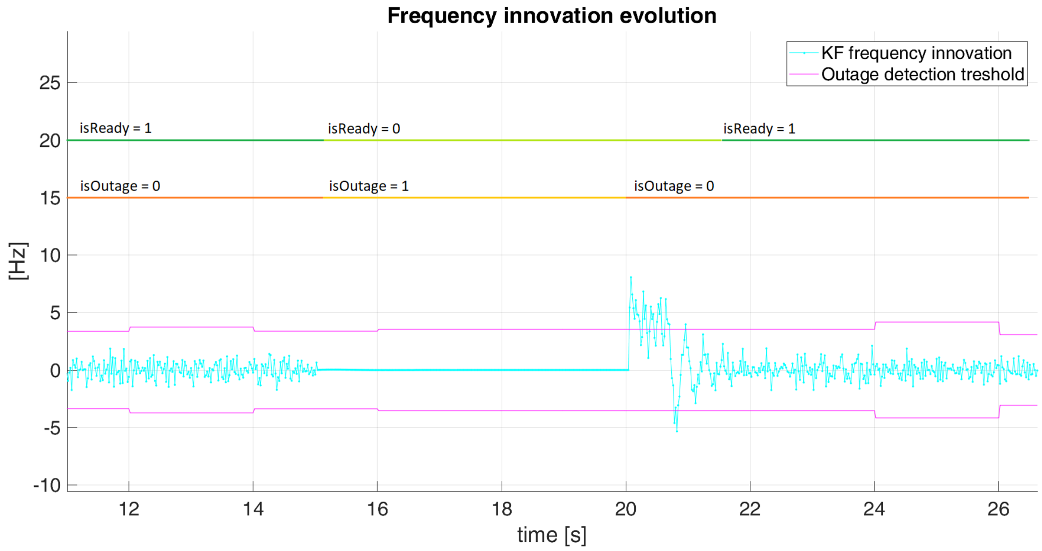


Figure 5.6: Outage detection algorithm working cycle. The two flags *isOutage* and *isReady* control when the receiver is declared to be in outage and when the algorithm is ready to declare a new outage respectively.

In addition, it is crucial to approach the restart of the DLL with extra caution. During the recovery of the PLL after an outage, the frequency estimation error is high and the difference between the frequency of local replica of the carrier and the frequency of the incoming signal may cause the DLL to lose the lock. For this reason the output of the DLL discriminator has been limited to 1 *chip*, if an higher value is computed it gets set to zero letting the Kalman filter propagate with the contribution of the carrier aiding term.

5.3.3 Threshold updating rule

The frequency innovation threshold of the outage detection algorithm gets updated every T_{th} seconds according to:

$$b\sqrt{var(z_f)} \tag{5.17}$$

Both b and T_{th} are parameters that the user can set. The variance estimation is computed every T_{th} seconds considering only the discriminator outputs when the receiver is not experiencing an outage and is ready to detect one. If these values are too few for the variance estimation, the threshold won't get updated.

6 Simulations and results

The proposed tracking loop presented in Chapter 5 has been implemented and extensively tested in Matlab as an upgrade of a semi-analytical software receiver simulator developed by Qascom back in 2015. This tool simulates the general behaviour of a GNSS receiver tracking GPS and GALILEO signals at different bands .
Many parameters can be configured to control the receiver (distance betwee E and L correlators, discriminators types, loop filter bandwidths, spoofing mitigation algorithms...) and the environment in which it's placed (noise level, multipath channel models, satellite elevation, receiver's speed, Doppler evolution, spoofers...).
Despite this, in addition to the implementation of whole tracking loop described in Chapter 5, other changes and enhancements have been incorporated. The original simulator's version was improved to simulate the scenario of a dynamic receiver in an urban area. This will be described in Subsection 6.2.2.

To test the effectiveness of the proposed design, multiple simulations have been performed comparing the Kalman filter based loop with the standard one. The characteristics and parameters of the two tracking loops being compared are presented in Table 6.1.

Table 6.1: Compared tracking loop configurations

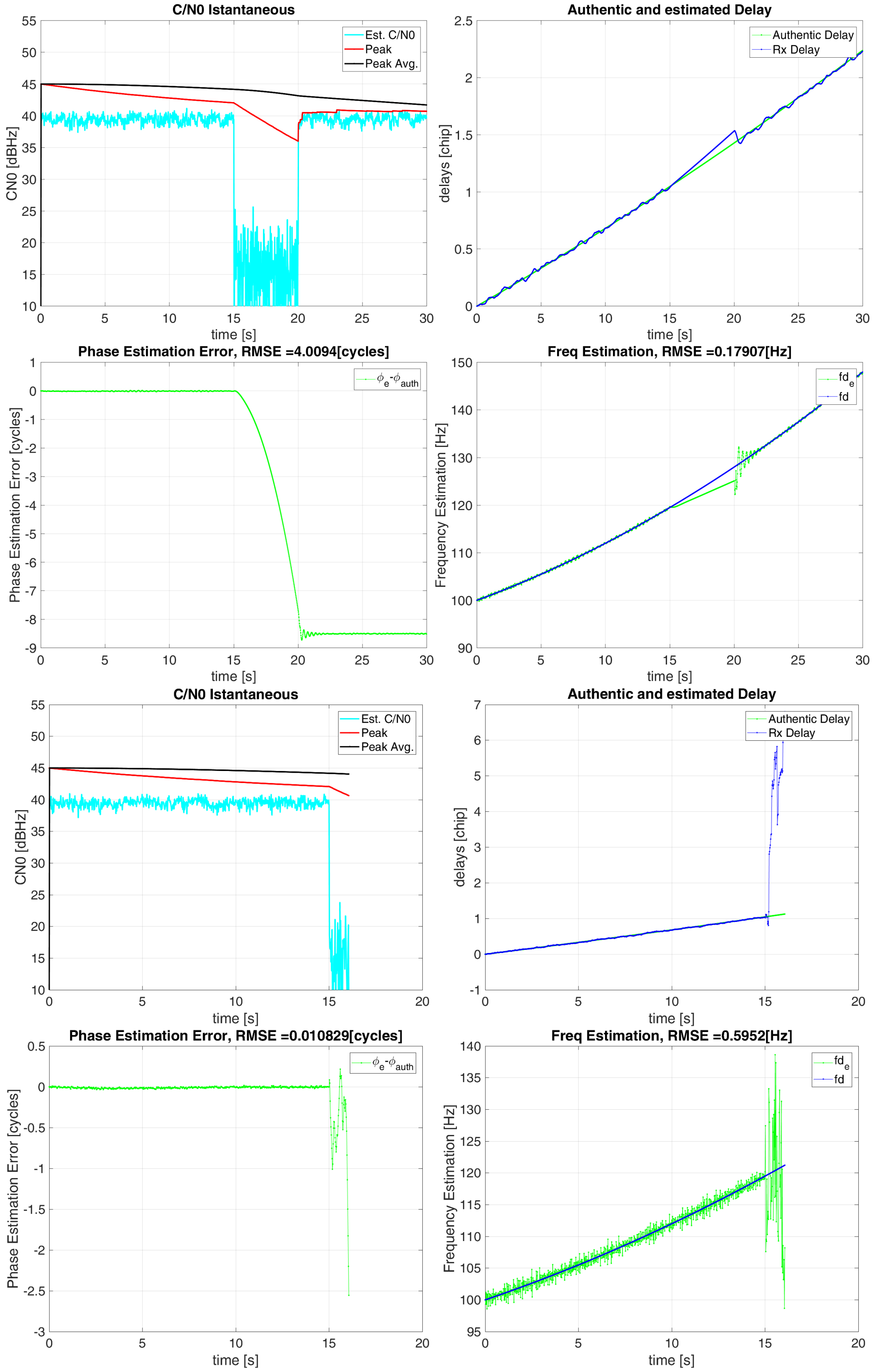
	KF based tracking loop	Standard tracking loop
DLL bandwidth	$2Hz$	$2Hz$
PLL bandwidth	$3Hz$	$3Hz$
FLL bandwidth	$4Hz$	$4Hz$
DLL discriminator	$EMLP$	$EMLP$
PLL discriminator	$atan$	$atan$
FLL discriminator	$atan2$	$atan2$
b value [1]	12	
n value [2]	100	
$C/N0$ threshold	$17dBHz$	
T_{th} [1]	$2s$	

