Ph.D. dissertation

Doctoral Program on Electronic Engineering and Telecommunication

Analysis and Implementation of Anti-Spoofing Detection Methods based on Galileo Signal Authentication Service

Rafael Terris Gallego

Directors/Supervisors: Gonzalo Seco Granados, José A. López Salcedo

Department of Telecommunication and Systems Engineering

School of Engineering

Universitat Autònoma de Barcelona

Bellaterra, September 22th, 2024

Abstract

Global Navigation Satellite System (GNSS) have become, despite their relatively brief history, a fundamental component in many applications. Accurate and reliable positioning and timing are crucial for a wide range of services. However, in recent years, the robustness of these systems against malicious attacks such as spoofing has become a growing concern. The European Galileo program is actively enhancing the resilience of these systems by developing new services. These include Open Service Navigation Message Authentication (OSNMA), which provides data authentication for navigation bits, and Signal Authentication Service (SAS), which offers ranging authentication through the encryption of spreading code chips. Similar strategies are employed in the Chips-Message Robust Authentication (CHIMERA) system for Global Positioning System (GPS).

This thesis deals with SAS, originally name Assisted Commercial Authentication Service (ACAS), which is currently being defined by the Galileo program. This service uses the Timed Efficient Stream Loss-tolerant Authentication (TESLA) keys, provided by the OSNMA on the E1-B signal, to re-encrypt segments of the encrypted E6-C signal. This re-encrypted segments, referred to as Re-Encrypted Code Sequences (RECSs), are made available in the GNSS Service Centre (GSC) prior to the broadcasting of the E6-C signal, so they can be downloaded by a compatible receiver. Once the corresponding key is disclosed in the E1-B signal, the receiver can decrypt the RECS and use it to correlate with the broadcast E6-C signal. A successful correlation peak under specific conditions confirms the signal's authenticity.

In the present dissertation, the SAS is analyzed in detail to address several of the crucial questions that emerge from its definition phase: what differences does it entail compared to a conventional GNSS service, considering that the RECSs are only provided for certain predefined instants? How does the length of these RECSs and their periodicity affect the probability of detecting the correlation peak? What operating modes can be considered? What are the design parameters that have the greatest impact on its performance? What performance can we expect in real-world scenarios? The answers to these questions aim to consolidate the definition of the service parameters but could also serve as a valuable tool for selecting the hardware configuration of an SAS receiver and as a performance benchmark for practical implementations.

The contribution of this thesis is twofold. The first part entails a comprehensive examination of the service from a theoretical perspective, with special emphasis on the acquisition procedure. Initially, a generic approach is introduced, establishing a general framework that uses only the E6-C samples. Subsequently, this approach is specialized to incorporate E1-B samples for obtaining useful estimates for the E6-C band correlation process. This specialized approach, termed as "Nominal Operating Mode", has been adopted as the default approach for SAS by the Galileo program. A series of simulations are carried out to assess the impact of key parameters on service performance, such as the recommended length of the RECSs. In addition, guidelines are provided for the implementation of this new service, along with an identification of potential threats to face off.

In the second part, a novel test platform based on Software Defined Radio (SDR) is introduced to evaluate the signal level performance of SAS using real signals. This platform facilitates the synchronized capture of samples from both E1-B and E6-C signals, a crucial aspect for characterizing the alignment of these signals —an essential factor for the anticipated authentication mechanism in SAS. The results obtained validate the convenience of using estimates from the E1-B signal for ACAS. Furthermore, the platform is used to evaluate the performance of SAS in terms of acquisition-level probability of detection across various RECSs and signal-to-noise ratio configurations.

Resumen

Los sistemas de navegación global por satélite (GNSS) se han convertido, a pesar de su relativamente breve historia, en un componente fundamental en muchas aplicaciones. La precisión y fiabilidad en el posicionamiento y la sincronización son cruciales para una amplia gama de servicios. Sin embargo, en los últimos años, la robustez de estos sistemas frente a ataques malintencionados como el *spoofing* se ha convertido en una preocupación creciente. El programa europeo Galileo está mejorando activamente la resiliencia de estos sistemas mediante el desarrollo de nuevos servicios. Estos incluyen Autenticación del Mensaje de Navegación del Servicio Abierto (OSNMA), que proporciona autenticación de datos para los bits de navegación, y el Servicio de Autenticación de Señal (SAS), que ofrece autenticación de rangos a través de la encriptación de los chips del código de ensanchamiento. Estrategias similares se emplean en el sistema CHIMERA para el GPS.

Esta tesis trata sobre el SAS, originalmente denominado ACAS, que actualmente está siendo definido por el programa Galileo. Este modo utiliza las llaves TESLA, proporcionadas por el OSNMA en la señal E1-B, para reencriptar fragmentos de la señal E6-C encriptada. Estos fragmentos reencriptados, denominados RECS, se suben al Centro de Servicios Galileo (GSC) antes de la emisión de la señal E6-C, para que puedan ser descargados por un receptor compatible. Una vez que se revela la llave correspondiente en la señal E1-B, el receptor puede desencriptar los RECS y usarlos para realizar la correlación con la señal E6-C transmitida. La presencia de un pico de correlación bajo condiciones específicas confirma la autenticidad de la señal.

En la presente disertación se analiza detalladamente el SAS para abordar varias de las preguntas cruciales que surgen de su fase de definición: ¿Qué diferencias supone con respecto a un servicio GNSS convencional, considerando que los RECS solo se proporcionan para ciertos instantes predefinidos? ¿Cómo afectan la longitud de estos RECS y su periodicidad a la probabilidad de detectar el pico de correlación? ¿Qué modos de operación se pueden contemplar? ¿Cuáles son los parámetros de diseño que tienen mayor impacto en su rendimiento? ¿Qué rendimiento podemos esperar en escenarios reales? Las respuestas a estas preguntas tienen como objetivo consolidar la definición de los parámetros del servicio, pero también podrían servir como una herramienta valiosa para seleccionar la configuración de *hardware* de un receptor SAS y como un punto de referencia de sus prestaciones para implementaciones prácticas.

La contribución de esta tesis es doble. En la primera parte se proporciona un examen exhaustivo del servicio desde una perspectiva teórica, haciendo especial hincapié en el procedimiento de adquisición. Inicialmente se presenta un enfoque genérico, que establece un marco general que utiliza solo las muestras de la señal E6-C. Posteriormente, este enfoque se especializa para incorporar muestras de E1-B para obtener estimaciones útiles para el proceso de correlación de la señal E6-C. Este enfoque específico, denominado "Modo de Operación Nominal", ha sido adoptado como el enfoque predeterminado para el SAS por el programa Galileo. Diversas series de simulaciones son realizadas para evaluar el impacto de los parámetros clave en el rendimiento del servicio, como la longitud recomendada de los RECS. Además, se proporcionan pautas para la implementación de este nuevo servicio, junto con una identificación de posibles amenazas con las que lidiar.

En la segunda parte se introduce una nueva plataforma de prueba basada en Radio Definida por Software (SDR) para evaluar el rendimiento a nivel de señal del SAS utilizando señales reales. Esta plataforma habilita la captura sincronizada de muestras de las señales E1-B y E6-C, un aspecto crucial para caracterizar la alineación de estas señales, a su vez un factor esencial para el mecanismo de autenticación en que se basa SAS. Los resultados obtenidos validan la idoneidad de usar estimaciones de la señal E1-B para el SAS. Además, la plataforma se utiliza para evaluar el rendimiento del SAS en términos de probabilidad de detección a nivel de adquisición en varias configuraciones de RECS y relaciones de señal a ruido.

Acknowledgements

Engaging in a doctoral thesis project, while immensely rewarding, is no easy task: it requires great dedication, continuous effort, and overcoming countless obstacles in situations that demand giving your best. It's a project that, although it requires significant individual work, also relies heavily on support to carry it out successfully. Therefore, I want to express my gratitude and dedicate this thesis to the following individuals.

To Gonzalo and José, my thesis supervisors, who gave me the opportunity to fulfill one of my dreams. They opened their doors for me to collaborate in their research group, and it has undoubtedly been one of the best decisions I have made. I have been able to benefit from their vast experience in signal processing and satellite navigation, as well as their valuable advice; I hope to repay them someday. To Nacho, responsible for the projects that have made the research conducted in this thesis possible.

To my old university friends, Andreu, Pere, Pasqui, and José, with whom I have maintained a great friendship over the years, sharing countless dinners during the Miercolins in Barcelona; to Marta, with whom I studied side by side during our five years of college, and whose family, Celia, Fernando, and Anna, welcomed me as one of their own during my time in Barcelona.

To my former colleagues at Indra, Pau and Marcos, with whom I shared so many projects and moments during the nearly ten years I worked with them. Together, we shared many experiences that have allowed us to forge a friendship that transcends beyond the workplace.

To Manuel, partner and colleague at Albedo, with whom I have suffered but also learned to manage and overcome the challenges that come with any new project. He and Valentín welcomed me with open arms and also gave me the opportunity to travel around the world, surely one of the best experiences of my life. We'll always have Japan.

To my lifelong friends, Claudi, Clàudia, Xavi, Pame, Clara, Neri, and Anna, who have accompanied me all these years, and whom I hope will continue to accompany me in the years to come; to Leire, life partner, who has made me the best person I could be.

To my father Manuel and mother Annie, who would have loved to still be here, and to feel proud of everything they did for me; to my brother Felipe and Ester, with whom I will be able to share this achievement.

And finally, to my wife Núria and my son Nüc, who are the most important things I have had in my life, and who have given me the strength and love necessary to reach the end.

Rafael Terris
September 22th, 2024

Agradecimientos

Involucrase en un proyecto de tesis doctoral, a pesar de ser sumamente gratificante, no es una tarea fácil: requiere de una gran dedicación, un esfuerzo continuo y superar incontables obstáculos en situaciones que requieren dar lo mejor de sí mismo. Es un proyecto que, aunque precise de un importante trabajo individual, necesita en todo momento de un gran acompañamiento para llevarlo a cabo con éxito. Por ello quiero agradecer y dedicar esta tesis a las siguientes personas.

A Gonzalo y José Antonio, mis directores de tesis, que me dieron la oportunidad de cumplir uno de mis sueños. Me abrieron sus puertas para colaborar en su grupo de investigación y, sin duda alguna, ha sido una de las mejores decisiones que he tomado. He podido aprovecharme de su enorme experiencia en el campo del procesado del señal y la navegación por satélite, así como de sus valiosos consejos; espero poderles devolver esa deuda algún día. A Nacho, responsable de los proyectos que han hecho posibles las investigaciones realizadas en esta tesis.

A mis antiguos compañeros de universidad, a Andreu, Pere y Pasqui, además de José Antonio, con lo que he podido mantener una gran relación de amistad a lo largo de todos estos años, y compartir incontables cenas durante los Miercolins en Barcelona; a Marta, con la que estudié codo a codo durante los cinco años de la carrera, y cuya familia, Celia, Fernando y Anna, me acogió como un miembro más durante mi estancia en Barcelona.

A mis antiguos compañeros de Indra, Pau y Marcos, con los que compartí tantos proyectos y momentos durante los casi diez años que trabajé con ellos. Juntos compartimos muchas experiencias que nos han permitido forjar una amistad que ha trascendido mucho más allá de lo laboral.

A Manuel, socio y compañero de Albedo, con él que he sufrido pero también aprendido a gestionar y sobrellevar las dificultades que entraña todo nuevo proyecto. Él y Valentín me acogieron con los brazos abiertos, y me dieron también la oportunidad de viajar alrededor del mundo, seguramente una de las mejores experiencias de mi vida. Siempre nos quedará Japón.

A mis amigos de toda la vida, a Claudi, Clàudia, Xavi, Pame, Clara, Neri y Anna, que me han acompañado todos estos años, y que espero que me acompañen todos los que tengan que venir; a Leire, compañera de vida, que ha hecho de mí la mejor persona que podía ser.

A mi padre Manuel y madre Annie, a los que les hubiera encantado estar todavía aquí, y sentirse orgulloso de todo lo que hicieron por mí; a mi hermano Felipe y a Ester, con los que sí podré compartirlo.

Y finalmente a mi mujer Núria y mi hijo Nüc, que son lo más importante que he tenido en mi vida, y que me han dado la fuerza y amor necesarios para llegar al final.

Rafael Terris
22 de septiembre, 2024

Contents

List of Figures

List of Tables

Listings

Acronyms

ACAS Assisted Commercial Authentication Service

ACF Autocorrelation Function

ADC Analog to Digital Conversion

AER Authentication Error Rate

AES (Advanced Encryption Standard

AGC Automatic Gain Control

A-GNSS Assisted GNSS

API Application Programming Interface

AUC Area Under the Curve

AVAR Allan Variance

AWGN Additive White Gaussian Noise

BDS BeiDou Navigation Satellite System

BGD Broadcast Group Delay

BOC Binary Offset Carrier

BPSK Binary Phase Shift Keying

CAF Cross Ambiguity Function

CAS Commercial Authentication Service

CBC Cipher Block Chaining

CBOC Composite Binary Offset Carrier

CDF Cumulative Distribution Function

CDMA Code Division Multiple Access

CHIMERA Chips-Message Robust Authentication

CLI Command Line Interface

COTS Commercial Off-The-Self

CRB Cramer-Rao Bound

CRLB Cramer-Rao Lower Bound

CRPA Control Reception Pattern Antenna

CS Commercial Service

CSAC Chip Scale Atomic Clock

CSV Comma-Separated Values

DLL Delay-Lock-Loop

DSM Digital Signature Message

DSP Digital Signal Processing

DSSS Direct-Sequence Spread Spectrum

DVB-T Digital Video Broadcasting – Terrestrial

EC European Commission

ECS Encrypted Code Sequence

$\mathbf{ECS^R}$ Received Encrypted Code Sequence

FDE Fault Detection and Exclusion

FDMA Frequency Division Multiple Access

FFT Fast Fourier Transform

FN False Negative

FP False Positive

FPGA Field Programmable Gate Array

FPR False Positive Rate

G2G Galileo Second Generation

GEO Geostationary Earth Orbit

GNSS Global Navigation Satellite System

GPS Global Positioning System

GSC GNSS Service Centre

GST Galileo System Time

HAS High Accuracy Service

HS-GNSS High Sensitivity GNSS

ICD Interface Control Document

IFFT Inverse Fast Fourier Transform

IGSO Inclined Geosynchronous Orbit

IMU Inertial Measurement Unit

INS Inertial Navigation System

IOT Internet of Things

KDI Key Delay Index

LBS Location-Based Service

LE Logic Element

LFSR Linear Feedback Shift Register

LHCP Left-Hand Circular Polarization

LMS Land-Mobile Satellite

LOS Line-of-Sight

M2M Machine-to-Machine

MAC Message Authentication Code

MACK Message Authentication Code and Key

MBOC Multiplexed Binary Offset Carrier

MCS Master control station

MEO Medium Earth Orbit

MFMC Multi-Frequency Multi-Constellation

MIMO Multiple-Input Multiple-Output

MLE Maximum Likelihood Estimation

MMARIO Message and Measurement Authentication Receiver for Initial Operations

MPD Multi-Peak Detection

MSB Most Significant Bit

NCO Numerical Controlled Oscillator

NMA Navigation Message Authentication

NNSS Navy Navigation Satellite System

OCXO Oven-Controlled Crystal Oscillator

OFB Output Feedback

OS Open Service

OSNMA Open Service Navigation Message Authentication

PAM Pulse Amplitude Modulation

PAULA Precise and Authentic User Location Analysis

PCS Parallel Code Phase Search

PD Probability of Detection

pdf Probability Density Function

PFA Probability of False Alarm

PLL Phase-Lock-Loop

PPP Precise Point Positioning

PPSP Primary Peak to Secondary Peak

PRN Pseudo-Random Noise

PRS Galileo Public Regulated Service

PSK Phase Shift Keying

PVT Position, Velocity and Timing

QP Quasi-Pilot

RAIM Receiver Autonomous Integrity Monitoring

RAND Randomization flag

RECS Re-Encrypted Code Sequence

RF Radio-Frequency

RFI Radio-Frequency Interference

RHCP Right-Hand Circular Polarization

RNSS Regional Navigation Satellite System

ROC Receiver Operating Characteristic

RTK Real-Time Kinematic

SAS Signal Authentication Service

SBAS Satellite Based Augmentation System

SCA Spreading Code Authentication

SCE Spreading Code Encryption

SCER Security Code Estimation and Replay

SCI Secondary Code Index

SDR Software Defined Radio

SHA Secure Hash Algorithm

SIS Signal-In-Space

SNR Signal to Noise power Ratio

SPS Standard Positioning Service

SQM Signal Quality Monitoring

SS Spread Spectrum

SV Space Vehicle

TBA Time Between Authentications

TCXO Temperature-Compensated Crystal Oscillator

TEC Total Electron Content

TESLA Timed Efficient Stream Loss-tolerant Authentication

TN True Negative

TNR True Negative Rate

TOA Time-Of-Arrival

TP True Positive

TPR True Positive Rate

UAV Unmanned Aerial Vehicle

USB Universal Serial Bus

VCTCXO Voltage-Controlled Temperature-Compensated Crystal Oscillator

VSS Vestigial Signal Search

Notation

In the sequel, matrices are denoted by uppercase boldface letters, vectors are denoted by lowercase boldface letters, and scalars are denoted by italics letters. Other specific notation has been described as follows.

$\mathbf{A}^*, \mathbf{A}^{\mathrm{T}}, \mathbf{A}^{\mathrm{H}}, \mathbf{A}^{-1}$	Complex conjugate, transpose, conjugate transpose (Hermitian), and inverse of matrix $\mathbf{A}$, respectively.
$\|\mathbf{a}\|$	Euclidean norm of vector $\mathbf{a}$ (i.e. $\|\mathbf{a}\| = \sqrt{\mathbf{a}^H\mathbf{a}}$).
$\|a\|$	Absolute value of scalar a.
$\widehat{a}$	Estimate of a.
$\mathrm{Re}\{\cdot\}, \mathrm{Im}\{\cdot\}$	Real and imaginary part operators.
$\max\{a, b\}$	Maximum between a and b.
$\min\{a, b\}$	Minimum between a and b.
$\max_i\{\cdot\}$	Maximum among i.
$\min_i\{\cdot\}$	Minimum among i.
$\log_N(\cdot)$	Logarithm to the base N.
$\ln(\cdot)$	Natural logarithm (i.e. logarithm to the base e).
$\exp\{x\}$	Exponential function (i.e. e^x).
$\neq$	Not equal to.
$\approx$	Approximately equal to.
$\sim$	Follows a given distribution.
$\doteq$	Defined as.
$\mathbb{R}, \mathbb{C}$	Set of real and complex numbers, respectively.
$\mathrm{E}[\cdot]$	Statistical expectation.
$\mathcal{N}\left(\mu, \sigma^2\right)$	Normal (i.e. Gaussian) distribution with mean μ and variance σ^2.
δ_{ij}	Kronecker delta (i.e. $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise).
$\partial(\cdot)/\partial(x)$	Partial derivative with respect x.