[1] *Checkout Subsection 5.3.3*

[2] *Checkout Subsection 5.3.2*

6.1 Artificial outages in AWGN channel

The initial approach to test the efficacy of the tracking loop implementation was to simulate a signal blockage by selecting a time window in which the signal power drops.
Figure 6.1 displays how the proposed tracking loop performs during a 5 seconds outage in an AWGN channel compared to the standard one.



C/N0 Istantaneous

Authentic and estimated Delay

Phase Estimation Error, RMSE =4.0094[cycles]

Freq Estimation, RMSE =0.17907[Hz]

C/N0 Istantaneous

Authentic and estimated Delay

Phase Estimation Error, RMSE =0.010829[cycles]

Freq Estimation, RMSE =0.5952[Hz]

Figure 6.1: Proposed tracking loop design estimates (on top) and standard tracking loop ones (bottom) while tracking a simulated outage.

This demonstrates the effectiveness of the proposed solution. Despite complete signal blockage and subsequent power loss, the lock is not lost. Code delay and frequency estimates, propagated by the Kalman filter, are reliable enough to restart the tracking procedure upon power restoration. Note that the phase estimate locks into an integer multiple of a cycle, it could have also locked to a multiple of half a cycle which would result in a counterphase lock.

6.2 DLR Land mobile Multipath Channel Model

After verifying the functioning of the tracking loop in a simplified scenario, a more accurate channel model generated with the DLR (Deutsches Zentrum für Luft- und Raumfahrt) Land Mobile Multipath Channel Model [44] has been tested.
This channel model, developed by the German Aerospace Centre, is a combination of statistical and deterministic approaches. The signal's reflections are modeled statistically, according to a user-defined geometric scenario, instead the direct path is modeled taking into account shadowing and refracting phenomena in a deterministic way. This allows the user to design a schematic of a city road [13] by specifying the receiver position on the street and the mean and variance values of the buildings length, height and gaps as well as position and dimensions of trees and lampposts. The values used in the simulations performed are presented in Table 6.2.

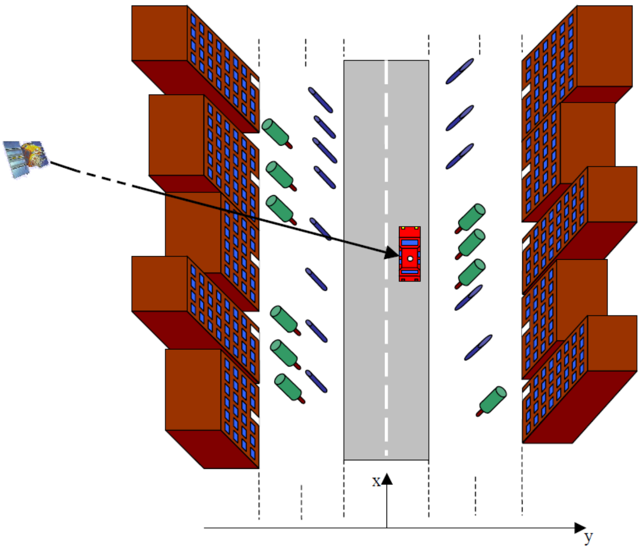


Figure 6.2: DLR urban environment model. *Illustration by [13]*

Table 6.2: DLR urban scenario simulation parameters

Building parameters	
BuildingRow1	1
BuildingRow2	1
BuildingRow1YPosition	-7 m
BuildingRow2YPosition	7 m
HouseWidthMean	25 m
HouseWidthSigma	1 m
HouseWidthMin	15 m
HouseHeightMin	5 m
HouseHeightMax	30 m
HouseHeightMean	10 m
HouseHeightSigma	5 m
GapWidthMean	8 m
GapWidthSigma	4 m
GapWidthMin	5 m
BuildingGapLikelihood	0.8

Tree parameters	
TreeHeight	3 m
TreeDiameter	3 m
TreeTrunkLength	2 m
TreeTrunkDiameter	1 m
TreeAttenuation	1.1 dB/m
TreeRow1Use	1
TreeRow2Use	1
TreeRow1YPosition	-5.5 m
TreeRow2YPosition	5.5 m
TreeRow1YSigma	0 m
TreeRow2YSigma	0 m
TreeRow1MeanDistance	15 m
TreeRow2MeanDistance	10 m
TreeRow1DistanceSigma	3 m
TreeRow2DistanceSigma	3 m

Pole parameters	
PoleHeight	9 m
PoleDiameter	0.4 m
PoleRow1Use	1
PoleRow2Use	1
PoleRow1YPosition	5.5 m
PoleRow2YPosition	-5.5 m
PoleRow1YSigma	0 m
PoleRow2YSigma	0 m
PoleRow1MeanDistance	15 m
PoleRow2MeanDistance	15 m
PoleRow1DistanceSigma	5 m
PoleRow2DistanceSigma	5 m

The satellite's elevation is set to 50°, in order to generate fading effects on the LOS signal from the taller buildings. The vehicle's speed is set to $50 km/h$ and the satellite's signal is affected by a Doppler, modeled as a polynomial, with a shift of $10 Hz$ and a Doppler rate of $5 Hz/s$.