Chapter 1

Introduction

The flourishing market for GNSS is a reflection of their critical role in a myriad of applications, spanning from everyday navigation aids in smartphones and vehicles to sophisticated uses in geodesy, agriculture, and defense. However, the ubiquity and reliance on GNSS for such a wide array of critical functions have highlighted its vulnerabilities, particularly in the face of malicious threats such as spoofing and masking. These vulnerabilities pose risks to personal navigation, but also threaten commercial aviation or autonomous vehicles.

In response to these challenges, there has been a clear focus on enhancing the robustness of GNSS services through cryptographic protections. Authentication services, such as those currently being developed within the European Galileo program, represent a substantial advancement in this direction. The development and integration of such cryptographic mechanisms and authentication services are crucial to maintaining the confidence and reliability that countless users place in GNSS technology on a daily basis.

1.1 Motivation and Objectives

The primary objective of this thesis is to analyze SAS, a new service designed to improve Galileo robustness through the encryption of spreading codes. The author, in collaboration with other members of the GNSS community, has been involved in the definition of this service since the European Commission (EC) unveiled the *Precise and Authentic*

User Location Analysis (PAULA) project. This project, aimed at strengthening the development of High Accuracy Service (HAS), also sought to conduct a feasibility study of SAS, with plans to start the first tests by 2024. Building on this primary objective, the current dissertation concentrates on the following research lines.

- Develop a model to assess the impact of critical parameters defined within the service, focusing on acquisition processes..
- Measure the performance of the key parameters of the service in terms of detection probability.
- Offer recommendations for optimal values for the key parameters of the service.
- Identify and analyze the primary threats to the service, providing insights for future enhancements.
- Implement a hardware platform to evaluate the service's performance using real-word signals.

1.2 Thesis Outline and Publications

This section offers a concise overview of the chapters that make up this dissertation. Additionally, the work presented in this thesis has resulted in several publications in international conferences and journals, which are listed below.

Chapter 2 presents the fundamentals of GNSS technologies. It begins by focusing on the basic concepts of satellite navigation systems, including the positioning principle, underlying architecture, the signals used, and the characteristics of receivers. This section places special emphasis on the signal acquisition and detection process, as it is one of the key aspects that will be analyzed in the SAS for Galileo. The latter half of the chapter provides a brief history of the authentication techniques used in GNSS. This serves to justify the need for a service like SAS to provide robust and resilient navigation solutions.

Chapter 3 describes the SAS, from the general concept to its specific implementation for Galileo. As the service relies on the TESLA keys provided by OSNMA, an overview of the latter is also included. The RECS files, which contain the re-encrypted segments of the E6-C signal for use in the correlation process, are described in detail. The key parameters of the SAS are outlined, along with a brief description of the cryptographic operations involved. Finally, the mechanisms used for the authentication process are examined. The results of this chapter have been published in the following international journal paper:

I. Fernandez-Hernandez, J. Winkel, C. O'Driscoll, S. Cancela, R. Terris-Gallego, *et al.*, "Semiassisted Signal Authentication for Galileo: Proof of Concept and Results," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 4393-4404, Aug. 2023.

Additionally, the latest specifications of SAS have been presented (and are set to be published) at the following international conference:

I. Fernandez-Hernandez, J. Winkel, C. O'Driscoll, G. Caparra, R. Terris-Gallego, *et al.*, "Galileo Signal Authentication Service (SAS)," in *Proceedings of the 37th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2024)*, Baltimore, Maryland, Sep. 2024, pp. 3292 –3307. [Online]. Available: `https://www.ion.org/publications/abstract.cfm?articleID=19707`.

Chapter 4 proposes guidelines for the implementation of Galileo SAS receivers, focusing on signal-level processing. In SAS, only fragments of the signal are provided, leading to significant differences compared to a conventional GNSS receiver. These differences are highlighted and a general acquisition model for SAS is proposed. Indeed, the receiver clock uncertainty plays a major role in this process, along with other key service parameters, such as the RECS length and periodicity. To analyze the impact of these parameters in terms of the probability of detection, a MATLAB simulator is used, which is described here. The results of this chapter have been published in the following international conference paper:

R. Terris-Gallego, I. Fernandez-Hernandez, J. A. López-Salcedo, and G. Seco-Granados, "Guidelines for Galileo Assisted Commercial Authentication Service Implementation," in *Proceedings of the International Conference on Localization and GNSS (ICL GNSS)*, Tampere, Jun. 2022.

Chapter 5 presents the default mode retained for SAS, referred to as the "Nominal Operating Model". This mode aims to mitigate the impact of receiver clock uncertainty on the acquisition process by using the E1-B signal. Indeed, the Position, Velocity and Timing (PVT) from E1-B can provide a time reference when the receiver clock is unreliable, significantly reducing the overall uncertainty in the acquisition process. This reduction results in an improved probability of detection, as the simulations provided show. A summary of the different operating modes for SAS is also provided, along with an analysis of the potential threats that a receiver might encounter. The outcomes of this chapter have been published in the following international conference paper:

R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, and I. Fernandez-Hernandez, "Operating Modes and Performance Evaluation of Galileo Assisted Commercial Authentication Service," in *Institute of Navigation Conference (ION+ GNSS 2022)*, 2022.

Chapter 6 introduces a novel experimental test platform based on SDR for evaluating SAS. This platform complements theoretical simulations with results obtained from real-world scenarios. The various components of the platform are described here along with its configuration. The platform includes two low-cost bladeRF boards, which enable the synchronous capture of E1-B and E6-C samples. The alignment of both bands is crucial for the "Nominal Operating Mode" of SAS, characterized by the use of real datasets obtained in different scenarios. These datasets, described herein, are also used to evaluate the performance of SAS under various configurations. Additionally, an efficient detection using other Galileo signal like E6-B is also analyzed. The findings of this chapter have been published in the following international conference papers:

R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, and I. Fernandez-Hernandez, "E1-E6 SDR platform based on bladeRF for testing Galileo Assisted Commercial Authentication Service," in *European Navigation Conference, ESA ESTEC*, Noordwijk, The Netherlands: MDPI, 2023.

R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, and I. Fernandez-Hernandez, "Preliminary Evaluation of Galileo ACAS using Existing E1-E6 Open Signals and a Low-Cost SDR Platform," in *Institute of Navigation Conference (ION+ GNSS 2023)*, Denver, Sep. 2023.

R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, and I. Fernandez-Hernandez, "Efficient Detection of Galileo ACAS Sequences using E6-B Aiding," in *European Navigation Conference, ESA ESTEC*, Noordwijk, The Netherlands: MDPI, 2024.

1.3 Other Contributions

Within the framework of this thesis, several contributions have been made in the context of the EC-funded PAULA project, referred as "DEFIS/2020/OP/0002: Test Platform on Galileo HAS/CAS/OSNMA" [8], including:

- UAB, "PAULA Research Report: Detailed Analysis On Assisted Commercial Authentication Service," Tech. Rep., 2023
- GMV, "PAULA Final Report," Tech. Rep., 2023. [Online]. Available: `https://etendering.ted.europa.eu/cft/cft-document.html?docId=155981`

Finally, the author is involved in the follow-up project of PAULA, known as Message and Measurement Authentication Receiver for Initial Operations (MMARIO), referred to as "DEFIS/2023/OP/0011: Contribution to Radio-navigation Accuracy and Resilience" [11]. This EC-funded project aims to build the first operational platform for SAS.

Chapter 2

A Short Technical Background in GNSS Authentication

2.1 Introduction

Determining the geographical position has been a challenging task in human history. From ancient navigation methods that relied on celestial observations [12], a remarkable journey has been undertaken, which has led to the use of radio frequency signals emitted by orbiting satellites. Known as Global Navigation Satellite Systems (GNSSs), they have revolutionized positioning and timing applications and provide unparalleled accuracy, especially in open-sky environments. This has been facilitated by their technological maturity, coupled with cost-effectiveness and global coverage. Today, GNSSs are an essential tool in contemporary society: more than 10 billion devices are expected to be used around the world by the beginning of the next decade [13].

The origins of satellite navigation systems date back to the early 1960s, when the United States launched the Transit military system, also known as NAVSAT or Navy Navigation Satellite System (NNSS) [14]. The positioning principle was based on the finding that the Doppler shift could establish the location of the satellite in relation to the receiver station. However, the first GNSS arrived in the 1970s with the GPS, also for American military purposes. Although it took more than 20 years to become fully operational, the GPS paved the way for widespread civilian and commercial use, encouraging the development of other GNSS technologies, like Russia's GLONASS, China's Compass-BeiDou, or its European counterpart, known as Galileo. These global navigation systems coexist with the so-called Regional Navigation Satellite System (RNSS), that provide only coverage for certain regions (e.g. India's NAVIC or Japan's QZSS).

The fundamental principle underlying these technologies involves the reception and processing of signals from multiple satellites to determine the receiver's position, mainly based on the estimation of the time of arrival. Today, GNSS technologies underpin a myriad of applications, from transportation to critical Location-Based Services (LBSs), and are integral to the growing fields of the Internet of Things (IOT), Machine-to-Machine (M2M) communications, or Smart Cities, boosted by cloud-based GNSS receivers [15]. However, the ubiquity of GNSS has raised concerns within the GNSS community regarding its vulnerability to malicious attacks. The advent of advanced yet affordable technologies has facilitated these concerns. As a result, intensive research efforts have been carried out in recent years to develop solutions that provide increased protection and robustness to GNSS.

The first part of this chapter (Section 2.2) is dedicated to introducing the fundamentals of GNSS. It begins with an overview of the architecture and signals involved in GNSS, and concludes with a discussion of key aspects of GNSS receivers, with a particular focus on signal detection. The second part of the chapter (Section 2.3) provides a brief history of GNSS authentication mechanisms. It offers a summary of the most common techniques employed both at the receiver and the system sides.

2.2 Fundamentals of GNSS

The primary objective of GNSS is to determine the user's PVT solution based on the signals transmitted by satellites. The fundamental positioning principle is illustrated in Section 2.2.3. Section 2.2.1 delves into the architecture and components of GNSS, while Section 2.2.2 examines the characteristics of the signals emitted by satellites. Section 2.2.4 provides an overview of GNSS receivers, while Section 2.2.7 describes the characteristics of the receiver clock. Finally, Section 2.2.5 discusses the fundamentals of the detection theory applied in GNSS, including a concise introduction to ROC curves, which are elaborated in more detail in Section 2.2.6.

The discussion presented in this section relies primarily on references [16], [17], [18], [19], [20], [21] and [22], to which the interested reader is encouraged to refer for a more in-depth explanation.

2.2.1 GNSS Architecture

The deployment of any GNSS is typically divided into three complementary segments: spatial, ground control, and user. These are described below and are depicted in Figure 2.1.

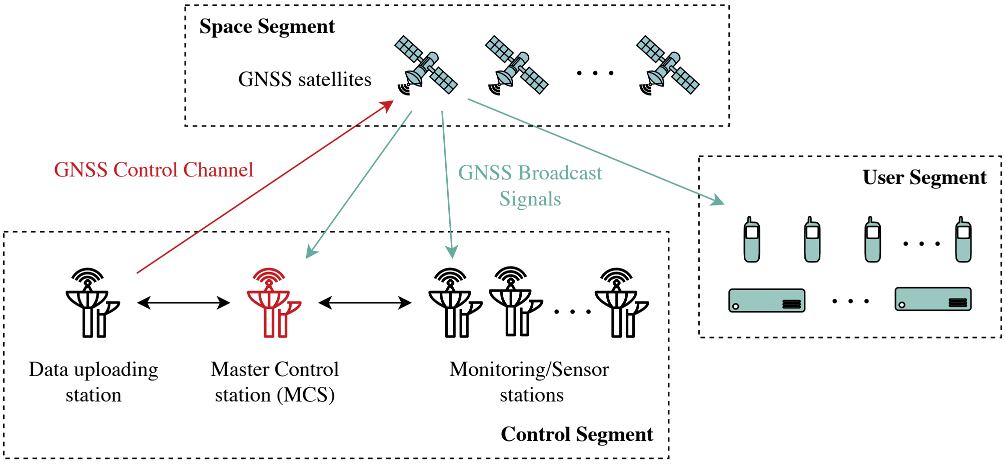


Figure 2.1: GNSS architecture.

Space Segment

The Space Segment consists of a constellation of GNSS satellites orbiting the Earth that broadcast signals with the information required to compute the navigation solution, including its time, orbit, and status, but also clock and atmospheric corrections that may have an impact on signal transmission. These signals are also known as ranging signals, since they are used to compute the pseudoranges.

Each GNSS has its own constellation of satellites, and each satellite, commonly referred to as the Space Vehicle (SV), is uniquely identified by its own identifier. They are placed in such a manner that a minimum of four satellites are in the receiver's Line-of-Sight (LOS) at all times, providing a worldwide coverage. The main characteristics of the four main GNSS in operation at the time of this writing are provided in Table 2.1.

Constellations	**GPS**	**Galileo**	**GLONASS**	**BeiDou**
Country/Region	United States	Europe	Russia	China
Orbit	MEO	MEO	MEO	MEO, IGSO, GEO
Altitude	20180 km	23220 km	19100 km	21530-35790 km
Orbital planes	6	6	3	3
Nominal satellites	24	24	24	30

Table 2.1: Space segment for currently operating GNSSs.

It is worth mentioning that of the 30 satellites that make up the BeiDou system, three satellites orbit at Geostationary Earth Orbit (GEO), and three satellites orbit at Inclined Geosynchronous Orbit (IGSO), providing coverage for specific areas of China and its surroundings. All the remaining BeiDou satellites, as well as the satellites of other GNSS systems, orbit at Medium Earth Orbit (MEO).

Figure 2.2 illustrates the expandable 24-slot GPS satellite constellation, as defined in the Standard Positioning Service (SPS) Performance Standard. It is designed to maintain the availability of at least 24 operational GPS satellites for 95% of the time. To achieve this, up to 31 operational GPS satellites have been in orbit [23].

Ground Control Segment

The Ground Control Segment is responsible for monitoring and controlling the GNSS constellation health. It is also in charge of updating the information encapsulated in the navigation messages when required, including satellite clock, almanac and ephemerides corrections. The ephemerides encompass the entire set of parameters necessary to determine thePVT solution, while the almanac represents a subset of the ephemeris parameters with reduced precision.

To perform these tasks, the Ground Control Segment encompasses the following components: the Master control station (MCS), the monitoring stations and the communication stations.

- The MCS analyzes the signals and data received from the monitoring stations, subsequently sending correction data to the communication stations.

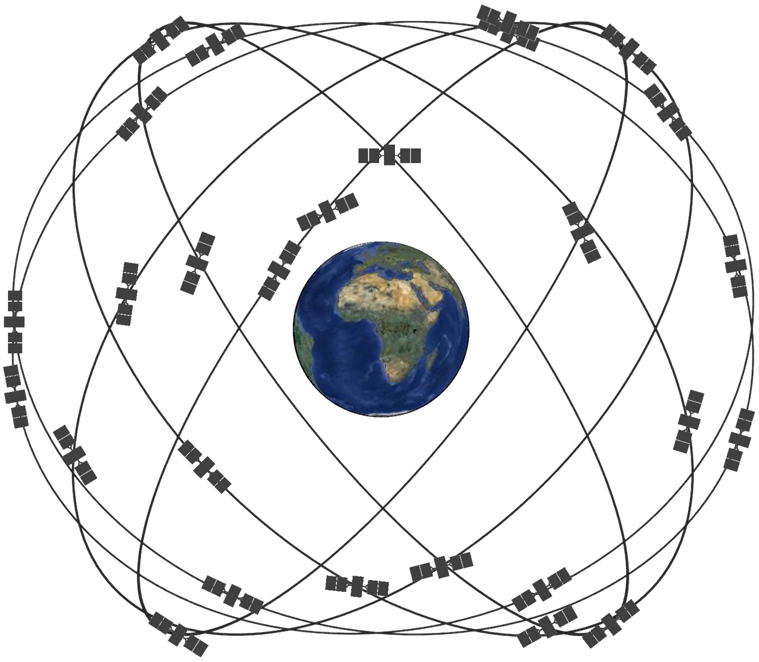


Figure 2.2: GPS expandable 24-slot constellation layout (US Government)

- The monitoring or sensor stations, which monitor the satellite signals and status, along with local meteorological data, relay this information to the MCS. These stations are spread around the world to optimize coverage and their locations are precisely known by the GNSS.
- The communication or data transmission/upload stations facilitate the transmission of data provided by the MCS to the satellites.

User Segment

The User Segment consists of receiver equipment that processes downlink signals received from the GNSS satellites to compute the PVT. This includes all GNSS receivers used in ground, air, marine, and space applications. Depending on the envisioned application, receiver cost and functionality can vary significantly, ranging from small, cost-effective chips embedded in smartphones to rack-sized, high-precision expensive platforms dedicated to providing high accuracy timing.

Thus, GNSS receivers can be categorized based on various criteria, such as form factor type (handheld, smartphones, specialized, etc.), processed frequency bands (single frequency, dual frequency, multiple frequency), or intended use (civilian, open, public-regulated, commercial, military).

2.2.2 GNSS Signals

The design of GNSS has to fulfill numerous requirements to operate correctly, including the provision of positioning accuracy, robustness against common impairment such as interference, multipath effects or noise disruption. The adoption of Spread Spectrum (SS) techniques for GNSS signals can cope with these requirements, as well as providing enhanced security, since the signal's predictability is reduced, making it less susceptible to interception and exploitation.

Spread spectrum techniques are characterized for transmitting a signal across a wider bandwidth than the one occupied by the original generated signal. This spread enhances the security and resistance to interference in communication, which is especially important in military communications, that have widely adopted this techniques.

Of the many flavors of the spread spectrum techniques, GNSS make use of the Direct-Sequence Spread Spectrum (DSSS), In this technique, the original data signal is multiplied by a much faster sequence, which is typically binary. The bits that compose this sequence, known as spreading code, are termed as chips, as they do not carry any useful information. As this spreading code has a much higher rate than the data itself, the resulting signal spreads over a larger bandwidth. Consequently, the signal appears as a low-power wide-band signal, barely distinguishable from floor noise to unintended recipients, as illustrated in Figure 2.3.

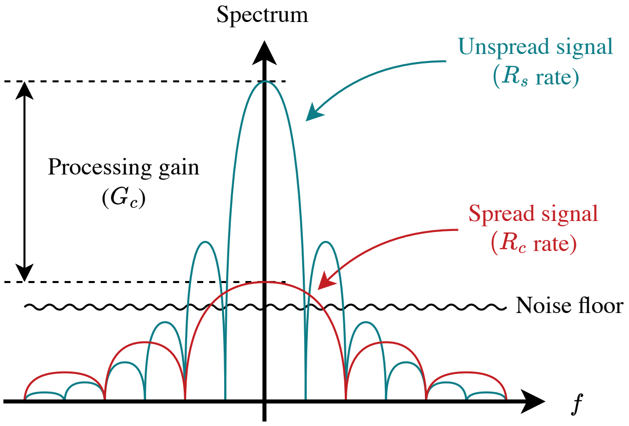


Figure 2.3: Illustration of the processing gain achieved in spread spectrum techniques.

The ratio between the rate of the spreading code and the rate of the original signal is called the *spreading factor* or *processing gain*, ranging typically from 10 to 60 dB:

$$G_p = \frac{R_c}{R_s} \tag{2.1}$$

where R_s is the data symbol rate and R_c the spreading code chip rate.

With the correct spreading code, the receiver can recover the signal back to its original form, benefiting from this gain if the correlation is successful. The higher the spreading gain, the greater the robustness against interference, especially in the case of narrowband interference.

The GNSS signal is build by concatenating continuously such spreading codes to form a unique binary sequence. As each GNSS satellite is generally associated to a specific spreading code, this serves as a unique fingerprint to identify it. This enables the ability to dissociate it among the different satellites. To facilitate this dissociation, the spreading codes are chosen to be as maximum as orthogonal as possible. This is accomplished by using cyclic Pseudo-Random Noise (PRN) sequence, with correlation properties that resembles to white noise: the autocorrelation of a given sequence results in distinguishable peak when aligned with itself, and practically zero otherwise, while the crosscorrelation between to different sequences (corresponding to distinct satellites) is (nearly) zero. This allows detecting the presence of the satellites individually and measure the different propagation times to the receiver, which is the principle used for positioning, as depicted in Section 2.2.3.

Thanks to DSSS, all GNSS signals share the same transmission medium at the same time and frequency. This type of communication is known as Code Division Multiple Access (CDMA) and is extensively used in wireless mobile communications systems. GPS, Galileo and BeiDou rely on CDMA, but it is not the case of GLONASS, which adopts Frequency Division Multiple Access (FDMA) instead, while uses the same spreading code for all its satellites.

The GNSS signals are characterized by the following four main elements: the modulation used to shape the chip pulse; the sequence of chips chosen as the spreading or ranging code, also known as the primary code; the type of data symbols modulated onto this primary code; and the carrier frequency chosen to transmit the signal through the propagation path. The most relevant aspects of these elements are described next.

Chip modulation

Most of the legacy GNSS signals make use of Binary Phase Shift Keying (BPSK) modulation to shape the chip pulse, which corresponds to a rectangular pulse. This simple modulation scheme allows an easy signal processing at the receiver, but modernized signals have leveraged the technological advances to adopt more refined schemes, like Binary Offset Carrier (BOC) modulation. Instead of using a simple pulse for each chip, each chip is multiplied by a binary sequence with frequency equal to or higher than the chip rate. This results in a more efficient bandwidth use and better positioning accuracy, at the expense of increased complexity at the receiver.

The BOC modulation is further detailed in Section 3.3.1. Currently, BOC signals are used, together with BPSK, in GPS, Galileo and BeiDou, while GLONASS is restricted to BPSK-only schemes.

Ranging code

The spreading code used for ranging measurement is the cornerstone of the performance of a given GNSS signal. They are chosen to optimize its correlation properties, which plays a pivotal role in the peak identification and such, in the positioning performance. For instance, in the so-called C/A code of GPS, the Linear Feedback Shift Register (LFSR)-generated Gold codes are used, whereas in other signals like Galileo E6, custom-designed memory codes are employed.

Typical chip rates range from approximately 1 to 10 Mchips per second (Mcps). The higher the rate of the code, the higher the accuracy of the measurement, but the higher the occupied bandwidth and the computational complexity, also. For example, the previously named C/A code of GPS has a period of 1023 chips, for a time duration of 1 millisecond, which results in a chip rate of 1023 Mcps. Longer codes are available in all GNSSs for high precision applications.

Data symbols

As explained in Section 2.2.1, GNSS need to transmit useful data to the receivers in order to compute the PVT solution. The information, including the ephemeris, almanac, status or clock corrections, is encapsulated in the navigation message, which is modulated on top of the chips that compose the primary codes. For example, in GPS C/A, data is transmitted at 50 bps, so each symbol is equivalent to twenty 1-ms primary codes. However, not all the GNSS signals carry this kind of information; instead, the data sequence modulated is known by the receiver. They are known as secondary codes, and combined with the primary codes, allow to create a new much larger sequence that boost the correlation properties. Therefore, depending on the information conveyed by the modulated data symbols, two types of channels can be distinguished: data channels, where the symbols are unknown to the receiver (i.e., navigation message) and pilot channels, where the symbols are known a priori (i.e., the secondary code).

Carrier frequency

GNSS signals are transmitted in L-band, ranging from 1.2 to 1.6 GHz, since they do not suffer from significant attenuation under common weather conditions. Each GNSS signal is allocated in a specific sub-band, with a carrier frequency centered in this band. The bandwidth occupied by the signals depends basically on the chip rate and their chip modulation. Furthermore, some bands are shared among the different systems to allow better interoperability between them, as is the case of GPS L5 and Galileo E5a bands. The spectrum used for the GNSSs is shown in Figure 2.4.

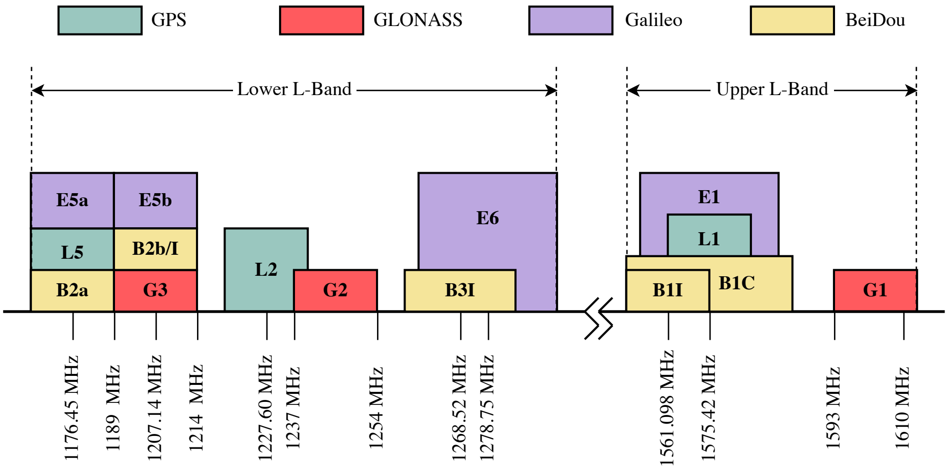


Figure 2.4: Spectrum of GNSS frequency bands.

The resulting carrier-modulated signal transmitted by the GNSS satellites is a band-pass signal, that can be represented as

$$s(t) = \sqrt{2P_t}\, x(t) \cos(2\pi F_c t + \theta_c) \tag{2.2}$$

where P_t is the transmitted power, F_c is the carrier frequency, θ_c is the carrier phase (hereafter assumed to be zero without loss of generality), and $x(t)$ is the binary sequence resulting from the combination of the data symbols (if any) and the chips that conform the spreading codes.

However, thanks to the orthogonality between the sine and cosine, two (real-values) binary sequences are usually transmitted at the same time over a single carrier frequency, as

$$s(t) = \sqrt{2P_{t,I}}\, x_I(t) \cos(2\pi F_c t) - \sqrt{2P_{t,Q}}\, x_Q(t) \sin(2\pi F_c t) \tag{2.3}$$

where $x_I(t)$ and $x_Q(t)$ are referred to as the *in-phase* and *quadrature* components, respectively, and $P_{t,I}$ and $P_{t,Q}$ denote their respective transmitted powers.

This property is leveraged in GNSS by transmitting both the data and pilot channels over the same carrier, as shown in Section 3.3.

The baseband equivalent of (2.3), which contains all the useful information of the GNSS signal, is given by

$$b_s(t) = \sqrt{2P_{t,I}}\, x_I(t) + j\sqrt{2P_{t,Q}}\, x_Q(t) \tag{2.4}$$

Tiered-structure of GNSS signal components

To understand the inner structure of GNSS signals, is useful to interpret each signal component as tiered structure of the constituent elements of which they are composed of, as schematized in Figure 2.5. This approach is based on Chapter 1 of [24].

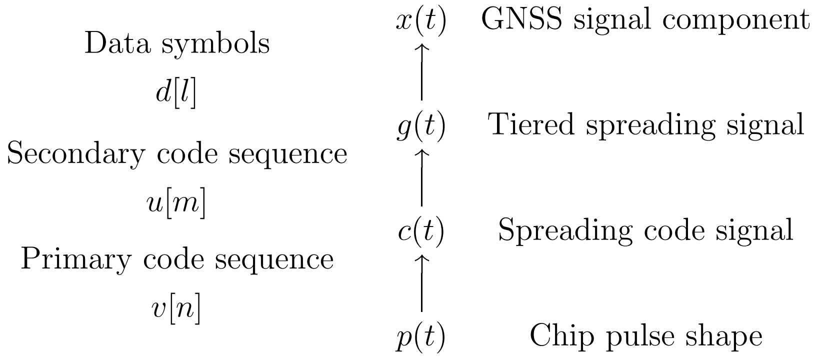


Figure 2.5: Schematic representation of the constituent elements upon which the tiered structure of a GNSS signal component is built.

Any GNSS signal (in-phase or quadrate) component can be expressed as

$$x(t) = \sum_{l=-\infty}^{\infty} d[l]g(t - lT_s) \tag{2.5}$$

where $d[i]$ are the data (discrete-time) symbols, transmitted by the tiered (continuous-time) waveform $g(t)$ whose time duration is defined by its symbol period T_s. Pilot channels transmitting no data symbols can be modeled by considering $d[l] = 1 \; \forall l$ in (2.12).

The tiered symbol waveform $g(t)$ is composed of concatenated spreading code signals modulated by the secondary code sequence, and can be represented as

$$g(t) = \sum_{m=0}^{N_\mathrm{r}-1} u[m]c(t - mT_\mathrm{code}) \tag{2.6}$$

where N_r is the number of primary code repetitions, $\{u[m]\}_{m=0}^{N_r-1}$ is the secondary code sequence, and $c(t)$ is the spreading code signal whose time duration by the primary code period T_code. The absence of secondary codes are modeled by considering $u[m] = 1 \; \forall m$.

The spreading coded signal $c(t)$ turns to be another tiered waveform, which is composed of concatenated chip pulses modulated by primary code sequence. It is given by

$$c(t) = \sum_{n=0}^{N_c-1} v[n]p(t - nT_c) \tag{2.7}$$

where N_code is the number of chips of the primary code, $\{v[m]\}_{n=0}^{N_\mathrm{code}-1}$ is the primary code sequence, and $p(t)$ is the chip pulse whose time duration is defined by its chip period T_c.

The shaping pulse $p(t)$ used in the GNSS signal component depends on the modulation chosen (BPSK or BOC), as depicted in Section 2.2.2. They are given by the following expressions:

$$p_\mathrm{BPSK}(t) = \Pi\left(\frac{t - T_c/2}{T_c}\right) \tag{2.8}$$

$$p_\mathrm{BOC}(t) = \sum_{q=0}^{N_{sc}-1} (-1)^q \Pi\left(\frac{t - qT_{sc}/2 - T_{sc}/4}{T_{sc}/2}\right) \tag{2.9}$$

where T_{sc} is the subcarrier period and $N_{sc} = \frac{T_c}{T_{sc}/2}$ is the number of half subcarrier periods within one chip period (T_c). To ensure that the shaping pulse is causal, the pulse should be delayed by a given amount ($T_c/2$ for BPSK and $T_{sc}/4$ for BOC).

Some examples of BOC pulses are given below, which correspond to the BOC signals used for the Galileo E1, and further analyzed in Section 3.3.1. An illustration of such pulses is given in Figure 2.6.

$$p_{\mathrm{BOC}(1,1)}(t) = \Pi\left(\frac{t - T_c/4}{T_c/2}\right) - \Pi\left(\frac{t - 3T_c/4}{T_c/2}\right) \tag{2.10}$$

$$p_{\mathrm{BOC}(6,1)}(t) = \Pi\left(\frac{t - T_c/24}{T_c/12}\right) - \Pi\left(\frac{t - 3T_c/24}{T_c/12}\right) + \ldots - \Pi\left(\frac{t - 23T_c/24}{T_c/12}\right) \tag{2.11}$$

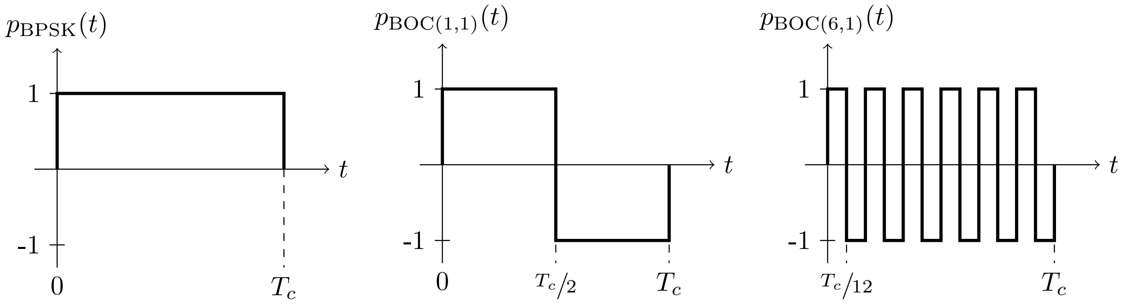


Figure 2.6: BPSK, BOC(1,1) and BOC(6,1) chip pulses.

Finally, we can merge the equations (2.5)-(2.6)-(2.7) and express the GNSS signal component as

$$x(t) = \sum_{l=-\infty}^{\infty} \sum_{m=0}^{N_r-1} \sum_{n=0}^{N_c-1} d[l]u[m]v[n]p(t - nT_c - mT_{\mathrm{code}} - lT_s) \tag{2.12}$$

where it is equivalent to a conventional Pulse Amplitude Modulation (PAM) signal that can expressed as $x(t) = \sum_{n=-\infty}^{\infty} a[n]p(t - nT_c)$, whose values $a[n]$ are a combination of the data symbols and primary and secondary codes.

In Figure 2.7, the composition of the C/A signal of GPS is illustrated. It uses a primary code composed by 1023 chips whose time duration is $T_{\mathrm{code}} = 1$ ms (i.e., $T_c \approx 1$ ms per chip), with a BPSK modulation (i.e., rectangular chip pulses). On top of that, a navigation message is transmitted, whose data symbols (bits in this case) have a duration of $T_s = 20$ ms, that is, $N_r = 20$ primary code repetitions. This corresponds to a symbol rate of $R_s = 50$ bps. Finally, the resulting sequence is multiplied by the sinusoidal carrier (centered in the L1 band, i.e., $f_c = 1575.42$ MHz).

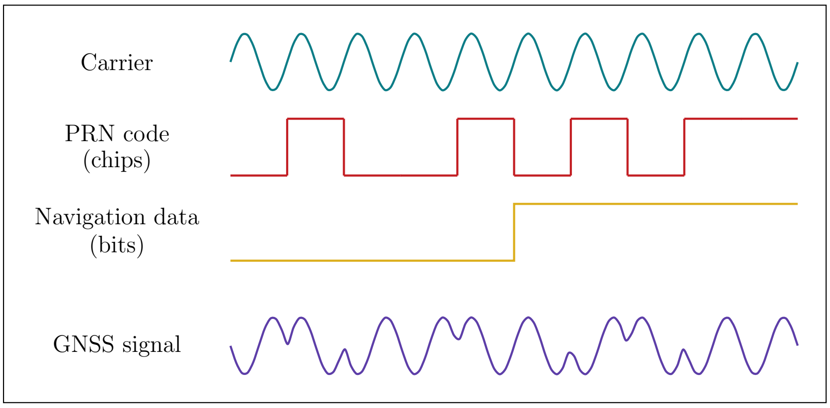


Figure 2.7: Illustration of GNSS signal composition.

2.2.3 GNSS Positioning Principle

From the myriad of techniques used for localization [25], the Time-Of-Arrival (TOA) is the one retained in GNSS [26]. It is based on calculating range estimates by measuring the propagation delay between each visible satellite and the receiver. For this purpose, each satellite transmits a timestamp associated with it, denoted as $t_s^{(k)}$ for the k-th satellite.When the user's receiver captures the signal, it measures the time of reception of the transmitted timestamp, denoted as $t_u^{(k)}$, and calculates the propagation delay for the k-th satellite as

$$\tau^{(k)} = t_u^{(k)} - t_s^{(k)} \tag{2.13}$$

In practice, the receiver obtains the transmit time by aligning a local-generated replica of the transmitted signal with the received signal and reading its internal clock at the very same instant. This principle is illustrated in Figure 2.8.

The distance or range is simply obtained by multiplying the propagation delay by the speed of the electromagnetic waves which are emitted by the satellites (i.e., the speed of light, denoted c):

$$d^{(k)} = c\tau^{(k)} \tag{2.14}$$

Geometrically speaking, the estimated distance $d^{(k)}$ provides the radius of the spherical surface centered at the satellite, which exact position can be calculated from the information provided by the navigation message. This surfaces contains the receiver's

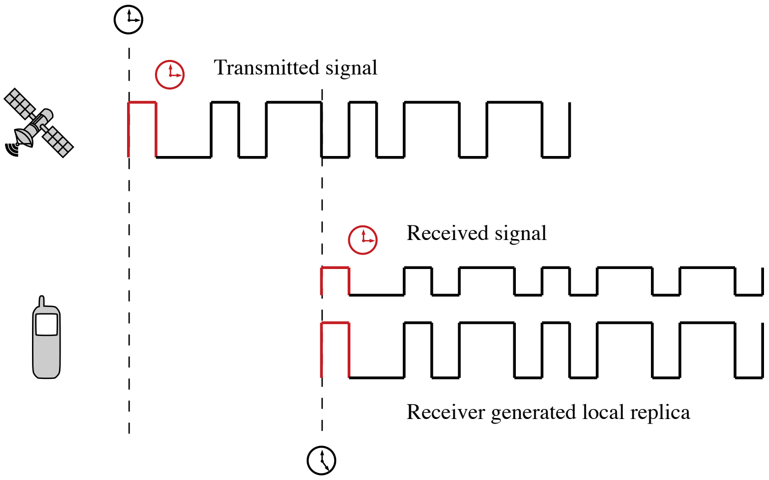


Figure 2.8: Calculation of the time delay propagation in GNSS.

location, so the intersection of a given number of such surfaces will provide the position of the user's receiver. If we assume a in a two-dimensional positioning, we would need three of such surfaces (circumferences in this case). This approach is known as *trilateration*, and it is illustrated in Figure 2.9.