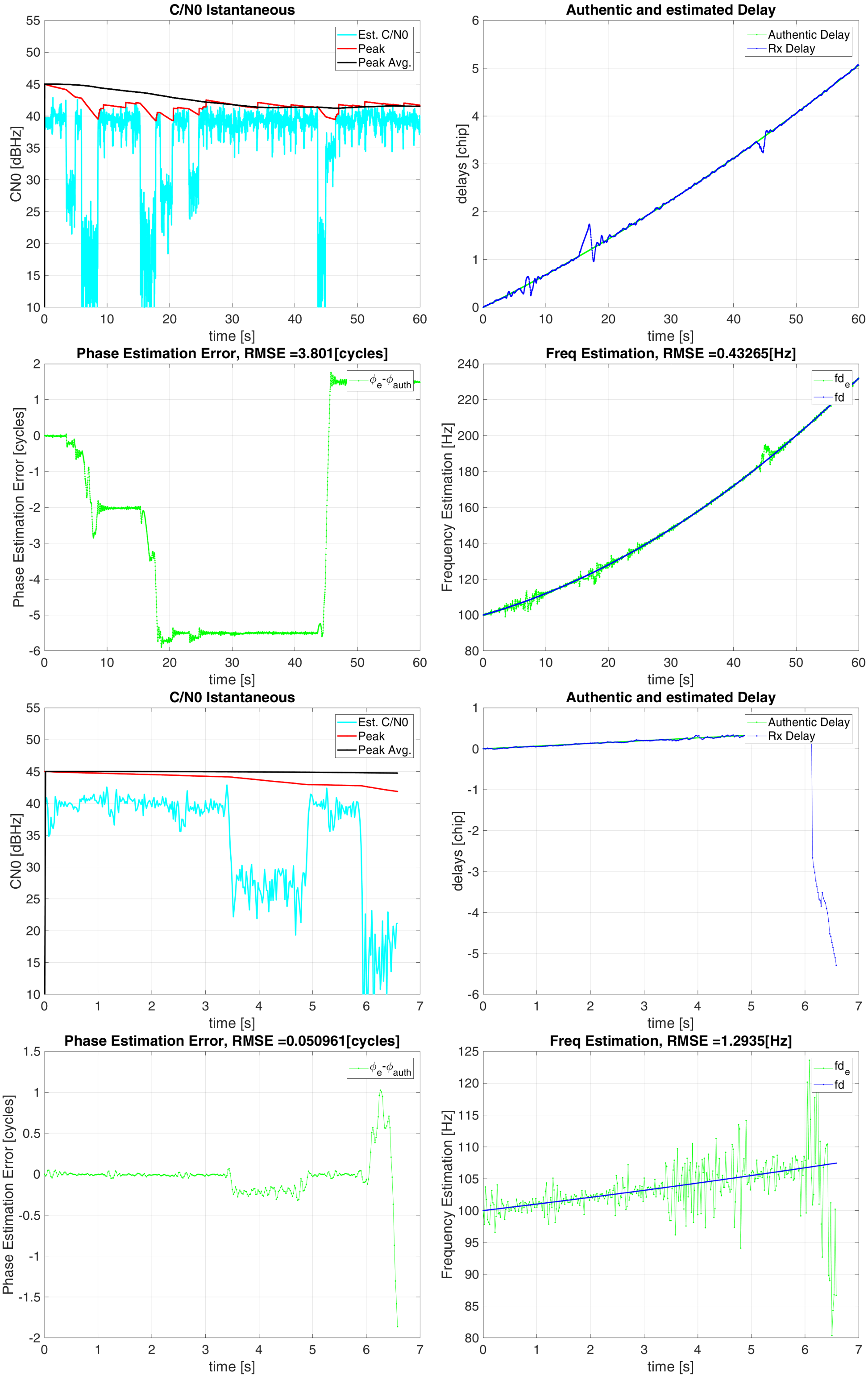


Figure 6.3: Proposed tracking loop design estimates (on top) and standard tracking loop ones (bottom) while tracking a polynomial Doppler evolution in a DLR simulated channel.

6.2.1 Realistic static Doppler profiles

The use of a polynomial Doppler profile is a big limitation in the simulation of a vehicle in a urban environment. To achieve a more realistic Doppler evolution, data collected from a static GNSS receiver positioned on the roof of Qascom's head office have been analyzed. In particular, the estimated carrier frequency sampled by the receiver with a period of $1s$ has been interpolated using the *spline* algorithm in Matlab and then used as an authentic Doppler evolution given as input to the semi-analytic simulator.
Figure 6.4 displays an example of the obtained Doppler profile.

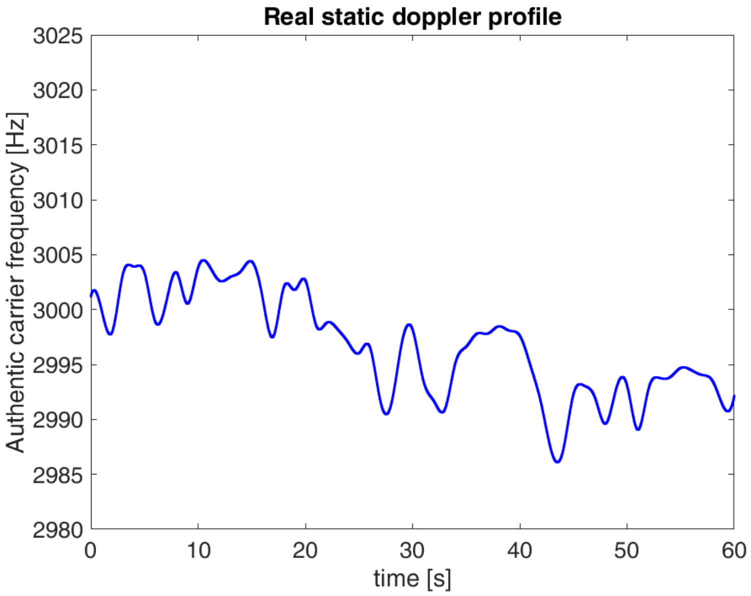


Figure 6.4: Static real Doppler profile example.

Figure 6.5 presents a comparison between the Kalman-based tracking loop and the standard one while tracking the recorded real Doppler profile in the DLR land mobile channel generated according to Table 6.2. This shows the enhanced reliability of the proposed tracking loop while subjected to a static urban simulated scenario.