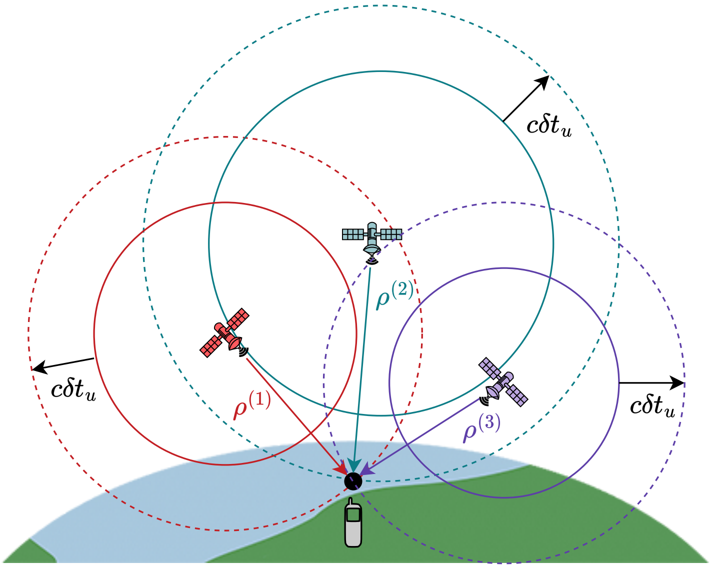


Figure 2.9: GNSS positioning principle and the effect of receiver clock offset (2D positioning).

However, as it can be observed in Figure 2.9, the circumferences represented with solid lines do not intersect. This is due to the fact that the transmit and receive times are measured in different time scales: all the satellite share ideally the same time scale (commonly referred to as the GNSS time, which varies between different constellations), but the receiver depends on its own independent clock. Hence, all ranges measurements are shifted by an unknown term $c\delta t_u$, where δt_u denotes the receiver clock offset, that is, the difference between the receiver and GNSS times. If we account for this offset, the resulting circumferences, represented with dashed lines, intersect in the user's receiver position. The true geometric distance d^k shifted by the receiver clock offset is known as the *pseudoranges*, which is given by

$$\rho^{(k)} = d^{(k)} + c\delta t_u \tag{2.15}$$

This is illustrated in Figure 2.10. The satellite-to-receiver pseudorange is computed as the sum of integer number of spreading codes plus the code delay measurement.

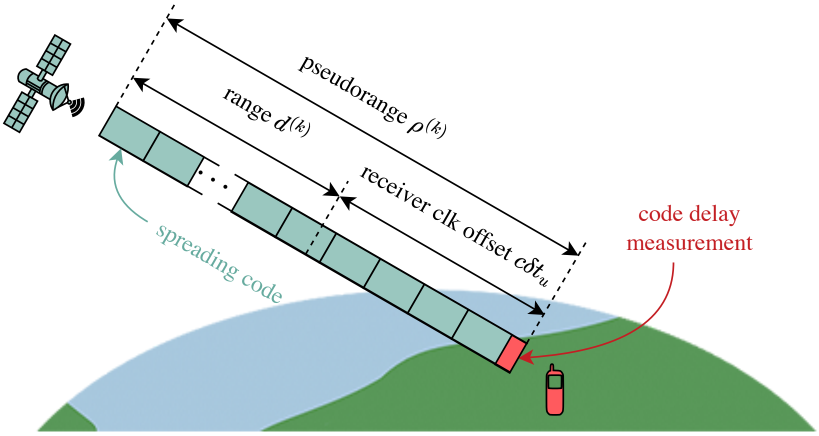


Figure 2.10: Range vs pseudorange in GNSS.

Following the same reasoning as explained for the 2D case, at least four satellites are needed to compute a three-dimensional (3D) position. That is why all GNSS constellations are designed to provide at least this minimum number of satellites at anytime and anywhere. It is worth noting that, when resolving the set of equations that result from the distances measurements, the receiver need also to estimate its clock offset, and consequently it can be synchronized with the specific GNSS time scale.

Nonetheless, the distance measurements $d^{(k)}$ are subject to several disturbing effects that causes the signals transmitted from satellites suffer from time delays other than the propagation-related. This includes relativistic, atmospheric and multipath effects. From these, the ionospheric effects plays a very important role, since it can be very large. However, since the delays introduced by the ionosphere are frequency-dependent, they can be mitigated by using dual frequency receivers. In indoor scenarios, the noise contribution is by far the most relevant issue, and advanced high-sensitivity techniques are required [22]. The contribution of all these sources of time-delay errors is summarized in what is known as the corrected pseudorange, which is given by

$$\rho^{(k)} = d^{(k)} + c\delta t_u + \epsilon^{(k)} \tag{2.16}$$

where ϵ^k accounts for the bias contribution of all the effects that may disturb the signal but cannot be corrected.

In summary, time-delay estimation is, therefore, a crucial aspect in the GNSS for calculating the PVT. However, apart from the pseudorange estimated from the propagation delay (i.e., the code phase delay), GNSS also enable the measurement of the carrier phase. This measurement allows for higher accuracy in estimating the pseudorange, as changes in carrier phase over time also reflect changes in the ranges but are approximately two orders more precise. Indeed, techniques like Real-Time Kinematic (RTK) and Precise Point Positioning (PPP) based on carrier-measurement can provide accuracies in the centimeter-level, compared to the meter-level achieved typically by code-based measurements [27].

2.2.4 GNSS Receivers

GNSS receivers are primarily devoted to determine the user's position, velocity and/or time (PVT) based on received signals coming from the satellites in view. These are typically divided in three differentiated blocks: the front-end, the signal processing module —composed generally by the acquisition module and the tracking module— and the navigation module, which are described below. A schematic of the receiver architecture is presented in Figure 2.11.

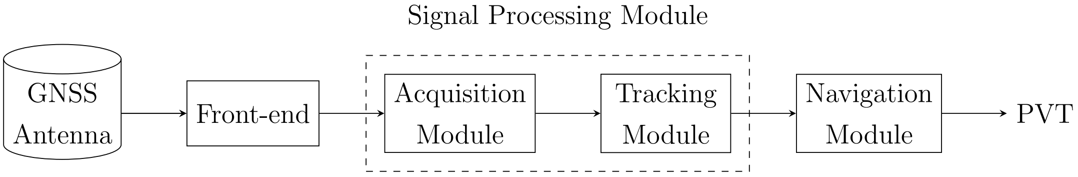


Figure 2.11: Schema of GNSS receiver architecture.

Front-end

The receiver's front-end is a generic block that is in charge of converting the analog bandpass signal captured by the antenna, centered at specific carrier frequency, to a digital baseband signal. This allows the rest of the GNSS receiver to operate exclusively in the digital domain and therefore apply common Digital Signal Processing (DSP) algorithms. To accomplish that, the front-end includes all the typical analog signal conditioning tasks used in communication receivers: band-pass filtering and amplification, base-band conversion, and analog-to-digital conversion.

Due to the nature of the signal propagation in space, the GNSS received signal at the antenna will be affected by three main parameters:

- the time delay (denoted τ) due to the signal propagation from the satellite to the user receiver,
- the Doppler frequency shift (denoted F_d) mainly due to the relative movement between the satellite and the user,
- and the carrier phase (denoted θ), resulting from the combined effects of both time delay and Doppler shift onto the carrier.

Considering the factors mentioned above and the presence of N_sat broadcasting satellites operating within the same frequency band being processed by the receiver, the received passband signal can be modeled as

$$\tilde{r}(t) = \sum_{k=1}^{N_\mathrm{sat}} \sqrt{2P_\mathrm{r}^k}\, x^k(t-\tau^k)\cos(2\pi(F_\mathrm{c}+F_\mathrm{d}^k)t+\theta^k) + \tilde{n}(t) \tag{2.17}$$

where $x(t)$ represents a single GNSS component as defined in (2.5), P_r denotes the received power (accounting for receiver implementation losses), and $\tilde{n}(t)$ is modeled as an Additive White Gaussian Noise (AWGN) noise that includes both background and receiver noise.

Despite the model presented in (2.17) being prevalent in GNSS literature, we consider the next alternative approach [24] to be more convenient, by encompassing all the contribution of the satellites, except the one of interest, into a bandpass noise $\tilde{w}(t)$ that also include the term $\tilde{n}(t)$. This alternative model can be expressed as

$$\tilde{r}(t) = \sqrt{2P_\mathrm{r}}\, x(t-\tau)\cos(2\pi(F_\mathrm{c}+F_\mathrm{d})t+\theta) + \tilde{w}(t) \tag{2.18}$$

The simplified model highlights the key parameters while maintaining a more simpler notation. However, as the model presented in (2.17), does not account for the compression or expansion effect on the signal component $x(t)$ caused by the Doppler shift, commonly referred to as code Doppler [24]. In general, this effect can be neglected in GNSS due to the high carrier frequencies used, which significantly diminish these effects. The model can be further refined by incorporating additional disturbances such as ionospheric scintillation, multipath effects, or interference from other RF signals. However, these factors are not considered here for the sake of simplicity.

Once the received GNSS signal has been adequately conditioned and down-converted by the front-end, it can be generally expressed in baseband form as

$$r(t) = \sqrt{P_r}\, x(t-\tau) e^{j(2\pi F_d t + \theta)} + w(t) \tag{2.19}$$

where both $r(t)$ and $w(t)$ are now complex-valued signals.

Finally, the discrete-time equivalent of the signal, resulting from the output of the Analog to Digital Conversion (ADC) and sampled at $t = nT_s$ (where $T_s = 1/F_s$ is the sampling interval and F_s is the sampling frequency), can be expressed as

$$r[n] = \sqrt{P_\mathrm{r}}\, x[n-n_d] e^{j(2\pi f_d n + \theta)} + w[n] \tag{2.20}$$

where $n_d = \tau/F_s$ and $f_d = F_d/F_s$ are the discrete-time equivalents of the time delay and Doppler effect, respectively.

Acquisition Module

After the received signal has been conditioned and discretized by the front-end, the receiver aims to estimate the time delay n_d and Doppler frequency shift f_d parameters that affect the signal. These estimates are crucial for computing the associated pseudoranges, which are then used to determine the final PVT solution.

The acquisition module provides an initial coarse estimation of these parameters by correlating the received signal with the spreading codes stored in the receiver's memory (i.e., the local replicas) on a per-satellite basis. The objective is to identify a pair of tentative values, (n_d', f_d'), that closely match the true values, (n_d, f_d). Achieving this alignment between the received signal and the local replica results in a distinct correlation peak, enabling the identification of the satellite in view.

Therefore, the acquisition stage involves a two-dimensional search over both the time delay and Doppler frequency, requiring the exploration of multiple combinations of time and frequency. The bi-dimensional grid that results from this stage is usually referred to as the acquisition search space. This search can be executed in several ways. The most straightforward method is the so-called serial search, in which the receiver sequentially examines potential combinations of time delay and Doppler frequency, testing each one individually. The resulting correlation values for all possible tentative pairs yield to what is known as the Cross Ambiguity Function (CAF), that can be expressed as

$$Y(n_d, f_d) = \frac{1}{N_{\text{scode}}} \sum_{n=0}^{N_{\text{scode}}-1} r[n]c[n-n_d]e^{-j2\pi f_d n} \tag{2.21}$$

where $N_{\text{scode}} = T_{\text{code}}/T_s$ is the number of samples per spreading code.

However, the serial approach can be highly time-consuming due to the exhaustive nature of the search process. In practice, receivers commonly employ a parallel search approach (or its variants), which leverages the properties of the Fourier Transform to concurrently search across a range of time delays and Doppler frequencies. This last method significantly accelerates the acquisition process by taking advantage of parallel processing capabilities:

$$Y(n_d, f_d) = \mathcal{F}^{-1}\{\mathcal{F}\{r[n]\}\mathcal{F}^*\{c[n]e^{j2\pi f_d n}\}\} \tag{2.22}$$

where $\mathcal{F}$ is the discrete-time Fourier Transform operator, typically performed by mean of the Fast Fourier Transform (FFT).

A summary of the search strategies can be found in [28].

Whatever search method is used, the CAF can be decomposed into its real and imaginary components. Departing from (eq.1), we find that

$$\begin{aligned} Y(n_d, f_d) &= \frac{1}{N}\sum_{n=0}^{N-1} r[n]c[n-n_d]\cos(2\pi f_d n) - j\frac{1}{N}\sum_{n=0}^{N-1} r[n]c[n-\tau]\sin(2\pi f_d n) \\ &= Y_I(\tau, f_d) + jY_Q(\tau, f_d) \end{aligned} \tag{2.23}$$

The readiness of the CAF depends on the sharpness of the correlation peak and the noise floor. To reduce noise, it is possible to increase the integration time by averaging different CAFs. This process is referred to as coherent integration, and it enhances the Signal to Noise power Ratio (SNR) at the output of the correlators, provided that the averaged CAFs are phase-aligned.

If the phases of the CAFs are not aligned, it could even result in a loss of SNR, as adding CAFs with opposite phases may cancel out the signal of interest. When phase variations occur over time (e.g., due to phase noise or the Doppler effect), non-coherent integration becomes necessary. In this case, the phase of the CAFs is removed before averaging, typically by computing its envelope. This can be done in various ways, such as taking the absolute value of the CAF or its squared value. A detailed analysis of the methods used for signal detection is provided in [29].

The differences between coherent and non-coherent integration in GNSS have been extensively analyzed in the literature [reference]. In the case of coherent integration, noise reduction occurs by a factor of $1/N_i$, where N_i represents the number of averaged or integrated CAFs. For non-coherent integration, which typically uses the squared envelope (one of the most common approaches due to its well-understood statistical properties), noise reduction is approximately by a factor of $1/\sqrt{N_i}$, due to squaring losses. These losses, in turn, depend on the value of the C/N_0 [1].

Once the CAF has been evaluated (and the averaging has been completed), the acquisition stage must remove any dependence on the input signal phase. If non-coherent integration has been applied, this dependence is already eliminated. Otherwise, the phase dependence is typically removed by calculating the envelope of the CAF. The most common method involves squaring and summing the in-phase and quadrature components, as shown in the following equation:

$$X(n_d, f_d) = |Y(n_d, f_d)|^2 = Y_I^2(n_d, f_d) + Y_Q^2(n_d, f_d) \tag{2.24}$$

This technique ensures that the result is independent of the signal phase, allowing the acquisition process to focus solely on the amplitude of the correlation peak. This envelope is then used as the input for the detector, which ultimately determines whether the satellite signal is present or not. The detector compares the envelope value to a

[1] The explanation lies in the fact that, in the coherent case, the noise terms are zero-mean Gaussian random variables, allowing the CAF values to be either positive or negative. This results in a more effective averaging process, as positive and negative noise contributions can cancel each other out. In contrast, for the non-coherent case, the values are all positive due to the squaring operation, which prevents such cancellation and leads to a less efficient noise reduction.

predefined threshold, and if the value exceeds this threshold, the satellite is considered detected; otherwise, it is deemed absent. This is further analyzed in Section 2.2.5.

An example of the envelope of the CAF is illustrated in Figure 2.12.

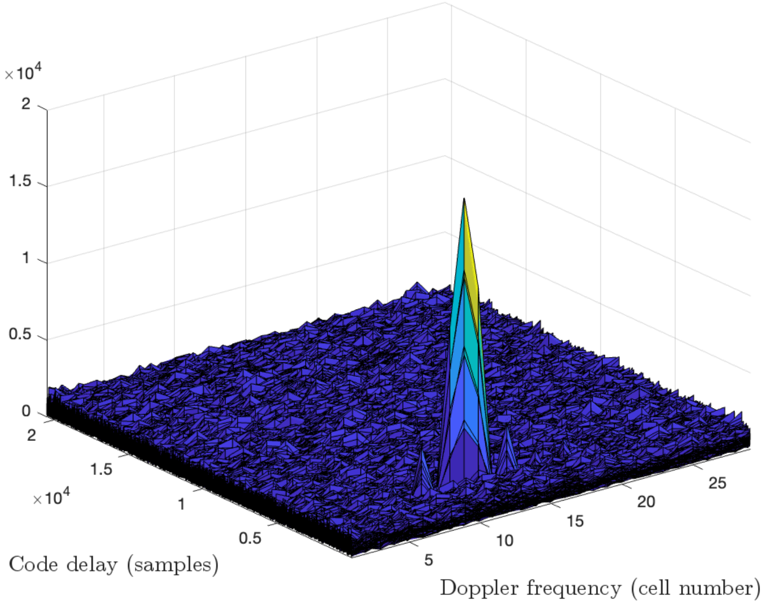


Figure 2.12: Illustration of the CAF that results from the bi-dimensional search in acquisition.

As a result of the acquisition process, the receiver will have a list of detected satellites with a coarse estimation of their respective time delay and Doppler frequency.

Tracking Module

The tracking stage of GNSS receivers is crucial for refining the coarse estimates of time delay (also commonly referred to as code delay or code phase) and Doppler frequency in the carrier obtained during the acquisition phase. These initial estimates, while essential for identifying satellites in view, often lack the precision required for accurate positioning. The purpose of the tracking stage is to refine these estimates while keeping track of changes over time [19].

Conventional GNSS receivers typically implement two parallel closed-loop architectures within the tracking module: a Delay-Lock-Loop (DLL) for *code tracking* and a

Phase-Lock-Loop (PLL) for *carrier tracking* [2]. These loops consist of three main components: the discriminator, the Numerical Controlled Oscillator (NCO), and the loop filter. The received signal is compared with a locally generated replica, and the difference between the two is input to the discriminator, which generates an error signal that is proportional to the parameter of interest. The output from the discriminator is smoothed by the loop filter, and this smoothed signal drives the NCO, which updates the local replica, thus closing the loop.

Additional techniques, such as adaptive Kalman filters, have been developed to enhance tracking robustness in challenging environments, such as urban areas or during ionospheric disturbances [30].

In snapshot receiver (open-loop) architectures, the tracking stage is omitted, and the coarse estimates from the acquisition stage are directly provided as the output observables. These architectures are often used in High Sensitivity GNSS (HS-GNSS) receivers, where the CAF computed during acquisition is refined through a narrow Doppler search to obtain more precise estimates. Snapshot architectures offer greater robustness in harsh environments with fading signals, as they are less susceptible to signal disruptions. However, they generally do not provide the same level of precision as closed-loop systems, which is why the latter remains the standard for most applications.

Navigation Module

The goal of the navigation module is to determine the user's position, velocity and/or time (the *navigation solution*) using the measurements provided by the signal processing module (either from the tracking module or directly from the acquisition module in snapshot-based architectures).

The starting point is the corrected pseudorange equation defined in (2.16), that we recall below:

$$\rho^{(k)} = d^{(k)} + c\delta t_u + \epsilon^{(k)}$$

The standard deviation of the ϵ term is around 1 meter when differential corrections are available, and around 5-10 meters where the receiver operations autonomously.

[2] Carrier tracking involves estimating both the frequency and phase of the carrier, thereby accounting for the Doppler frequency shift.

The geometric distance is the magnitude of the vector that extends from the receiver's position at the time of measurement, denoted as $\mathbf{x}_u = [x_u, y_u, z_u]$, to the satellite's position at the time of transmission, denoted as $\mathbf{x}^{(k)} = \left[x^{(k)}, y^{(k)}, z^{(k)}\right]$:

$$d^{(k)} = ||\mathbf{x}^{(k)} - \mathbf{x}_u|| = \sqrt{\left(x^{(k)} - x_u\right)^2 + \left(y^{(k)} - y_u\right)^2 + \left(z^{(k)} - z_u\right)^2} \tag{2.25}$$

As the positions of the satellites are assumed to be known by the receiver, the geometric distance depends solely on the user's position. Assuming that K satellites are used to compute the user's position, the system of equations can be written as:

$$\begin{bmatrix} \rho^{(1)} \\ \rho^{(2)} \\ \vdots \\ \rho^{(K)} \end{bmatrix} = \begin{bmatrix} d^{(1)}\left(x_u\right) \\ d^{(2)}\left(x_u\right) \\ \vdots \\ d^{(K)}\left(x_u\right) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} c\delta t_u + \begin{bmatrix} \epsilon^{(1)} \\ \epsilon^{(2)} \\ \vdots \\ \epsilon^{(K)} \end{bmatrix} \tag{2.26}$$

Each equation contains four unknowns: the clock offset and the three position coordinates. Therefore, a minimum of $K = 4$ satellites is required to calculate the position in a 3D scenario.

The system of equations of (2.26) is nonlinear due to the presence of the norm operator within (2.25). There are generally two main families of methods for solving this nonlinear system and calculating the navigation solution. The first family relies on linearizing the dependence of the distance on the user's position, typically by using the first-order term of the Taylor series. These are iterative methods that require an initial approximate value of the receiver's position to begin the calculation process [19]. The second family consists of methods that compute the position in a closed form, eliminating the need for an iterative process. They are based on the Bancroft algorithm [31]. However, these methods typically provide worse performance compared to iterative algorithms, as they often rely on squared operations to solve the problem, which amplifies the noise in the time-delay estimates.

2.2.5 GNSS Signal Detection

Signal detection theory is a branch of probability theory typically applied in various signal processing fields, including radar and communications. This theory facilitates decision-making under uncertainty, involving the distinction of whether a signal embedded in noise is present or absent. When applied to GNSS, it enables the correct identification of genuine satellite signals, which is paramount for reliable navigation and positioning.

In general, a signal detection problem is a binary hypothesis test, involving two hypothesis, commonly referred a the null hypothesis (denoted H_0) and the alternative hypothesis (H_1). In the GNSS case, these two hypothesis can be defined as follows:

- H_0: the signal from the satellite is not present (or not correctly aligned with the local replica).
- H_1: the signal from the satellite is present (and correctly aligned with the local replica).

As explained in Section 2.2.4, the receiver must take a decision about the presence or absence of the satellite as function of the metric obtained from the output of the CAF obtained in the acquisition stage. Since this decision variable is a random variable, it can be characterized by the two following Probability Density Functions (pdfs):

$$\begin{aligned}\mathrm{pdf}_{H_0}(x) &= \text{pdf of } x \text{ given that the signal from satellite is absent.}\\ \mathrm{pdf}_{H_1}(x) &= \text{pdf of } x \text{ given that the signal from satellite is present.}\end{aligned}$$

where $x \doteq X(\tau, f_d)$ is the envelope of the CAF evaluated in a given single cell.

These pdfs, and their corresponding complementary Cumulative Distribution Functions (CDFs) (denoted $\mathrm{cdf}_{H_0}(x)$ and $\mathrm{cdf}_{H_1}(x)$, respectively), completely determine performance of the detector. This performance is usually evaluated by the Probability of False Alarm (PFA), denoted P_{fa}, and by the Probability of Detection (PD), denoted P_{d}, which are defined as:

$$P_{\mathrm{fa}}(\gamma) = \mathrm{Prob}\{x > \gamma \mid H_0\} = \int_{\gamma}^{\infty} \mathrm{pdf}_{H_0}(x) = 1 - \mathrm{cdf}_{H_0}(\gamma) \tag{2.27}$$

$$P_{\mathrm{d}}(\gamma) = \mathrm{Prob}\{x > \gamma \mid H_1\} = \int_{\gamma}^{\infty} \mathrm{pdf}_{H_1}(x) = 1 - \mathrm{cdf}_{H_1}(\gamma) \tag{2.28}$$

where γ is the detection threshold.

One of the most commonly used tools to evaluate these probabilities is by plotting the probability of detection versus the probability of false alarm. This is known as the ROC curve and is described in Section 2.2.6.

It is evident that the performance of the detector depends on the distributions of the decision variables x, which in turn depend on how the CAF is evaluated. The statistics of the CAF are driven by the noise statistics, which are commonly assumed to be AWGN, with independent real and imaginary parts. One of the most commonly used techniques for evaluating the CAF (see Section 2.2.4) involves taking its square absolute value, so that the resulting decision variable is the sum of the squared real and imaginary parts of the CAF:

$$x = |Y(\tau, f_d)|^2 = Y_I^2(\tau, f_d) + Y_Q^2(\tau, f_d) \tag{2.29}$$

For the null hypothesis, it equals the sum of two zero-mean squared Gaussian distributions, resulting in a central chi-squared distribution with two degrees of freedom, which is equivalent to an exponential distribution with a rate parameter of $\frac{1}{2}$:

$$x \mid H_0 \sim \chi_2^2(\sigma_n^2) = e^{\frac{1}{2\sigma_n^2}} \tag{2.30}$$

where $\sigma_n^2 = \mathrm{Var}[Y_I^2(\tau, f_d)] = \mathrm{Var}[Y_Q^2(\tau, f_d)] = \frac{\sigma_{\mathrm{IF}}^2}{2N}$.

For the alternative hypothesis, it equals the sum of two non-zero-mean squared Gaussian distributions, resulting in a non-central chi-squared distribution with two degrees of freedom:

$$x \mid H_1 \sim \chi_{nc,2}^2(\lambda, \sigma_n^2) \tag{2.31}$$

where $\lambda = \frac{A^2}{4}$ is the non-centrality parameter.

Therefore, the corresponding pdfs for the null and alternative hypothesis are, respectively:

$$\mathrm{pdf}_{H_0}(x) = \frac{1}{2\sigma_n^2} e^{-\frac{x}{2\sigma_n^2}} \tag{2.32}$$

$$\mathrm{pdf}_{H_1}(x) = \frac{1}{2\sigma_n^2} e^{-\frac{x+\lambda}{2\sigma^2}} I_0\left(\frac{\sqrt{x\lambda}}{\sigma_n^2}\right) \tag{2.33}$$

where $I_0(\cdot)$ is the modified Bessel function of the first kind and zero order.

Depending on the detection technique applied, different pdfs would result. Apart from the coherent technique shown here, the non-coherent and differentially-coherent techniques are among the most well-known [32]. A detailed derivation of pdfs can be found in [33], which provides a comprehensive analysis of signal detection theory applied to GNSS.

Considering the previous pdfs we can now calculate the corresponding probabilities of detection and false alarm:

$$P_{\mathrm{fa}}(\gamma) = \int_{\gamma}^{\infty} \mathrm{pdf}_{H_0}(x) = \exp\left\{-\frac{\gamma}{2\sigma_n^2}\right\} \tag{2.34}$$

$$P_{\mathrm{d}}(\gamma) = \int_{\gamma}^{\infty} \mathrm{pdf}_{H_1}(x) = Q_1\left(\sqrt{\frac{\lambda}{\sigma_n^2}}, \sqrt{\frac{\gamma}{\sigma_n^2}}\right) \tag{2.35}$$

where $Q_1(\cdot)$ is the Marcum Q-function of order 1.

An illustrative plot of a binary hypothesis test problem is shown in Figure 2.13. A total of four cases are derived, depending on the hypothesis predicted by the detector (H_0 or H_1) and the actual hypothesis ($\widetilde{H}_0$ or $\widetilde{H}_1$), which are summarized below:

True Negative (TN) (Correct Rejection) – It is decided that there is no signal (H_0) when there is no actual signal ($\widetilde{H}_0$) (satellite correctly rejected).

False Positive (FP) (False alarm) – It is decided that there is signal (H_1) when there is no actual signal ($\widetilde{H}_0$) (satellite incorrectly detected).

False Negative (FN) (Miss) – It is decided that there is no signal (H_0) when there is actual signal ($\widetilde{H}_1$) (satellite incorrectly rejected).

True Positive (TP) (Hit) – It is decided that there is signal (H_1) when there is actual signal ($\widetilde{H}_1$) (satellite correctly detected).

From these definitions, one can easily determine that the probability of detection is basically the True Positive Rate (TPR), also known as the sensitivity of the detector, which is calculated as:

$$P_{\mathrm{d}} = \mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.36}$$

Consequently, the probability of false alarm is the equivalent to the False Positive Rate

(FPR):

$$P_{\text{fa}} = \text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{2.37}$$

The FPR and, therefore, the P_{fa}, can also be expressed from the True Negative Rate (TNR), also known as the specificity:

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{2.38}$$

$$\text{FPR} = 1 - \text{TNR} \tag{2.39}$$

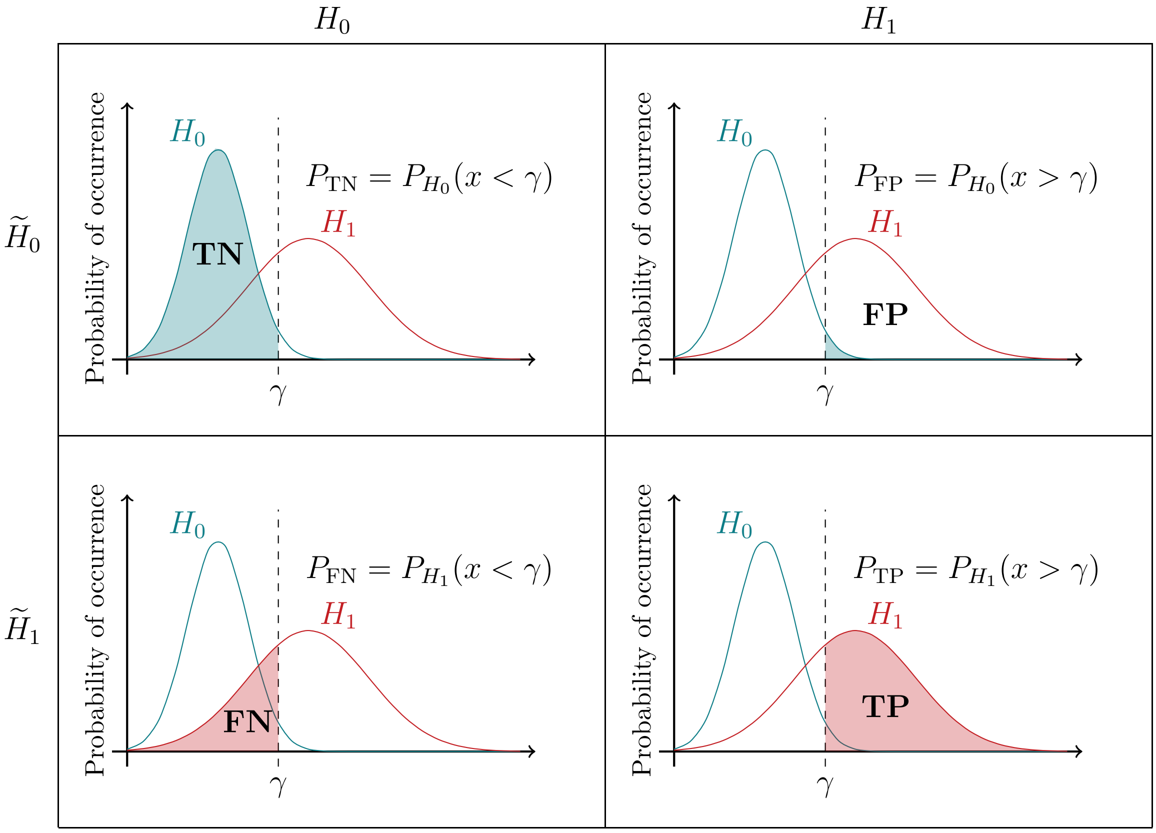

H_0
H_1
$\widetilde{H}_0$
$\widetilde{H}_1$
Probability of occurrence
H_0
H_1
γ
$P_{\text{TN}} = P_{H_0}(x < \gamma)$
TN
$P_{\text{FP}} = P_{H_0}(x > \gamma)$
FP
$P_{\text{FN}} = P_{H_1}(x < \gamma)$
FN
$P_{\text{TP}} = P_{H_1}(x > \gamma)$
TP

Figure 2.13: Illustrative plot of the binary hypothesis problem in signal detection theory.

As we can observe in Figure 2.13, the pdfs of both hypotheses are shifted, since they have different means. Indeed, the mean of pdf_{H_0} is $2\sigma_n^2$, whereas the mean of pdf_{H_1} is $2\sigma_n^2 + \lambda$, where λ is the non-centrality parameter. This shift enables the establishment of a detection threshold between the two distributions that allows us to discriminate between the two hypotheses. However, since there exists some overlapping between the two distributions, it is not possible to fully discriminate between them, leading to a certain probability of false alarm. Therefore, lowering the threshold improves the probability of detection at the expense of increasing the probability of false alarm, and vice versa.

In typical GNSS receivers, the detection threshold is usually established for a small value of the probability of false alarm to avoid a large number of incorrectly detected satellites (i.e., false positives). Therefore, the determination of such a threshold is of key importance for the correct operation of the receivers. For a given probability of false alarm for a single cell, denoted $P_{\mathrm{fa}}^{\mathrm{t}}$, this threshold can be derived from (2.35) as:

$$\gamma = -2\sigma_n^2 \ln P_{\mathrm{fa}}^{\mathrm{t}} \tag{2.40}$$

Consequently, the resulting probability of detection can be also expressed as function of this target probability of false alarm as:

$$P_{\mathrm{d}} = Q_1\left(\sqrt{\frac{\lambda}{\sigma_n^2}}, \sqrt{-2\ln P_{fa}^{\mathrm{t}}}\right) \tag{2.41}$$

The expressions of the probabilities of detection and false alarm derived in (2.32) and (2.33) respectively for coherent integrations can also be derived when applying N_i multiple non-coherent integrations:

$$P_{\mathrm{fa},N_i}(\gamma) = \exp\left\{-\frac{\gamma}{2\sigma_n^2}\right\}\sum_{k=0}^{N_i-1}\frac{1}{k!}\left(\frac{\gamma}{2\sigma_n^2}\right)^k \tag{2.42}$$

$$P_{\mathrm{d},N_i}(\gamma) = \mathrm{Q}_{N_i}\left(\sqrt{N_i\frac{\lambda}{\sigma_n^2}}, \sqrt{\frac{\gamma}{\sigma_n^2}}\right) \tag{2.43}$$

The expressions obtained for the probability of detection in (2.33) and (2.43) can be also expressed in terms of the output SNR ρ_c computed previously under ideal conditions, considering that $\rho_c = \frac{A^2/4}{\sigma_n^2} = \frac{\lambda}{\sigma_n^2}$:

$$P_{\mathrm{d}}(\gamma) = Q_1\left(\sqrt{\rho_c}, \sqrt{\frac{\gamma}{\sigma_n^2}}\right) \tag{2.44}$$

$$P_{\mathrm{d},N_i}(\gamma) = Q_{N_i}\left(\sqrt{N_i\rho_c}, \sqrt{\frac{\gamma}{\sigma_n^2}}\right) \tag{2.45}$$

Finally, when the whole search space is evaluated, the global probability of false alarm

P_{FA} can be obtained from the single cell probability of false alarm as:

$$P_{\mathrm{FA}}(\gamma) = 1 - (1 - P_{\mathrm{fa}}(\gamma))^M \tag{2.46}$$

where M is the number of independent cells evaluated in the acquisition search space.

Regarding the probability of global detection, denoted P_{D}, is usually considered approximately equal to the single cell one:

$$P_{\mathrm{D}}(\gamma) \approx P_{\mathrm{d}}(\gamma) \tag{2.47}$$

2.2.6 Understanding ROC Curves

A Receiver Operating Characteristic (ROC) curve is a technique used in signal detection theory for evaluating binary classification systems by providing a graphical representation of the performance of such classifiers [34]. By plotting the true positive rate against the false positive rate at various threshold settings, ROC curves offer a comprehensive insight into the trade-offs between detection rates and false alarms. This is crucial in GNSS applications where signal detection must be both accurate and reliable.

One of the key advantages of using ROC curves in GNSS signal detection is their insensitivity to changes in class distribution. This property is particularly valuable in GNSS contexts where the prevalence of signal events may vary significantly over time or space. The ability to maintain consistent performance evaluation despite these variations ensures that the detection algorithms remain robust under different operational scenarios.

Moreover, ROC curves support the comparison of multiple detection algorithms, enabling the identification of the most effective methods for GNSS signal detection. This can be achieved by comparing the areas under the ROC curves, known as Area Under the Curve (AUC), which quantifies the overall ability of the algorithm to discriminate between true positives and false positives.

The interested reader can refer to [35], which offers a comprehensive examination of ROC curves, elucidating their theoretical underpinnings and practical applications.

Some illustrations of typical ROC curves are shown in Figure 2.14, where the shaded areas indicate the AUC for each curve. “Classifier 1” performs better than “Classifier 2”, as the AUC of the first classifier is larger than that of the second. Indeed, for a given target probability of false alarm, the probability of detection of the first classifier ($P_{\mathrm{d},1}$) is larger than that of the second one ($P_{\mathrm{d},2}$). The worst-case scenario would have a perfectly diagonal ROC curve, indicating that the detector is performing no better than random guessing.

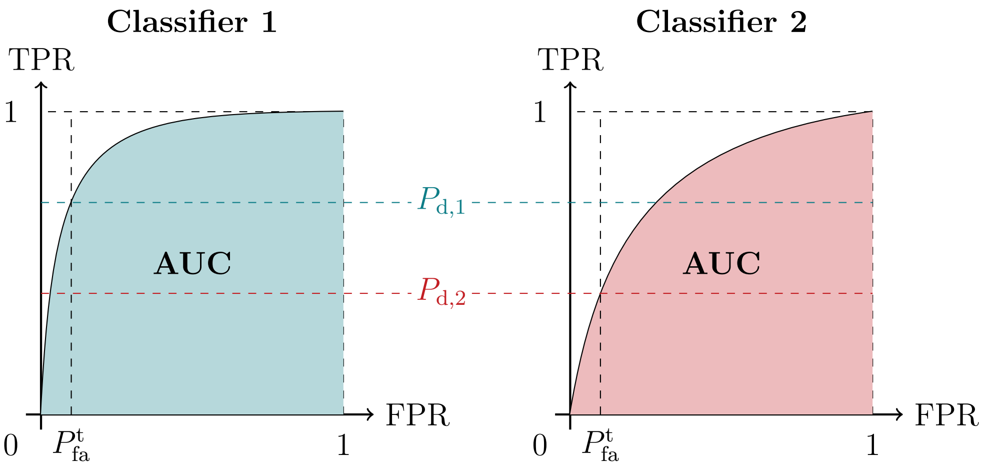


Figure 2.14: ROC curves illustration

.