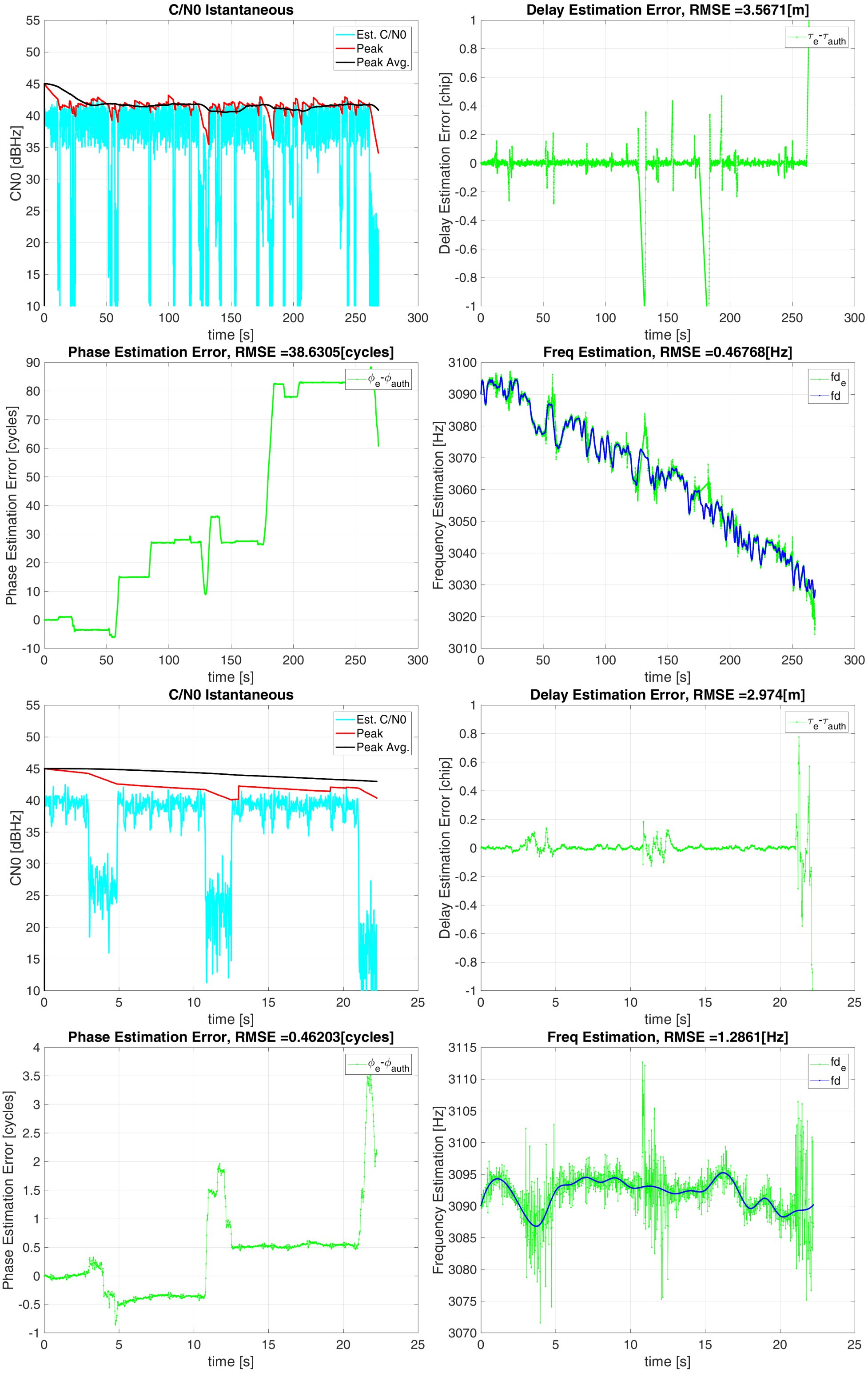


Figure 6.5: Proposed tracking loop design estimates (on top) and the standard tracking loop ones (bottom) while tracking a static real Doppler evolution in a DLR simulated channel.

To assess the gain in performance of the designed tracking loop, $N = 20$ simulations have been performed with different Doppler profiles and different DLR channel realizations. As a performance indicator the average lock time of each configuration has been estimated.

$$\bar{t}_{lock} = \frac{1}{N} \sum_{i=1}^{N} t_{lock} \tag{6.1}$$

Note that the simulations last 5 minutes, long enough to ensure that both the standard and the proposed tracking loop have, most likely, lost the signal's lock.

Table 6.3: Mean lock times in DLR static scenario estimated with 20 simulations.

Mean lock time	
KF-based traking loop	2 minutes and 25 seconds
Standard tracking loop	15 seconds

The evolution over time of the probability that the tracking loop retained the signal lock is depicted in Figure 6.6.

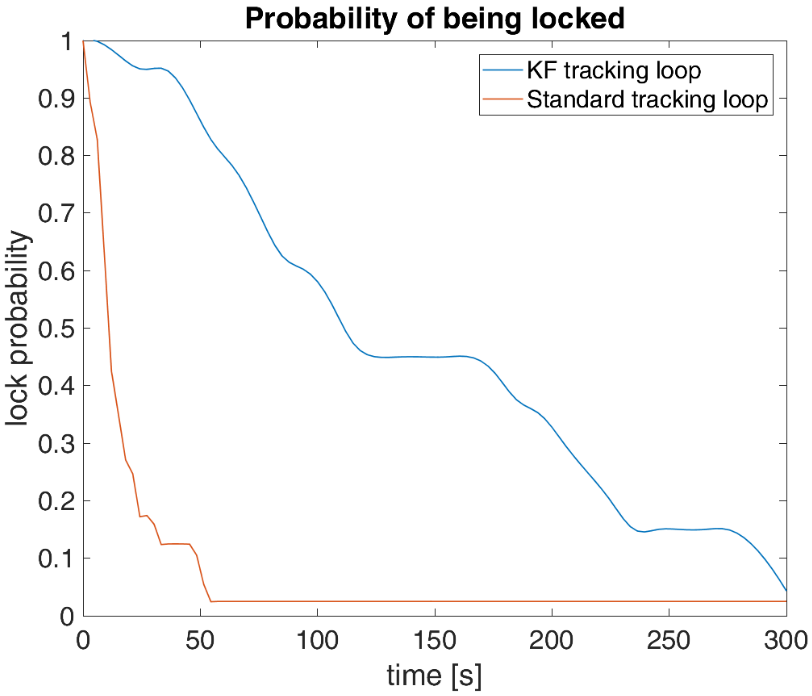


Figure 6.6: Lock probability evolution of the compared tracking loops in static urban scenarios.

6.2.2 Realistic dynamic Doppler profiles

To further enhance the authenticity of the simulation modelling a dynamic receiver in an urban scenario, the Doppler evolution presented in Subsection 6.2.1 has been adjusted to incorporate variations in the receiver's speed. By setting a fixed acceleration/deceleration of the receiver and fixing a speed gain or loss at a specified time, the static Doppler evolution is modified to account for the new speed values.

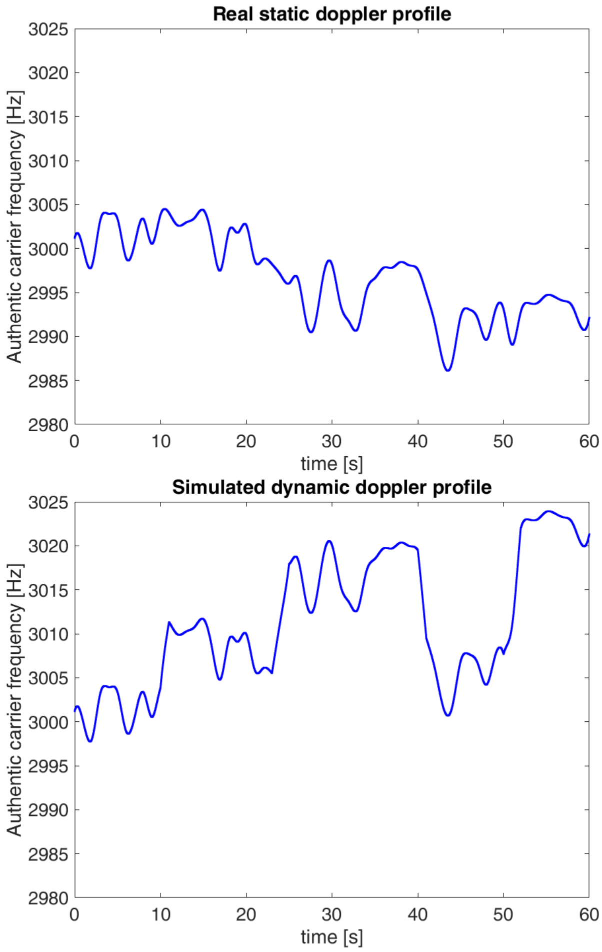


Figure 6.7: Static real doppler profile and simulated dynamic profile with speed changes at $10s$, $23s$,$40s$ and $50s$ assuming a vehicle acceleration/deceleration of $1.38m/s^2$ ($5km/h/s$).

These new dynamic Doppler profiles have been tested with the channel impulse response generated by the DLR simulator. In the figures below the two configurations presented in Table 6.1 are compared in the same scenario.

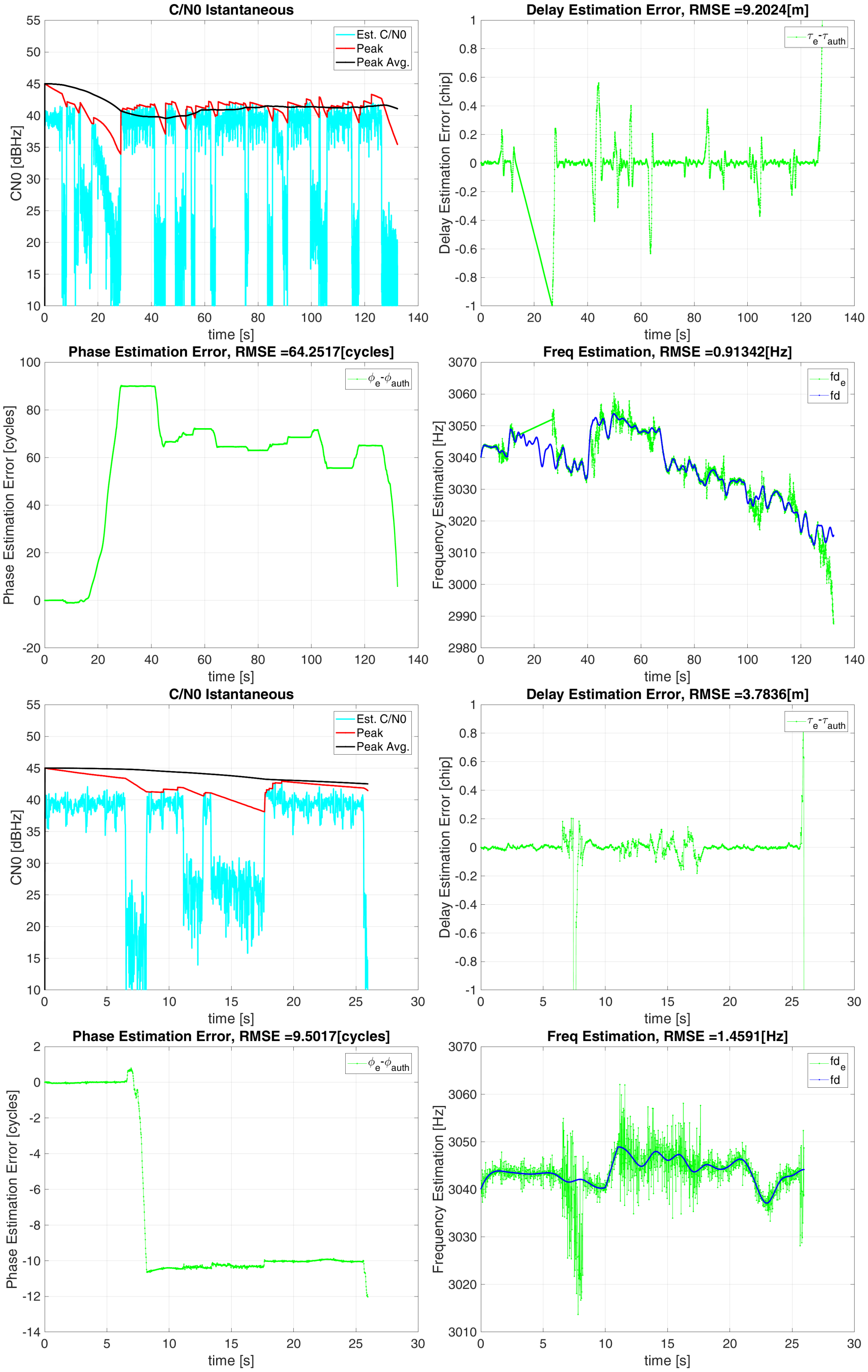


Figure 6.8: Proposed tracking loop design estimates (on top) and standard tracking loop ones (bottom) while tracking a dynamic Doppler evolution in a DLR simulated channel.

As expected the robustness enhanced tracking loop demonstrates greater resilience to signal fluctuations; nonetheless, as shows Figure 6.9, if the outage takes place just prior to, during or immediately following a change in the receiver dynamics, it becomes more likely that the Kalman filter won't propagate the estimates correctly leading to a loss of lock.

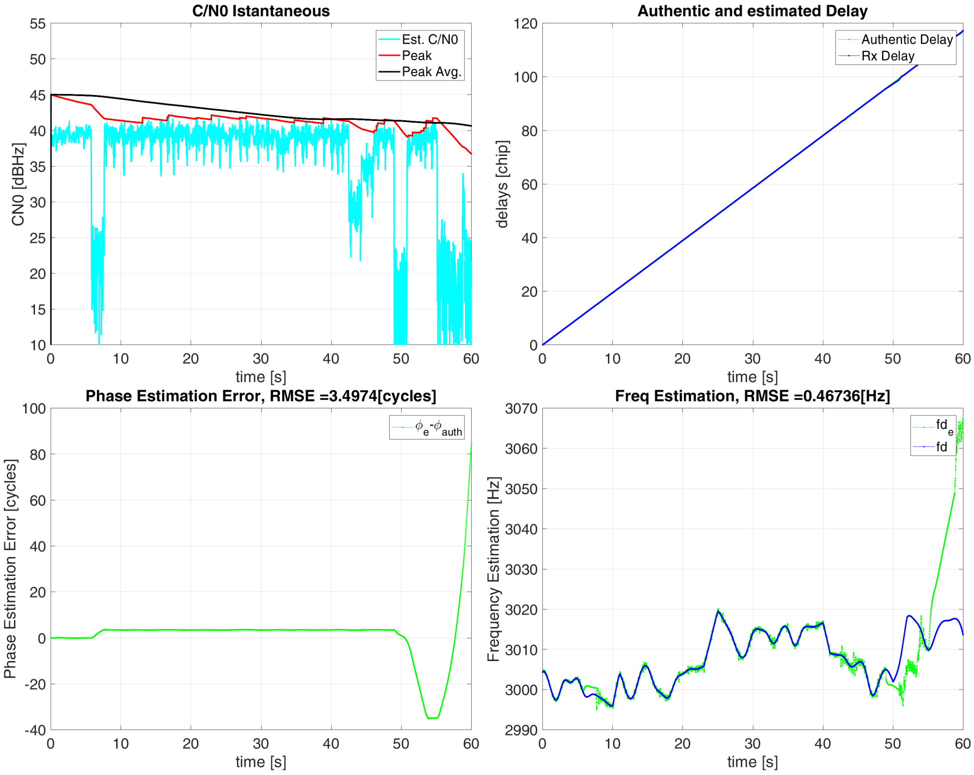


Figure 6.9: Example of loss of lock of the Kalman filter based tracking loop.

As in the case of the static scenario, the two versions of the tracking loop were compared through $N = 20$ simulations with the aim of calculating the average lock time Equation 6.1. The results obtained are shown in Figure 6.10.

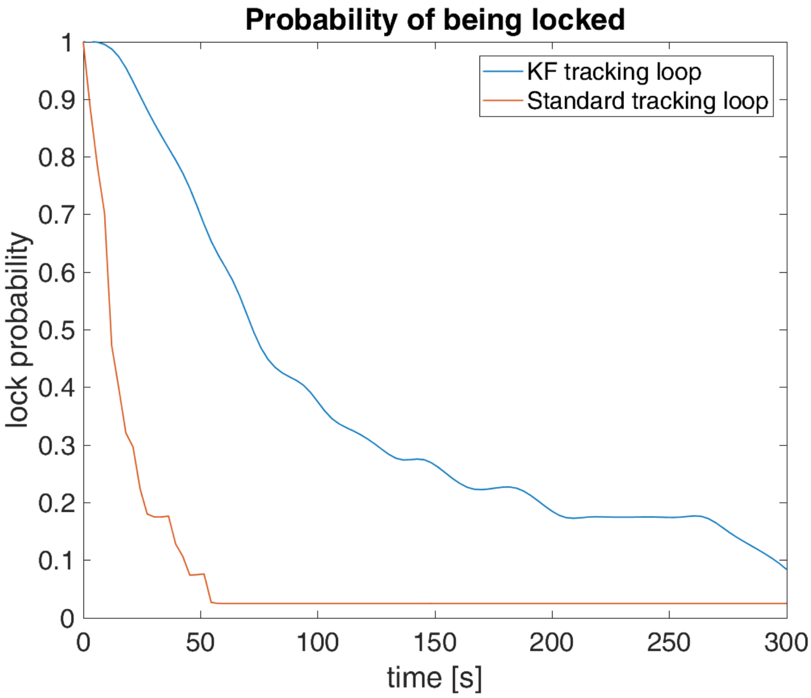


Figure 6.10: Lock probability evolution of the compared tracking loops in dynamic urban scenarios.