2.2.7 Receiver Clock Model

Phase noise is one of several impairments that affect GNSS signals, primarily caused by receiver clock instabilities. These phase variations can lead to phase wrappings that limit the time the receiver can integrate coherently. Therefore, it is of paramount importance to simulate phase noise using an appropriate clock model.

The performance of a clock is usually analyzed in terms of frequency instability [17]. To characterize this instability, the common standard deviation is not useful since the frequencies diverge for some exponential-type noises; instead, the two-sample or Allan Variance (AVAR) is used [36]. This statistical tool measures frequency instability for all those noise types, as it converges to finite values for all the major noise types observed in precision oscillators [37].

Different clock models can be applied [17], but the most common is the two-state model is based on [38], which in turn is based on [39]. Based on the equation of [39], the two-state clock model shown in Figure 2.15 is commonly used to simulate the phase noise:

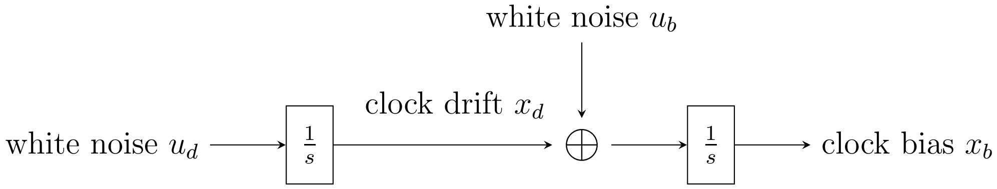


Figure 2.15: Two-state clock model.

From the analogue model of clock error depicted in figure, two discrete-time equations can be derived:

$$x_d[n+1] = x_d[n] + u_d[n]T_s \tag{2.48}$$

$$x_b[n+1] = x_b[n] + u_b[n]T_s + x_d[n]T_s \tag{2.49}$$

where u_b and u_d are two independent Gaussian noise components, T_s is the sampling time, x_d is the clock drift, and x_b is the clock bias. More precisely, x_b represents the random fluctuations that have GNSS signal.

The power of noises u_b and u_d depends on the clock type and they are defined by:

$$u_b = \sqrt{\frac{S_f}{T_s}}\mathcal{N}(0,1) \tag{2.50}$$

$$u_d = \sqrt{\frac{S_g}{T_s}}\mathcal{N}(0,1) \tag{2.51}$$

where $\mathcal{N}(0,1)$ is a Gaussian noise with zero mean and unit variance, S_f and S_g are the spectral amplitudes of the noise and they are approximated by the following expression:

$$S_f = \frac{h_0}{2} \tag{2.52}$$

$$S_g = 2\pi^2 h_{-2} \tag{2.53}$$

The coefficients h_0 and h_{-2} (random walk frequency noise) are typically used to characterize clocks. They correspond to white frequency noise and random walk frequency noise, respectively. As stated in [38], "the flicker [frequency] noise (h_{-1} term) was ignored since it is impossible to model it with a finite-order state model. Ignoring the flicker noise results in a mismatch between the model and the desired Allan variance for flicker noise. To account for this, S_f and S_g may be increased as done in [39]. This results in a 'compromise model', with a higher drift at short (less than 1 second) and long averaging times, but a better match in the flicker noise region."

These coefficients depend on the quality of the clock. Some typical values for different types of clocks are provided in Section 4.3.6. The coefficient h_0 is related to short-term stability, while h_{-2} is related to long-term stability.

2.3 A Brief History on GNSS Authentication

2.3.1 Introduction

Almost three decades after the first GNSS became operational, positioning and navigation receivers have become a commodity. Billions of units are shipped every year, and the demand is continuously increasing as new applications and services are foreseen. The mass production and growing interest in the many navigation constellations available today have, however, a significant downside: it is relatively straightforward to generate and broadcast a fake GNSS signal with a relatively low-cost hardware, so any conventional receiver could be tricked to accept this counterfeit signal.

The need to mitigate this vulnerability, known as spoofing [40], has led to intensive research from the GNSS community to investigate effective countermeasures against it, as well as cryptographic protection, either for the navigation message or for the ranging codes. Indeed, the demand for techniques capable to authenticate these signals and detect spoofing attacks have increased notably in the last years [41].

A concept of GNSS authentication service was examined in [42], based on the idea that the presence of unpredictable information in the signal and data could be used to authenticate the signal, since a spoofer would not be able to predict this information. In [43] we find the first attempt to integrate an authentication mechanism for open signals in GNSS, which is based on secret spreading sequence. Another authentication scheme was proposed in [44], in which only the navigation data is authenticated. However, this last scheme can still be circumvented by replay attacks, since the navigation data can be acquired by a receiver and re-transmitted with a different delay. Since then, many approaches and techniques have been proposed and even implemented [45]. In this regard, Galileo has taken a lead over the rest of navigation systems, incorporating in its service baseline a Navigation Message Authentication (NMA) service, following recommendations from mission and feasibility studies launched in 2013 [46].

Proposed only some years ago [47], Galileo OSNMA, which consists of adding cryptographic information to the navigation data of the Open Service (OS) to ensure the data authenticity, has been recently made freely available to all users in the frame of a public observation phase [48], which is a clear commitment from the Galileo stakeholders to provide resilient location services to the GNSS community. Nevertheless, despite the many advantages of NMA-based schemes, their main aim is not to authenticate the ranging signal. They are not designed to provide a PVT solution completely protected against the spoofing attacks, but rather to be an extra layer of security that will contribute to improve the user level authentication.

To increase location security, Galileo users may rely on the forthcoming SAS. It is based on the encryption of the E6-C component at signal level to provide Spreading Code Authentication (SCA), which allows a greater level of protection against spoofing, in particular against replay attacks. Together with HAS, Galileo will become the first GNSS to provide both authentication and high-accuracy data [49]. These three new services (OSNMA, HAS, and SAS) are the results of the evolution on the definition of the original Commercial Service (CS), intended to be added-value services to be provided by Galileo [49].

SAS is currently under development, but an early service is envisaged by 2024, which will make use of some ancillary data from the E1-B signal allowing the receiver to decrypt the codes without the need of storing any secret key.

2.3.2 Understanding GNSS Signal Vulnerabilities

Recognizing and addressing threats is crucial in GNSS [50]. These threats can exploit receiver vulnerabilities, leading to minor inaccuracies in the best-case scenario and severe issues in the worst-case scenarios, particularly in critical and safety-of-life applications. Controlling and diverting civilian aircrafts [51] or ships [52] from their intended routes is an example of such threats, which becomes even more concerning in military scenarios.

The GNSS community has devoted significant effort to classify the different vulnerabilities a receiver may face, including environmental conditions, unintentional interference or more sophisticated attacks carried out by malicious users [53]. Next, we briefly describe some of the most important threats.

Interference is a disturbance generated by an external source that affects the useful signal. When applied to radio-frequency it is known as Radio-Frequency Interference (RFI). There are numerous forms of such interference, including electromagnetic interference or thermal noise. Interference can be inter-system or intra-system, pulsed or continuous, wide-band or narrow-band, in-band or out-band, unintentional or intentional [54].

Regarding unintentional interference, the effects from the ionosphere (scintillations[3]) is one of the main sources of error in GNSS. Additionally, the interference produced by Radio-Frequency (RF) signals from other systems, such as Digital Video Broadcasting – Terrestrial (DVB-T), whose harmonics could coincide with the GPS L1 or Galileo E1 bands, is one of the threats to be considered [55].

Jamming can be defined as the deliberate use of radio noise or signals in an attempt to disrupt communications. This usually involves emitting high power radio-frequency signals matching the frequencies used in GNSS, so that the receiver is unable to distinguish the original signal. It is, therefore, a form of intentional interference. Currently, some radio users use the term "jamming" to describe exclusively the intentional use of radio signals to disrupt communications, while the term "interference" is reserved for unintentional forms of disruption.

When the frequency of the jammer (i.e., the device performing jamming) is aligned with the target GNSS band, it can cause a severe degradation of the carrier-to-noise ratio of received signals by the receiver, or even a complete loss of the tracking loops when the transmitted power saturates the receiver front-end [56].

Meaconing is the interception and rebroadcast of navigation signals. When rebroadcasting at the same frequency (and possibly power-amplified), meaconing systems can cause inaccurate bearings to be obtained by GNSS receivers [57]. The term "meacon" is derived from the combination of "masking" and "beacon". Comparing to jamming, meaconing is a more refined form of threat, as it involves the reception and subsequent rebroadcast of the signal.

Spoofing refers to the act of disguising communication from an unknown source as if it originates from a known, trusted source. It is a malicious attempt to manipulate the GNSS receiver measurements, rendering their PVT unreliable. Spoofing can be

[3] Ionospheric scintillations are fluctuations in the phase and amplitude of GNSS signals, resulting from the electron density irregularities along the radio propagation path (ionosphere).

understood as an intelligent form of interference, causing the receiver to perceive a false location [58]. Spoofing attacks can aim to produce incorrect time, incorrect position, or both. Additionally, spoofers can generate asynchronous or synchronous attacks, with the latter meaning that the counterfeit signals are synchronized with the authentic GNSS signals. Generally, these attacks are classified according to their level of sophistication, as described in [59]. For instance, spoofing attacks using multiple coordinated antennas are more complex to realize than spoofers using a single-antenna approach. One of the most common attacks is the so-called Security Code Estimation and Replay (SCER) [60].

Spoofing is a more sophisticated attack than meaconing, as it involves demodulation of the received signal and sending modified satellite signals, while meaconing simply emits signals recorded at a different location and time[4]. Indeed, the process requires a deeper understanding and manipulation of the GNSS signal structure to create convincing counterfeit signals that deceive GNSS receivers.

Tampering is defined as the illegal act of touching or making changes to something when you should not. This kind of threat falls into a different category compared to the previous threats, as it assumes that the malicious user has physical access to the GNSS receiver. This threat is particularly critical when dealing with cryptographic keys stored in receivers; in such instances, it is essential to implement anti-tampering mechanisms to prevent unauthorized access to these keys [61].

In recent years, with increasing political tensions and military conflicts, the vulnerabilities of GNSS-based systems and applications have become even more evident [62]. This is why major players in the industry have implemented numerous anti-spoofing techniques, along with anti-jamming and interference-mitigation strategies, to ensure reliable GNSS positioning [63] [64].

Although some of the techniques can be applied to both jamming and spoofing, others are more specific to a particular threat. For instance, the receiver could implement RF filtering techniques to suppress out-of-band interference, an specific case of jamming attack, but could also perform "nulling"[5] using Control Reception Pattern Antennas (CRPAs), a

[4] It is worth mentioning that sometimes the term "spoofing" is used in such a way that it also encompasses the term "meaconing", considering the latter as a simpler particular case of a spoofing attack.

[5] Null-based beamforming or "nulling" is a technique that allows modifying the antenna pattern to create "nulls" in the desired direction.

more sophisticated RF technique aimed at both in-band jamming and spoofing attacks. [53].

When dealing with interference, including intentional jamming, the receiver usually needs to identify the kind of interference suffered (narrow-band/wide-band, continuous/pulsed, etc.) and apply the most suitable method to counter it, like adaptive notch filtering or Multi-Frequency Multi-Constellation (MFMC) processing [54] [56]. Although increasing the cost of the RF front-end, processing distinct bands provides increased diversity, in addition to removing ionospheric degradation.

This concept of diversity is indeed key in many anti-jamming and anti-spoofing techniques, as it enhances overall system robustness. As the receiver adds more levels of protection through additional checks, the complexity required for the jammer or spoofer to successfully carry out the attack must increase. This diversity can be achieved through many dimensions and is especially important for combating spoofing attacks.

Even though some of the GNSS vulnerabilities are quite complex to exploit, successful attacks can now be performed with minimal cost: a simple SDR and software freely available from the internet may suffice. Some examples using low-cost SDR boards such as the HackRF One and LimeSDR mini can be found in [65] and [66], respectively. Another example of SDR use can be found in [67], where a bladeRF board is used to implement jamming techniques for Unmanned Aerial Vehicles (UAVs).

Clearly, spoofing is among the main threats in the GNSS field, as the goal of such technique is to remain undetected by the receiver being attacked [51] [52]. Compared to jamming, which causes the receiver to cease functioning, spoofing causes the receiver to provide false information. This is why spoofing warrants special attention and why the following sections are specifically dedicated to describing anti-spoofing countermeasures.

A useful classification of spoofing protection defenses is to divide them into non-cryptographic and cryptographic techniques [68], which are briefly described in Sections 2.3.3 and 2.3.4, respectively. Another common classification of anti-spoofing techniques is to divide them into receiver-side and system-side techniques, as described in [69], which roughly corresponds to the previous classification[6].

[6] Cryptographic techniques, as detailed later in Section 2.3.4, require the implementation of authentication mechanisms in the GNSS signal broadcast and, therefore, can be classified as system-side techniques.

2.3.3 Anti-Spoofing: Non-Cryptographic Techniques

Non-cryptographic techniques, as opposed to cryptographic ones, do not require any kind of modification of the GNSS signal structure. Therefore, they use existing physical manifestations (such the receiver power or the angle-of-arrival) to protect the receiver against malicious attacks to the extent possible.

Numerous mitigation techniques have been and are being applied, ranging from very simple tests to highly complex mechanisms, the cost of which may not be suitable for mainstream receivers. This is an important aspect to consider when implementing these techniques: the countermeasures applied must be commensurate with the threats faced by the specific receiver. Indeed, some of these countermeasures can be implemented with basic algorithms in software, but others may require incorporating expensive hardware reserved only for high-end GNSS receivers.

As mentioned above, spoofing attacks are usually the more complex attacks to be carried out, and can completely fool the receiver with a non-authentic signal. A comprehensive introduction to the many problems spoofers can pose can be found in [59], which includes a list of possible techniques for countering spoofing. This resource differentiates between the simplest software-based methods and the more complex hardware-based solutions.

Software-based techniques like amplitude or time-of-arrival discrimination are considered as a first barrier to spoofers, since they are relatively easy to implement, and could be very effective against simplistic spoofing attacks. Hardware-based techniques, potentially requiring additional hardware components (such as additional antennas), may be more restrictive to implement but generally result in more efficient solutions. Essentially, anti-spoofing techniques involve discriminating certain physical magnitudes, such as amplitude, time-of-arrival, angle-of-arrival, or polarization, to determine if the signal received is authentic or counterfeit. Next, we provide a summary of these techniques.

Received Power/Amplitude – Amplitude discrimination can be achieved by monitoring the received power in the RF front-end to detect unexpected changes. This can be done using the Automatic Gain Control (AGC), as described in [70]. Spoofing attacks typically involve increasing signal levels, which prompts the AGC to decrease its gain to prevent the ADC from saturating. These gain variations in the AGC can be used to detect such attacks.

Time of Arrival/Pseudorange – Time of arrival discrimination involves comparing the pseudoranges (derived from the arrival times) obtained from different satellites to detect any mismatches among them. Spoofed signals often have inconsistent arrival times due to the difficulty in perfectly synchronizing counterfeit signals with authentic ones.

This kind of defense is implemented in the known as Receiver Autonomous Integrity Monitoring (RAIM) techniques. These techniques use redundant measurements from multiple satellites to continuously evaluate, detect, and mitigate potential errors or faults, ensuring the integrity of the position solution. Indeed, some receivers may simply raise an alarm when an inconsistent measurement is detected, whereas more advanced ones can exclude the faulty signals that may cause positioning instability to ensure continuous and reliable operation. This process, commonly known as Fault Detection and Exclusion (FDE), is crucial for applications where precision and safety are paramount, such as in avionics [64]. The consistency of the times of arrival and/or pseudoranges can be verified using traditional geometric approaches, as well as advanced methods that employ more sophisticated statistical models and algorithms [71].

Signal Quality Monitoring (SQM) It refers to the methods and technologies used to continuously assess the quality and integrity of the received GNSS signals. Unlike RAIM techniques that rely on observables (e.g., pseudoranges) at the PVT level, SQM techniques operate in the signal domain by leveraging the information at the output of the correlators. These techniques have been widely used to detect signal distortions produced by multipath effects, but they can also be applied to detect spoofing attacks. In both scenarios, a replica of the signal of interest is received at the receiver along with the desired signal, which can be exploited at the correlation stage.

When no spoofing attacks are present, a GNSS receiver ideally obtains a clear correlation peak indicating the presence of the satellite related to the PRN sequence used for such correlation. In the case of a spoofing attack, a secondary peak related to the spoofed signal may appear, clearly indicating the presence of a spoofer (if the multipath option can be ruled out). This can be exploited by the so-called Multi-Peak Detection (MPD) techniques, which are based on detecting more than one correlation peak in the acquisition search space [72]. When the delay of the spoofed signal is very similar to that of the authentic signal, the correlation peaks might not be distinguishable. In such cases, monitoring the shape of the correlation peak to analyze possible distortions caused by the presence of a counterfeit signal is a valuable anti-spoofing defense. This approach can help discern the sometimes subtle differences between authentic and spoofed signals [73] [74].

Angle/Direction of Arrival – These techniques exploit differential carrier phase measurements taken between multiple antennas [75]. They require the use of multi-antenna setups, which can sometimes be implemented only in high-grade receivers due to their cost or the physical dimensions required. However, receivers equipped with these techniques typically can only be spoofed by sophisticated coordinated spoofing attacks.

Multi-antenna techniques analyze the consistency of the spatial signature of the electromagnetic field of the signal. In a typical spoofing attack, a single spoofing source generates multiple satellite signals, all originating from a single direction different from the actual satellite directions. This discrepancy can be detected by analyzing differential carrier phase measurements between the antenna elements [76].

Signal Polarization – These techniques use dual-polarized antennas to exploit additional information on the electromagnetic field of the received signal. While they require additional hardware, they do not necessitate large or costly multi-antenna systems —only the same radiating element and additional circuitry with the appropriate electronic phase rotation.

The underlying principle is detailed next. Authentic signals are transmitted with very pure Right-Hand Circular Polarization (RHCP); however, due to degradation caused by reflections, non-zero Left-Hand Circular Polarization (LHCP) components will be present in the associated electromagnetic field at the receiver. Signals from different satellites generally arrive from various directions, resulting in differing polarizations of the electromagnetic field for each satellite. In contrast, the electromagnetic fields of spoofed signals are expected to be received with the same polarization, as they all follow the same path from the spoofer to the receiver [77].