Table 6.4: Mean lock times in DLR dynamic scenario estimated with 20 simulations.

Mean lock time	
KF-based traking loop	1 minute and 45 seconds
Standard tracking loop	16 seconds

As might be expected, the introduction of greater dynamics in the receiver causes a deterioration in performance. However, by increasing the distance between the early and late correlators, from 0.5*chips* to 1*chip*, the performance of the tracking loop becomes similar as in the static scenario, achieving an average lock time close to 2 minutes 30 seconds. Wider correlators make the receiver less accurate but more robust to dynamics. Figure 6.2.2 displays a comparison between the three versions.

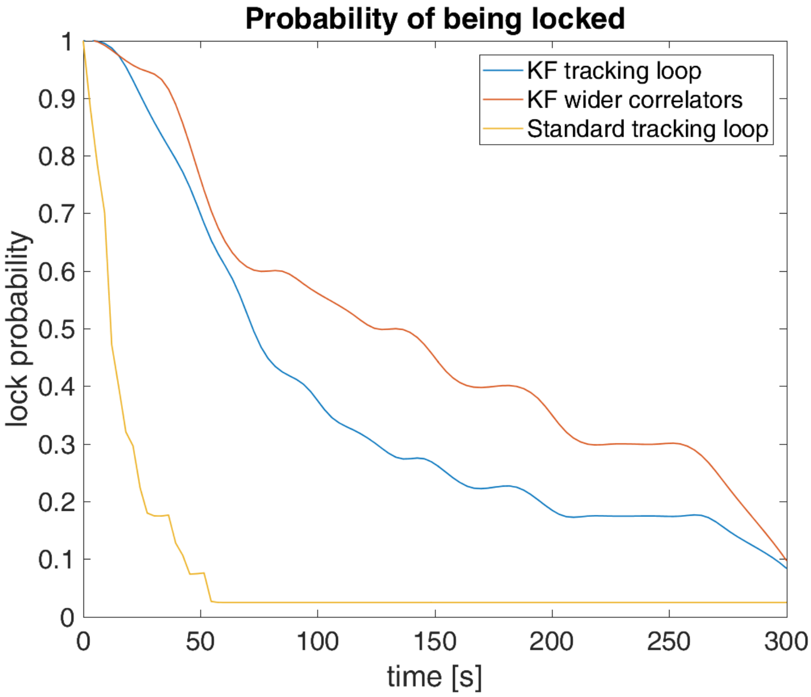


Figure 6.11: Lock probability evolution of the compared tracking loops and of a wide correlator KF loop in dynamic urban scenarios.

6.3 GNSS receiver live records

The last scenario used to test the designed receiver is based on some live recordings of the Doppler and $C/N0$ estimates of a Septentrio GNSS receiver placed on a vehicle driven among streets in Bassano del Grappa while tracking several satellites belonging to GALILEO and GPS constellations. In particular these estimates are sampled every second, then interpolated and used as Doppler and $C/N0$ profiles as input to the semi-analytic simulator.
Figure 6.12 shows a comparison between the two setup presented in Table 6.1 while tracking GALILEO's E21 satellite, transmitting in E5b band.

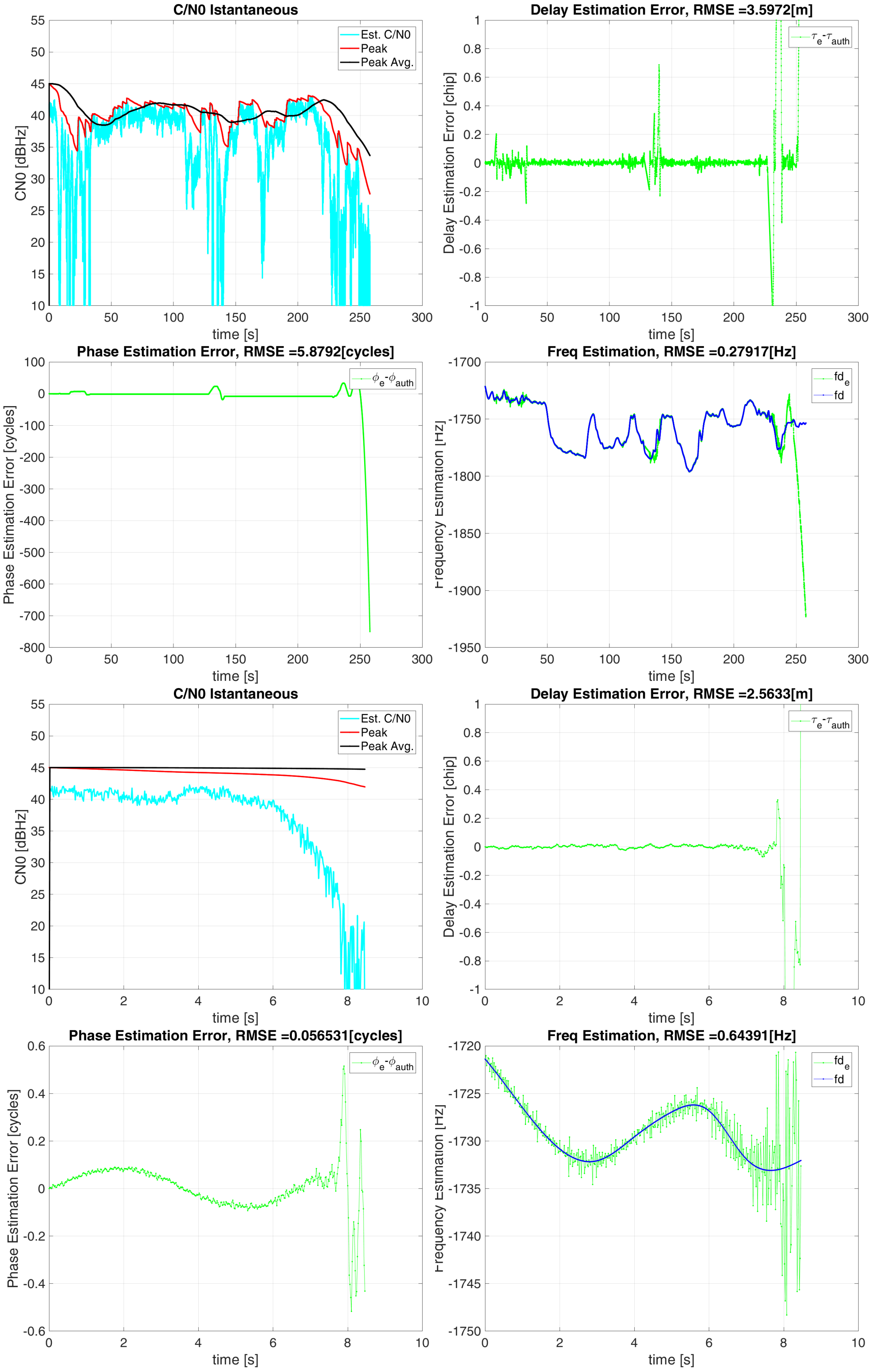


Figure 6.12: Proposed tracking loop design estimates (on top) and standard tracking loop ones (bottom) while tracking a live recorded dynamic scenario.

The result is quite similar to the simulated one presented in Figure 6.8. Similarly as shown in Figure 6.9, a change in the vehicle dynamics concurrently with a signal's blockage will most likely make the receiver loose the lock.

The lock time has been evaluated over $N = 100$ windows of 5 minuts taken from data captured while tracking different satellites belonging to both GPS and GALILEO constellations. Figure 6.13 displays the obtained results.

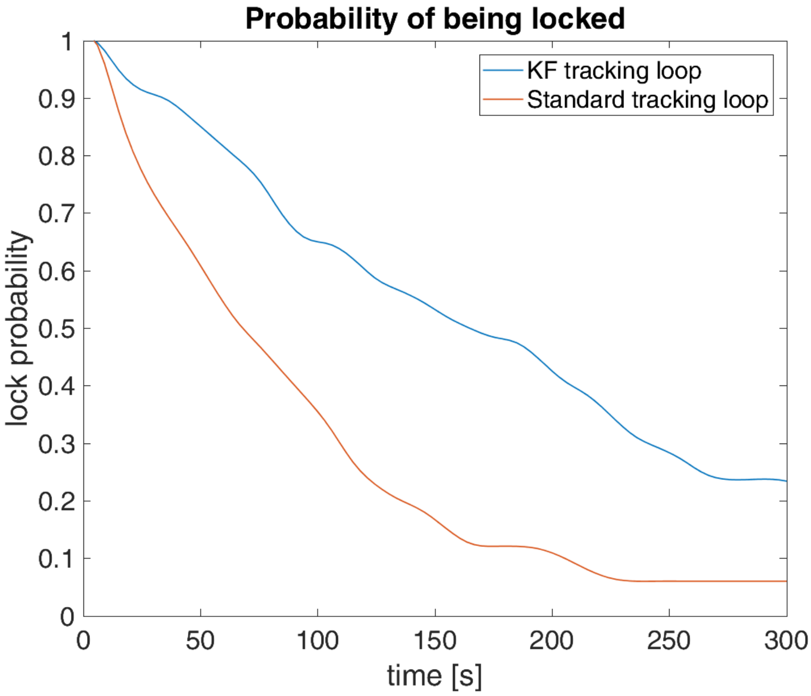


Figure 6.13: Lock probability evolution of the compared tracking loops in live dynamic scenarios.

Table 6.5: Mean lock times in live recorded scenario estimated with 100 simulations.

Mean lock time	
KF-based traking loop	3 minutes and 12 seconds
Standard tracking loop	2 minutes

Note that at the end of the 5 *minutes simulations over* 20% *of KF tracking loop runs where still locked*

The results observed are in line with what the previous simulations underlined. In this case the average lock time of the proposed tracking loop is over 3 minutes, a bit higher than the one observed in Subsection 6.2.2 since the simulated scenario tends to be more hostile.

7
Conclusion

7.1 Summary

The main purpose of this thesis was to enhance the tracking capability of a software defined GNSS receiver in the hostile scenario of an urban environment. In the previous chapters the implementation of a Kalman filter-based tracking loop assisted by an outage detection algorithm has been analyzed and the novel tracking loop's performances have been evaluated in both completely simulated and more realistic scenarios using a semi-analytic simulator.
The mean lock times in each simulation scenario, of both the standard and the proposed tracking loop, are the following:

Table 7.1: Mean lock times in each scenario

	DLR static	DLR dynamic	Live recordings
KF-based traking loop	2 min 25 s	1 min 45 s	3 min 12 s
Standard tracking loop	15 s	16 s	2 min

The enhancement of performance is evident, nevertheless some clarifications should be made. Firstly it should be noted that the Doppler and $C/N0$ evolutions derived from the live receiver recordings are the outcome of its tracking procedure sampled every second, and not an exact capture of the actual Doppler and $C/N0$ measurements of the incoming signal.
Another aspect to be considered is that the 5 minutes long windows generated from the live

recordings have been manually selected to avoid signal outages longer than $10s$ and windows with no signal blockage. In the first case the probability to recover from the blockage is too low for both the configurations, whereas in the second case both the tracking loops would easily track the signal.

Lastly the parameters outlined in Table 6.1, employed to configure the proposed and the standard loop, highly influence the behaviour of the receiver. The values considered have been chosen after a rough optimization for both the Kalman-based and standard tracking loop. Therefore, the configurations displayed in Table 6.1 may not be optimal.

7.2 Future work

The objective of this study was to find an effective way to make the receiver more robust in an hostile scenario. The outcomes demonstrated a general improvement, with room for further developments.

The most important step would be transferring the Kalman filter's and outage detection algorithm's codes from Matlab to C++ and integrating them in the software defined QN400 receiver. In this way the effectiveness of the proposed solution could be verified directly by testing the receiver on field.

Another critical aspect is the selection of the Kalman filter and outage detection algorithm parameters; a Monte Carlo optimization would refine the values presented in Table 6.1.

Moreover an approach similar to the Context Aware Navigation presented in Section 4.1 could be introduced: the receiver, using perhaps a machine learning based approach, could recognize the current type of scenario and adjust its settings accordingly.

Some further extensions could be made in the Kalman filter structure: the state prediction equation could be redefined taking into account IMU aiding which, exploiting measurements coming from gyroscopes and accelerometers, could improve the prediction of the tracking loop estimates taking into account the measured dynamics of the receiver.

References

[1] E. D. Kaplan and C. Hegarty, *Understanding GPS/GNSS: principles and applications*. Artech house, 2017.

[2] M. H.-P. J. Sanz Subirana, J.M. Juan Zornoza, *GNSS Data Processing, Volume I: Fundamentals and Algorithms*. ESA Communications, 2013.

[3] D. R. T. Charles Jeffrey, Roger Munro, *An introduction to GNSS: A primer in using Global Navigation Satellite Systems for positioning and autonomy*. Hexagon, 2023.

[4] C. A. Ogaja, *Introduction to GNSS Geodesy: Foundations of Precise Positioning Using Global Navigation Satellite Systems*. Springer Nature, 2022.

[5] [Online]. Available: https://gssc.esa.int/navipedia

[6] T. Al Sharabati, "Effect of pseudo random noise (prn) spreading sequence generation of 3gpp users' codes on gps operation in mobile handset," *Journal of Communications Software and Systems*, vol. 12, no. 4, pp. 182–189, 2016.

[7] P. J. Teunissen and O. Montenbruck, *Springer handbook of global navigation satellite systems*. Springer, 2017, vol. 10.

[8] J. Zidan, E. I. Adegoke, E. Kampert, S. A. Birrell, C. R. Ford, and M. D. Higgins, "Gnss vulnerabilities and existing solutions: A review of the literature," *IEEE Access*, vol. 9, pp. 153 960–153 976, 2020.

[9] A. Hussain, A. Ahmed, M. A. Shah, S. Katyara, L. Staszewski, and H. Magsi, "On mitigating the effects of multipath on gnss using environmental context detection," *Applied Sciences*, vol. 12, no. 23, p. 12389, 2022.