Inertial Measurements – Inertial Navigation Systems (INSs) are navigation aids that calculate the position, orientation, and velocity of a moving object without the need for external references. These systems extract raw data collected by an Inertial Measurement Unit (IMU), which commonly measures acceleration and angular velocity. When a GNSS receiver is equipped with these auxiliary sensors, it is possible to compare the inertial data with the data retrieved from the GNSS signal. Discrepancies between these measurements could indicate the presence of a spoofer [78].

GNSS and INS can be combined using a variety of integration schemes, ranging from simple loosely coupled integration to more complex tightly coupled schemes [79]. INS are also sometimes used in combination with RAIM techniques, where the redundancy is provided through inertial measurements, unlike conventional RAIM that relies on satellite redundancy for detection [80].

In addition to the techniques described above, there are many more, such as the use of high-frequency antenna motion [81] or the discrimination by Doppler frequency shift difference [82] [83]. The study of anti-spoofing techniques is, in fact, constantly evolving, and new proposals to combat spoofing attacks, which are becoming increasingly sophisticated, emerge every day. A comprehensive overview of the different anti-spoofing techniques and their classification can be found in [84]. Additional surveys on spoofing detection and mitigation can be found in [85] [86] [87].

2.3.4 Anti-Spoofing: Cryptographic Techniques

Cryptographic defenses employ cryptographic algorithms to conceal part of the information during signal generation, rendering it infeasible to predict or forge by unintended users. This allows an authorized receiver to distinguish authentic signals from counterfeits. In GNSS, this can be achieved by encrypting the spreading code chips or the navigation message bits, as proposed in [43]. Since the secret key is not public, the spoofer cannot forge the encrypted information.

Cryptographic methods have been extensively studied in the literature [44] [88], but all of these methods require modifications to the GNSS signal structure. Military signals, such as the P(Y) code used in GPS, already provide protection based on Spreading Code Encryption (SCE). However, civilian GNSS systems were not initially designed to incorporate any authentication mechanisms, making them vulnerable to various threats, especially spoofing. Consequently, various institutions have been investing in and implementing new services and signals to provide resilience against spoofing and other malicious attacks. This is crucial to ensure that end-users can rely on GNSS for their critical applications.

The encryption and authentication mechanisms can be implemented at two distinct but complementary levels:

Data level – This is achieved by encrypting the broadcast navigation message bits. In this case, a spoofer will be able to track the GNSS signal but will not be able to demodulate the navigation message and, therefore, will not be capable of calculating the authentic PVT. Navigation Message Authentication (NMA) techniques fall into this category.

Range level – Also referred to as signal level, this is done by encrypting the spreading code chips. This offers an increased layer of protection compared to the data level, as a spoofer would not be able to track the signal or attempt to guess the navigation message bits. Spreading Code Authentication (SCA) schemes belong to this domain, based on.

Next, we outline the new services that have recently emerged or are currently in development, aiming to provide data-level, range-level authentication, or both.

OSNMA – Open Service Navigation Message Authentication

The first civilian service to implement cryptographic defenses in GNSS is the so-called Open Service Navigation Message Authentication (OSNMA) [47], which offer data level authentication in the Galileo E1-B open signal . The security code is embedded in the navigation signal, which is easy to validate but challenging to forge. It is based on the TESLA protocol [89]. It thus serves as a digital signature that verifies the authenticity of the navigation data demodulated by the receiver. Freely available to all users, OSNMA is oriented towards mass-market solutions to ensure its protection benefits the maximum number of users possible. Additionally, the use of unpredictable bits in OSNMA can enhance anti-replay techniques to combat spoofing attacks [90]. Further information about this service can be found in Section 3.6.

Although Galileo OSNMA is the most visible and successful implementation of NMA to date, other NMA-based schemes similar to OSNMA have been proposed for various GNSS, including GPS [91] and BeiDou Navigation Satellite System (BDS) [92]. Additionally, Satellite Based Augmentation System (SBAS) authentication schemes have been proposed, such as in [46].

SAS – Signal Authentication Service

OSNMA is intended to be complemented by Galileo Signal Authentication Service (SAS), based on SCE. Originally known as Assisted Commercial Authentication Service (ACAS), it will offer range authentication using the E6-C pilot signal, soon to be encrypted. In SAS, which is currently being developed, the cryptographic keys disclosed by OSNMA in E1-B are employed to eliminate the need for the receiver to store any secret keys. This avoids the use of tamper-resistant receivers, leading to reduced costs. A detailed description of its features is provided in Section 3.2.

CHIMERA – Chips-Message Robust Authentication

SCA schemes can be also designed to operate in standalone mode, without any assistance [93]. Indeed, GPS is currently developing a new authentication service for L1-C open signal[7], known as Chips-Message Robust Authentication (CHIMERA), which is testing both standalone and assisted methods.

This service, based on the principles outlined in [95], implements protection at both range and data levels by jointly utilizing NMA and SCA: NMA is established through the digital signature of the navigation message, while SCA is accomplished by embedding markers (watermarks) into the civilian spreading code. This is achieved by substituting small segments of the spreading codes with a secret sequence that the receivers can later reproduce once they obtain the key.

For a standard receiver not using the service, the substitution by unknown fragments shortens the effective length of the spreading code. This results in a small correlation loss during the acquisition and tracking stages [96]. The cadence of this substitution, known as the (water)marker duty cycle, is a crucial aspect in the design of watermarking signal authentication systems like CHIMERA [43].

[7] The GPS L1-C signal consists of the data component (C) and the pilot component (D), operating at the same frequency as legacy GPS L1 C/A and Galileo E1 band [94].

CHIMERA implements two watermark channels for ranging authentication: the slow channel, which relies solely on the navigation message, and the fast channel, in which the receiver uses an additional communication link (out of band) to receive the key required for authentication. The fast channel thus eliminates the need for users to demodulate the navigation message data and allows for more frequent changes of the marker keys [96]. In the slow channels, watermarks are provided at a minute-based cadence, whereas in the fast channel, they are delivered at a seconds-based cadence (see Figure 2.16).

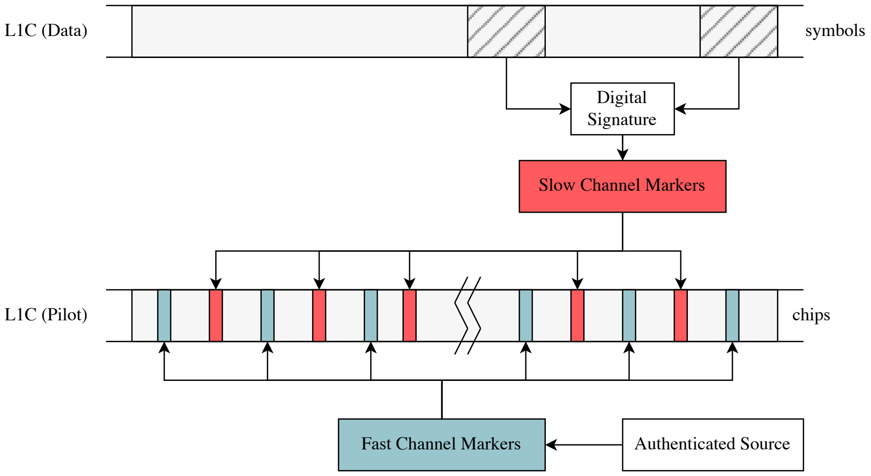


Figure 2.16: CHIMERA operating scheme.

CHIMERA has also been combined with OSNMA to offer enhanced protection capabilities [97]. However, other SCA-based methods are also possible [68] [69].

Comparison of Cryptographic-based Services

The underlying principle of the aforementioned services based on cryptographic techniques is to divide the signals into real-time and delayed components, a method applicable to any GNSS signal [98]. Furthermore, all of these services assume that in the event of a malicious attack, the (encrypted) authentic signal will arrive earlier than the unauthentic one [99]. Table 2.2 provides a summary of the key characteristics of these services.

However, Galileo and GPS have taken different approaches to implementing these principles. While Galileo has opted to offer two differentiated services—data authentication through OSNMA and range authentication via SAS—GPS is working towards delivering both authentications in a single service, the so-called CHIMERA. Although both approaches have their respective advantages and drawbacks, Galileo has taken the

Service	**Authentication level**	**GNSS band**	**Status**	SCA mode
OSNMA	Data level	Galileo E1-B	Operative	N.A.
SAS	Range level	Galileo E1-B + E6-C	In development	Assisted
CHIMERA	Data & Range level	GPS L1C (C+D)	In testing	Assisted + Standanlone

Table 2.2: Comparison of new authentication services in GNSS.

lead by recently launching OSNMA. While it does not yet provide a fully secure solution, it offers enhanced protection for GNSS receivers with minimal impact on their performance. Additionally, it has raised awareness among users about the risks of malicious attacks.

For range authentication, Galileo has chosen to encrypt the entire E6-C pilot signal. The main drawback of this approach is the complete loss of the pilot for receivers using the E6 band (e.g., future HAS-compatible receivers). In contrast, GPS's CHIMERA approach only "loses" certain fragments of the pilot component in the L1-C signal.

On the other side, while currently SAS is designed in such way that a receiver would need continuous assistance from a server, CHIMERA is testing both standalone and assisted modes for ranging authentication, even if this has been also been proposed for Galileo [100] and will be analyzed for Galileo Second Generation (G2G) [1].

Chapter 3

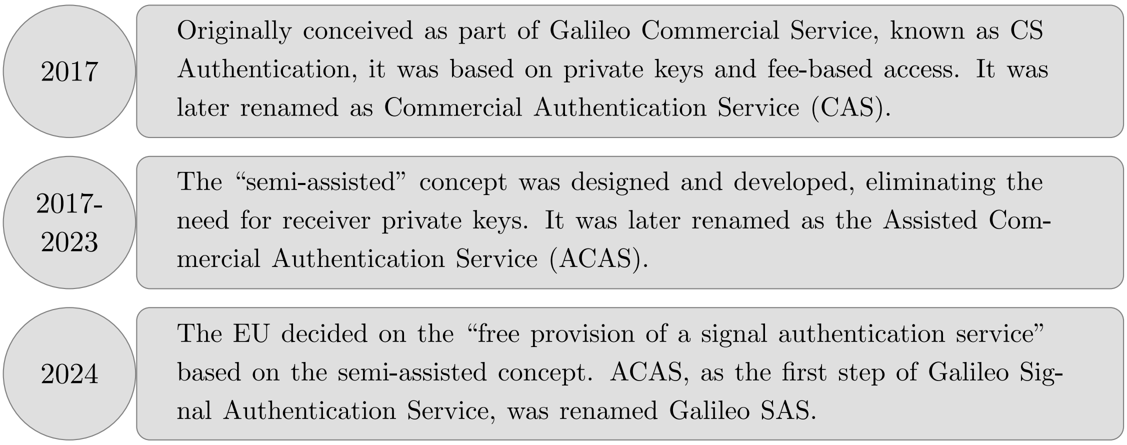
Galileo Signal Authentication Service

3.1 Introduction

From the original CS of Galileo, the European GNSS, several new services have emerged. The recently launched OSNMA in E1-B provides an affordable way to authenticate the PVT using a NMA scheme [47]. On the other hand, the HAS, currently being deployed in E6-B, enables PPP worldwide by providing orbit, clock and bias corrections [49]. These “added-value” services were intended to be offered for a fee, but they have been finally supplied for free [101].

A third service, known as SAS, which implements SCE in E6-C, is currently being developed by the EC [102]. This service will provide protection at chip-level by encrypting the spreading codes, which allows a greater level of protection against malicious attacks such as spoofing, which has been the subject of many analysis [41], [43], [60]. Together with the OSNMA, it aims to offer a fully secure solution for authenticating the PVT, without the need of modifying the current Galileo signal plan [103]. The background of SAS is illustrated in Figure 3.1 [2].

In this chapter, we review the concept of SAS and its adaptation for Galileo. We describe the main parameters involved in this new service and briefly explain the basics of the OSNMA service, upon which SAS relies. In addition, we detail the main cryptographic operations and the mechanisms envisaged for the authentication of the PVT.



2017

Originally conceived as part of Galileo Commercial Service, known as CS Authentication, it was based on private keys and fee-based access. It was later renamed as Commercial Authentication Service (CAS).

2017-2023

The "semi-assisted" concept was designed and developed, eliminating the need for receiver private keys. It was later renamed as the Assisted Commercial Authentication Service (ACAS).

2024

The EU decided on the "free provision of a signal authentication service" based on the semi-assisted concept. ACAS, as the first step of Galileo Signal Authentication Service, was renamed Galileo SAS.

Figure 3.1: SAS background.

3.2 SAS Concept

In SAS, the need of storing any secret key is avoided, while providing the user receiver with the ability to operate autonomously for long periods of time. This is achieved by making available fragments of the encrypted E6-C signal in the GSC, prior to the transmission of the signal. To prevent spoofers from generating a fake signal once these fragments are downloaded, these fragments of the encrypted E6-C signal, known as ECSs, are re-encrypted with a key yet to be disclosed, resulting in the so-called RECSs, which also include the broadcast time of the corresponding fragments. Therefore, no user, including a spoofer, will be able to decrypt these sequences before the disclosure of the key used to re-encrypt the E6-C signal.

The use of the TESLA key provided by the OSNMA protocol is the chosen option in SAS, which has the convenience to be already available to the user in the E1-B open signal. However, since its delivery is uncoupled from the reception of the E6-C signal, the receiver must store the required samples to perform the correlation, which will be done a-posteriori once the corresponding RECS is successfully decrypted.

Therefore, once the E6-C signal is broadcast, the receiver records a snapshot of samples at the time where the Received Encrypted Code Sequence ($\mathrm{ECS}^{\mathrm{R}}$) is expected, and waits for the related key to be disclosed in the E1-B signal. Once disclosed, the user receiver can decrypt the downloaded RECS to obtain the corresponding ECS and then perform the correlation with the samples previously recorded from the E6-C signal. If a correlation peak is detected and certain conditions are met, the signal can be successfully authenticated [1].

The computed solution from SAS is also useful for the the initialisation of the time synchronisation required by OSNMA, since the RECSs files are designed to include the transmission time associated to the corresponding ECS of the E6-C signal, that can be used to resynchronise the receiver [104].

The autonomy of the receiver to operate in standalone mode can be increased by making available a larger amount of the RECSs in advance, which eliminates the need of a continuous receiver-server communication. Of course, the autonomy of the receiver will depend on its storage capacity, which will ultimately determine the number of the RECSs that can be downloaded in its memory.

The predefined lengths of the RECS is one of the key parameters in the SAS design, since it determines the duration of the signal fragment used in the acquisition correlation. Together with the frequency bin size used for the Doppler search, if any, they will define the search space of the acquisition and, therefore, the ability to find the correlation peaks from the CAF, from which we generate the pseudoranges and the authenticated PVT solution.

The predefined instants at with the RECSs are chosen, i.e., the distance between to consecutive sequences is another key parameter in SAS, since it will determine how often the receiver can compute an authenticated solution. It has also significant impact on the computation of a solution when multiple integrations between different periods are performed.

It should be noted that, by default, SAS operates in snapshot mode, so the receiver will obtain the PVT solution directly from the correlation between the digital samples obtained from the GNSS receiver front-end and the local replica (i.e. the ECS) performed in the acquisition stage. Unlike conventional receivers, snapshot ones do not demodulate the data from the signal to compute the PVT; the satellite data is typically obtained from Assisted GNSS (A-GNSS) or another channel.

This operative is illustrated in Figure 3.2 (from the system side) and Figure 3.3 (from the receiver side). In this particular illustration, where each TESLA key (in a hashed form) is employed to decrypt a unique RECS, the RECS Period aligns with the duration of an OSNMA I/NAV frame (30 seconds), which is the frequency at which the keys are disclosed in the E1-B signal. Alternative configurations may also be considered, where each key is used to decrypt multiple RECSs [1], [4].

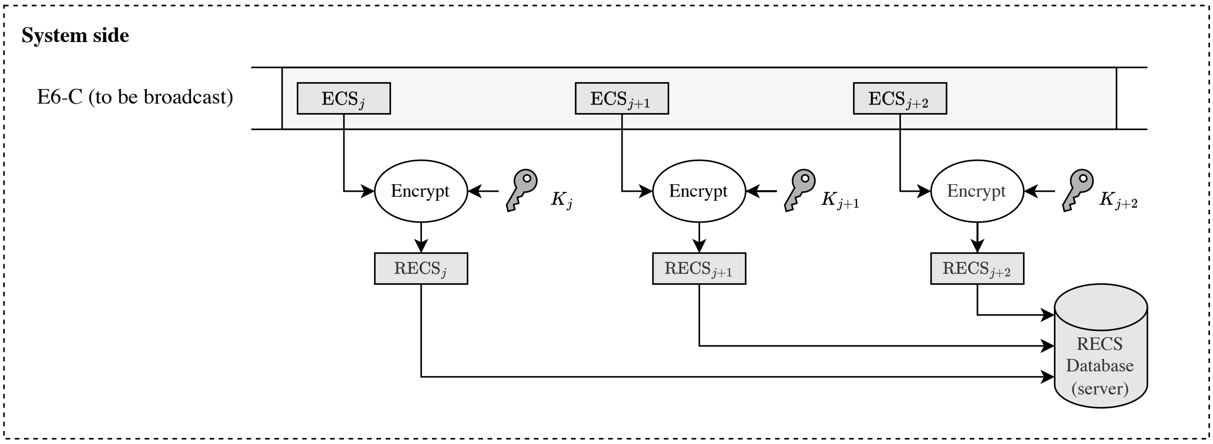


Figure 3.2: Schematic representation of the SAS operation (system-side).