[10] H. Yao, Z. Dai, W. Chen, T. Xie, and X. Zhu, "Gnss urban positioning with vision-aided nlos identification," *Remote Sensing*, vol. 14, no. 21, p. 5493, 2022.

[11] A. Blais, N. Couellan, and E. Munin, "A novel image representation of gnss correlation for deep learning multipath detection," *Array*, vol. 14, p. 100167, 2022.

[12] A. Becker, “Kalman filter,” *Dipetik Juli*, vol. 1, p. 2020, 2018.

[13] B. Krach, A. Lehner, and A. Steingass, “Technical note on the implementation of the land mobile satellite channel model-software usage,” 2005.

[14] J. J. Parker, F. Dovis, B. Anderson, L. Ansalone, B. Ashman, F. H. Bauer, G. D’Amore, C. Facchinetti, S. Fantinato, G. Impresario *et al.*, “The lunar gnss receiver experiment (lugre),” in *Proceedings of the 2022 International Technical Meeting of The Institute of Navigation*, 2022, pp. 420–437.

[15] G. NAVSTAR, “Joint program office (jpo), gps navstar user’s overview,” YEE-82-009D, Mar, Tech. Rep., 1991.

[16] K. Borre and D. Akos, “A software-defined gps and galileo receiver: single-frequency approach,” in *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, 2005, pp. 1632–1637.

[17] J.-A. Avila-Rodriguez, S. Wallner, G. Hein, E. Rebeyrol, O. Julien, C. Macabiau, L. Ries, A. DeLatour, L. Lestarquit, and J.-L. Issler, “Cboc: an implementation of mboc,” in *CNES-ESA, 1st Workshop on GALILEO signals and Signal Processing*, 2006.

[18] J. Avila-Rodriguez, G. Hein, S. Wallner, J. Issler, L. Ries, L. Lestarquit, A. de Latour, J. Gode, F. Bastide, T. Pratt *et al.*, “The mboc modulation–a final touch for the galileo frequency and signal plan. insidegnss, 43–58, gibbons media and research,” 2007.

[19] J. Á. Ávila Rodríguez, “On generalized signal waveforms for satellite navigation,” Ph.D. dissertation, München, Univ. der Bundeswehr, Diss., 2008, 2008.

[20] M. L. Psiaki and T. E. Humphreys, “Gnss spoofing and detection,” *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1258–1270, 2016.

[21] N. Vagle, A. Broumandan, and G. Lachapelle, “Multi-antenna gnss and ins/odometer coupling for robust vehicular navigation,” in *Proc. Int. Tech. Symp. Navigat. Timing*, 2017, pp. 4816–4828.

[22] M. L. Psiaki, T. E. Humphreys, A. P. Cerruti, S. P. Powell, and P. M. Kintner, “Tracking l1 c/a and l2c signals through ionospheric scintillations,” in *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*, 2007, pp. 246–268.

[23] A. Lehner and A. Steingass, "A novel channel model for land mobile satellite navigation," in *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, 2005, pp. 2132–2138.

[24] J. M. Tranquilla, J. Carr, and H. M. Al-Rizzo, "Analysis of a choke ring groundplane for multipath control in global positioning system (gps) applications," *IEEE Transactions on antennas and propagation*, vol. 42, no. 7, pp. 905–911, 1994.

[25] R. Zhang, H. Qin, Z. Zhou, and B. Li, "Gnss multipath mitigation algorithm with antenna arrays based on matrix reconstruction," in *2020 IEEE 20th International Conference on Communication Technology (ICCT)*. IEEE, 2020, pp. 881–887.

[26] C. C. Counselman, "Multipath-rejecting gps antennas," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 86–91, 1999.

[27] S. Hehenberger, V. Tripathi, S. Varma, W. Elmarissi, and S. Caizzone, "A miniaturized all-gnss bands antenna array incorporating multipath suppression for robust satellite navigation on uav platforms," in *2021 15th European Conference on Antennas and Propagation (EuCAP)*. IEEE, 2021, pp. 1–4.

[28] S. N. Boyko, A. S. Kukharenko, and Y. S. Yaskin, "Ebg metamaterial ground plane application for gnss antenna multipath mitigating," in *2015 International Workshop on Antenna Technology (iWAT)*. IEEE, 2015, pp. 178–181.

[29] P. Yang, F. Yan, L. Zhou, M. Gao, and F. Yang, "A small low-multipath gnss antenna using annular slot loaded ground plane," in *2016 IEEE 5th Asia-Pacific Conference on Antennas and Propagation (APCAP)*. IEEE, 2016, pp. 349–350.

[30] P. D. Groves, Z. Jiang, B. Skelton, P. A. Cross, L. Lau, Y. Adane, and I. Kale, "Novel multipath mitigation methods using a dual-polarization antenna," in *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010)*, 2010, pp. 140–151.

[31] J. Xu and J. Ding, "Gnss multipath suppression technology based on postcorrelation and independent component analysis," *Plos one*, vol. 17, no. 4, p. e0267216, 2022.

[32] A. Hussain, A. Ahmed, H. Magsi, and R. Tiwari, "Adaptive gnss receiver design for highly dynamic multipath environments," *IEEE Access*, vol. 8, pp. 172 481–172 497, 2020.

[33] W. Qiu, Q. Zeng, R. Xu, J. Liu, J. Shi, and Q. Meng, "A multipath mitigation algorithm for gnss signals based on the steepest descent approach," *Satellite Navigation*, vol. 3, no. 1, p. 14, 2022.

[34] Y. Qi, Z. Yao, and M. Lu, "General design methodology of code multi-correlator discriminator for gnss multi-path mitigation," *IET Radar, Sonar & Navigation*, vol. 15, no. 9, pp. 969–984, 2021.

[35] C. Macabiau, L. Deambrogio, V. Barreau, W. Vigneau, J.-J. Valette, G. Artaud, P. Thevenon, and L. Ries, "Kalman filter based robust gnss signal tracking algorithm in presence of ionospheric scintillations," in *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012)*, 2012, pp. 3420–3434.

[36] I. Cortés, J. R. van der Merwe, E. S. Lohan, J. Nurmi, and W. Felber, "Evaluation of low-complexity adaptive full direct-state kalman filter for robust gnss tracking," *Sensors*, vol. 23, no. 7, p. 3658, 2023.

[37] A. Tabatabaei and M. R. Mosavi, "Robust adaptive joint tracking of gnss signal code phases in urban canyons," *IET radar, sonar & navigation*, vol. 11, no. 6, pp. 987–993, 2017.

[38] E. Shytermeja, "Robust gnss positioning in urban environment," in *Accuracy of GNSS Methods*. IntechOpen, 2018.

[39] E. Munin, A. Blais, and N. Couellan, "Convolutional neural network for multipath detection in gnss receivers," in *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*. IEEE, 2020, pp. 1–10.

[40] L.-T. Hsu, "Gnss multipath detection using a machine learning approach," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[41] P. Geragersian, I. Petrunin, W. Guo, and R. Grech, "Multipath detection from gnss observables using gated recurrent unit," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*. IEEE, 2022, pp. 1–7.

[42] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[43] R. G. Brown and P. Y. Hwang, “Introduction to random signals and applied kalman filtering: with matlab exercises and solutions,” *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*, 1997.

[44] Steingass Alexander and Lehner Andreas , “Software model for satellite to land mobile multipath propagation .” 2019. [Online]. Available: https://www.KN−S.dlr.de/COS−LMS