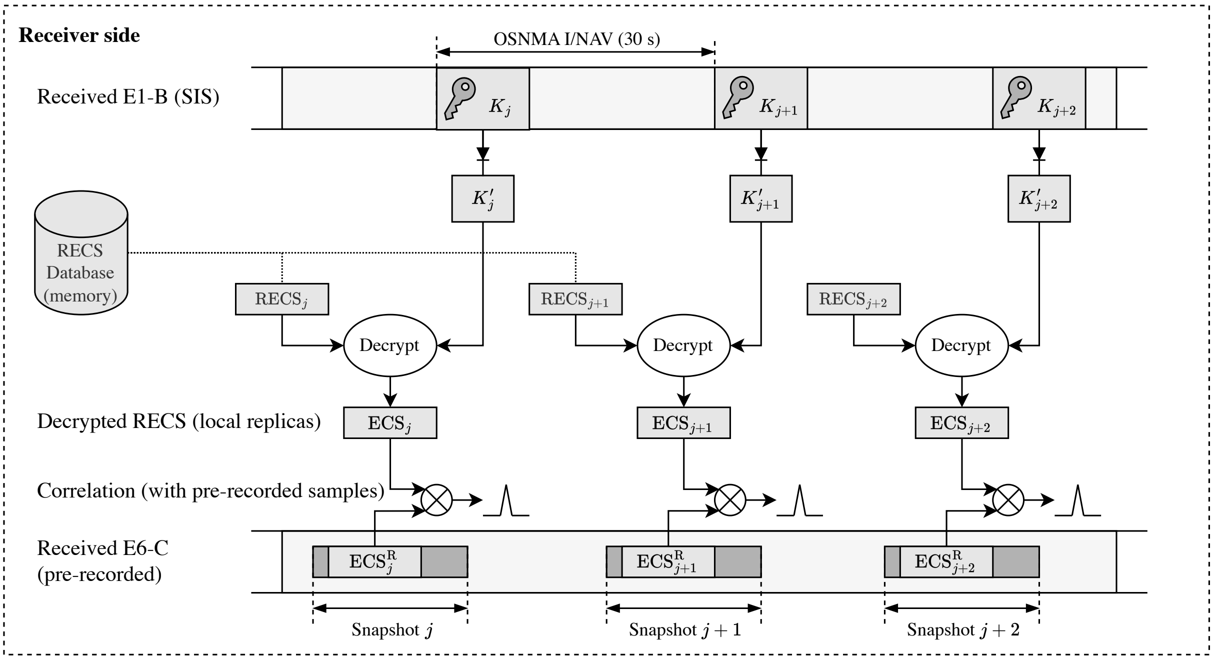


Figure 3.3: Schematic representation of the SAS operation (receiver-side).

3.3 Galileo Signals used in SAS

As the operation of SAS relies on the E1 and E6 Galileo open signals, this section aims to provide a summary of their main characteristics. Both signals are composed of a data component (-B) and a pilot component (-C), which share the same frequency band, along with the Galileo Public Regulated Service (PRS) signals[1].

The most relevant characteristics of the E1 and E6 open signals are summarized in

[1] The Galileo PRS is an encrypted navigation service restricted to government authorized users.

Table 3.1. The full specification of these signals can be found in the Galileo Interface Control Document (ICD) [105].

Parameter	**Galileo E1-B**	**Galileo E1-C**	**Galileo E6-B**	**Galileo E6-C**
Carrier frequency (f_c)	1575.42 MHz		1278.75 MHz	
Frequency band	1559-1591 MHz		1260-1300 MHz	
Spreading modulation	CBOC(6,1,1/11)		BPSK(5)	
Code chip rate (R_c)	1.023 Mcps		5.115 Mcps	
Primary code length (L)	4092 chips		5115 chips	
Code family	Random codes		Memory codes	
Signal component	Data	Pilot	Data	Pilot
Secondary code length	-	25 chips	-	100 chips
Data rate (R_s)	250 sps	-	1000 sps	-
Minimum received power	-157 dBW		-155 dBW	

Table 3.1: Main parameters for Galileo E6-C and E1-B signals.

3.3.1 BPSK and BOC signals

Traditional GPS signals use Phase Shift Keying (PSK) modulation schemes, particularly the basic BPSK modulation, which simplifies signal processing in the receiver. The evolution of GNSS has led to the analysis of new modulation techniques that offer higher performance, especially in terms of positioning accuracy [106]. Consequently, BOC modulation [107] has been adopted for next-generation radionavigation systems, including modernized GPS signals and Galileo.

While BPSK modulations concentrate power in the center of the spectrum, BOC modulations allow for widening the signal spectrum (i.e., increasing the effective or the Gabor bandwidth). This offers improved performance and the opportunity for spectrum sharing among different GNSSs. A wider Gabor bandwidth results in a sharper autocorrelation peak, thereby enhancing positioning accuracy. However, the main drawback is the increased complexity for the receiver, especially due to the occurrence of secondary peaks that may hinder the identification of the main correlation peak [108].

BOC modulation involves multiplying a pseudo-random code $c_p(t)$ with a subcarrier having a rate R_{sc} equal to or higher than the code chip rate R_c, denoted as BOC(R_{sc}, R_c). However, it is common to express these values relative to the fundamental frequency used as a reference ($R_{c,\text{ref}} = 1.023$ MHz). In this case, the BOC signal is denoted as BOC(m, n), where $m \doteq R_{sc}/R_{c,\text{ref}}$ and $n \doteq R_c/R_{c,\text{ref}}$. This multiplication results in the so-called cosine-phased or sine-phased BOC signals[2], defined respectively as

$$\text{BOC}_{\cos}(m,n) = c_p(t)\,\text{sign}\left[\cos(2\pi R_{sc}t)\right] \tag{3.1}$$

$$\text{BOC}_{\sin}(m,n) = c_p(t)\,\text{sign}\left[\sin(2\pi R_{sc}t)\right] \tag{3.2}$$

Essentially, a BOC signal can be described as an alternating sequence of "+1s" and "-1s". For example, the simplest form, BOC$(1,1)$, involves multiplying each chip shaping pulse (at rate R_c) by "+1" and "-1" (in the case of sine-phased) or by "-1" and "+1" (for cosine-phased). Higher-order BOC modulations are also implemented to further exploit autocorrelation properties, although at the expense of increased complexity for the receiver. This complexity arises from the need to address false locks as a result of the increased number of secondary peaks. In fact, the number of peaks in a BOC autocorrelation function is directly proportional to its modulation order.

3.3.2 Galileo E1 open signal

The E1 open signal is composed of two components:

Data component (E1-B) – It transmits the navigation message I/NAV. This is achieved by modulating the D_{E1-B} symbols (with a symbol rate defined by $R_{s,E1-B} = 250$ sps) into the PRN chip sequence C_{E1-B} (with a chip rate defined by $R_{c,E1-B} = 1.023$ Mcps and a length of $L_{E1-B} = 4092$ chips). This component can be expressed as

$$e_{E1-B}(t) = \sum_{i=-\infty}^{\infty} C_{E1-B}[i \bmod L_{E1-B}]D_{E1-B}\left[\text{floor}\left(i\frac{R_{s,E1-B}}{R_{c,E1-B}}\right)\right]\Pi\left(R_{c,E1-B}t - i\right) \tag{3.3}$$

where $i \bmod L$ equals i modulo L, floor rounds to the nearest lower integer, and $\Pi(t)$

[2] If not specified, the BOC signal is considered sine-phased by default.

is the rectangular pulse of unit length.

Pilot component (E1-C) – It comprises PRN chip sequence C_{E1-C} (with the same chip rate and length as E1-B). This component can be expressed as

$$e_{E1-C}(t) = \sum_{i=-\infty}^{\infty} C_{E1-C}[i \bmod L_{E1-C}]\Pi\left(R_{c,E1-C}t - i\right) \tag{3.4}$$

Each of these components is modulated by a subcarrier. The modulation employed is the Composite Binary Offset Carrier (CBOC), which is a particular implementation of Multiplexed Binary Offset Carrier (MBOC) modulation. It results from multiplexing a narrowband BOC(1,1) signal, denoted as $sc_a(t)$, with a wideband BOC(6,1) signal, denoted as $sc_b(t)$:

$$sc_a(t) = \text{sign}\left[\sin\left(2\pi R_{s,E1,a}t\right)\right] \tag{3.5}$$

$$sc_b(t) = \text{sign}\left[\sin\left(2\pi R_{s,E1,b}t\right)\right] \tag{3.6}$$

where $R_{s,E1,a} = 1.023$ Mcps and $R_{s,E1,b} = 6.138$ Mcps.

Ten elevenths of the power are allocated to the higher frequency subcarrier, while the remaining one eleventh is allocated to the lower one. Therefore, the resulting normalised (unit mean power) baseband E1 open signal broadcast for a given satellite is given by

$$s_{E1}(t) = \frac{e_{E1-B}(t)}{\sqrt{2}}\left[\alpha_{E1}sc_a(t) + \beta_{E1}sc_b(t)\right] - \frac{e_{E1-C}(t)}{\sqrt{2}}\left[\alpha_{E1}sc_a(t) - \beta_{E1}sc_b(t)\right] \tag{3.7}$$

where $\alpha_{E1} = \sqrt{\frac{10}{11}}$, $\beta_{E1} = \sqrt{\frac{1}{11}}$.

The E1 open signal generation is schematized in Figure 3.4 [105].

3.3.3 Galileo E6 open signal

The E6 open signal is composed of two components, the data and the pilot, which are described next.

Data component (E6-B) – It transmits the navigation message C/NAV. This is achieved by modulating the D_{E6-B} symbols (with a symbol rate defined by $R_{s,E6-B} = 1000$ sps) into the PRN chip sequence C_{E6-B} (with a chip rate defined by $R_{c,E1-B} = 5.115$ Mcps and a length of $L_{E8-B} = 5115$ chips). This component

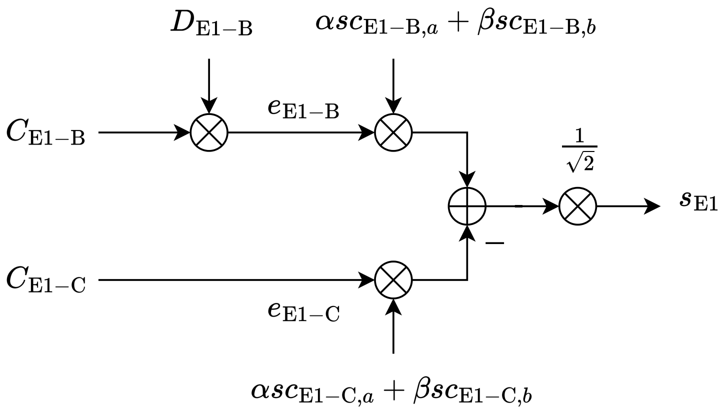


Figure 3.4: Galileo E1 open signal generation.

can be expressed as

$$e_{E6-B}(t) = \sum_{i=-\infty}^{\infty} C_{E6-B}[i \bmod L_{E6-B}] D_{E6-B} \left[\text{floor} \left(i \frac{R_{s,E6-B}}{R_{c,E6-B}} \right) \right] \Pi \left(R_{c,E6-B} t - i \right) \tag{3.8}$$

where $i \bmod L$ equals i modulo L, floor rounds to the nearest lower integer, and $\Pi(t)$ is the rectangular pulse of unit length.

Pilot component (E6-C) – It comprises the PRN chip sequence $C_{E6\text{-}C}$ (with the same chip rate and length as E6-B). This component can be expressed as

$$e_{E6-C}(t) = \sum_{i=-\infty}^{\infty} C_{E6-C}[i \bmod L_{E6-C}] \Pi \left(R_{c,E6-C} t - i \right) \tag{3.9}$$

In this case, the modulation used is the BPSK, specifically the BPSK(5). A BPSK(n) modulation indicates that the chip rate is n times higher than the reference rate $R_{c,\text{ref}}$. Therefore, the resulting normalised (unit mean power) baseband E6 open signal broadcast for a given satellite is given by

$$s_{E6}(t) = \frac{e_{E6-B}(t)}{\sqrt{2}} - \frac{e_{E6-C}(t)}{\sqrt{2}} \tag{3.10}$$

The E6 open signal generation is schematized in Figure 3.5 [105].

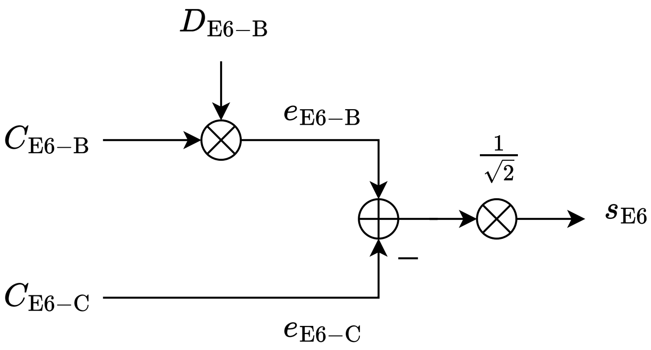


Figure 3.5: Galileo E6 open signal generation.

More details about the implementation of E6-B/C receivers can be found in [109].

3.3.4 Comparison of Galileo E1 and E6 signals

A convenient way to compare E1 and E6 signals is to compute the Autocorrelation Function (ACF) of their chip pulses, which are provided in (2.8) and (2.9) from Section 2.2.2. This is shown in Figure 3.6. The ACF of the underlying components of the E1 signal, namely, the BOC(1,1) and BOC(6,1), provided by (2.10) and (2.11), are also shown.

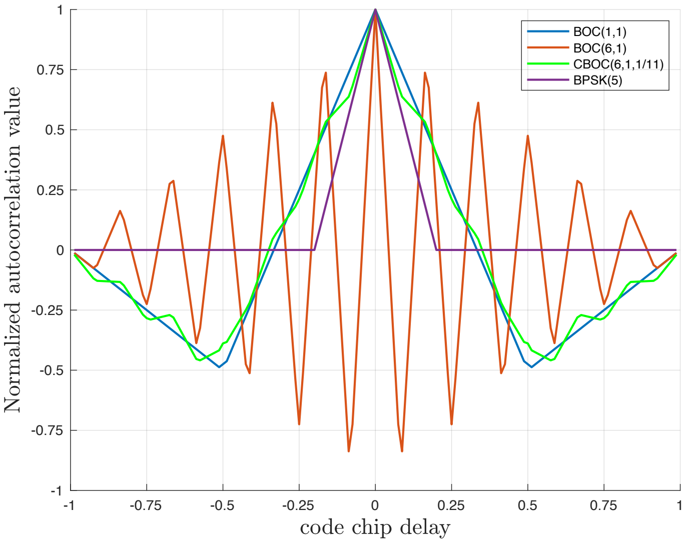


Figure 3.6: Autocorrelation functions of the chip pulses used in Galileo E1 and Galileo E6.

As can be observed, the higher BOC order implies narrower correlation peaks. However, the CBOC(6,1,1/11) used in Galileo E1 is mainly driven by the BOC(1,1) component due to its power allocation profile. When compared to the BPSK(5) used in the Galileo E6 signal, we observe that the main correlation peak is slightly narrower despite using a much higher rate. This is one of the main advantages of using BOC signals, as depicted in Section 3.3.1.

3.4 SAS Specification

As SAS is currently under development, its specification has been continuously evolving. The analysis carried out in this thesis is based on the initial specification provided in the context of the PAULA project [103]. The parameters involved and the structure of the RECS files are described next, based on this initial specification.

Since then, a new specification (v1.2) has been published [110], introducing new parameters and definitions. However, the underlying concepts and their application to SAS remain essentially unchanged and, therefore, do not impact the results and conclusions obtained in this thesis. In Section 3.5, we describe the main updates according to Specification v1.2, which is the latest specification available at the time of writing.

3.4.1 Parameters Definition

Each of the RECS files downloaded by the receiver starts with a header that defines the parameters of the service, the most relevant of which are summarized in Table 3.2:

The RECS Length in number of chips, denoted $N_{c,\text{RECS}}$ determines the duration of the signal fragment used in the acquisition correlation: the longer the RECS is, the higher the processing gain will be and, therefore, the lower the C/N_0 the receiver will be able to operate. Of course, the downside of working with large RECS is the increase in the size associated with the files to be downloaded and stored and, consequently, the reduction in the autonomy of the user's receiver. Additionally, T_{RECS} is defined to specify the duration in seconds.

The RECS Period, denoted τ_{RECS}, defines the distance between two consecutive RECS, and it determines how often the receiver can compute an authenticated PVT solution. It has also significant impact on the computation of a solution when multiple integrations between different periods are performed.

The RECS Offset, denoted δ_{RECS}, and the RECS Maximum Random Delay, denoted $\text{D}\tau_{\text{max}}$, are used to delay and randomize the position of the RECS within a given period; finally, the RECS Key Delay, denoted D_K, is used to determine the delay between the OSNMA key and the related RECS (in multiples of I/NAV subframes, i.e., 30 s).

Notation	**Definition**
$N_{c,\text{RECS}}$	RECS Length specified in chips.
T_{RECS}	RECS Length specified in seconds.
τ_{RECS}	RECS Period specified in seconds.
δ_{RECS}	RECS Offset specified in seconds.
$\text{D}\tau$	RECS Random Delay specified in seconds.
$\text{D}\tau_{\text{max}}$	RECS Maximum Random Delay specified in seconds.
$t_{\text{RECS-start}}$	Start time of the RECS File in seconds.
L_{RECS}	RECS File Length specified in seconds.
D_K	RECS Key Delay specified in I/NAV subframes (30 seconds).

Table 3.2: Definition of SAS parameters as defined in the initial specification.

3.4.2 RECS and BGD Files

As stated in Section 3.2, the RECSs are made available on the GSC server, so any receiver can download them to use later for SAS. For this purpose, these sequences are encapsulated as files together with other useful information to be processed for any SAS-compatible receiver.

Each RECS file is composed of a header and a body, which includes the chips of the RECS. In the first stages of the definition of SAS, this header was defined to include the broadcast time of the RECS and all RECS parameters defined in Section 3.4.1, including the RECS Offset, the RECS Length, the RECS Period or the RECS Maximum Random Delay. Also, as originally designed, the header includes the delay between delay between the RECS I/NAV subframe and the OSNMA key.

In addition to the RECSs files, the server also stores information about the E1/E5b

I/NAV E6-C estimated Broadcast Group Delays (BGDs), that is, the satellite group delays between the E6-C signal component and the I/NAV E1/E5b ionosphere-free combination. This allows to assist the SAS receiver to estimate the satellite clock offsets for E6-C.

As for the RECSs, the BGDs are also made available on the GSC server as files containing such estimations. Specific header and body content can be consulted in [103].

3.5 Updates in SAS Specification v1.2

In the latest specification available at the time of writing [110], an updated approach has been proposed. The approach followed is described in detail below.

The system provides various 16-ms RECS generated every 200 ms, including randomization and different delays with respect to the OSNMA key, according to new parameters defined as KDI and RAND. This is illustrated in Figure 3.7. Each 16-ms RECS is stored on the SAS server in a unitary RECS file. The header specification can be found in [110].

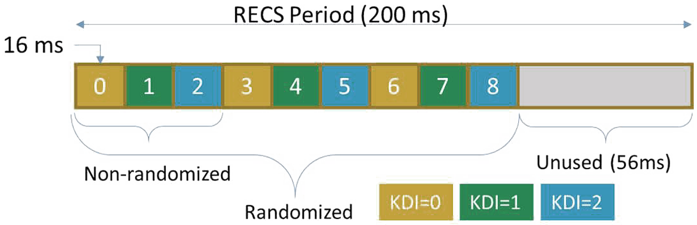


Figure 3.7: Structure of a RECS period and its combinations according KDI and RAND parameters.

To obtain the desired RECS files from the server, the user is expected to submit a query via HTTPS. The parameters to be specified in the request include the start time, duration, period, number of satellites, length of the RECS, and the randomization parameters (KDI and RAND).

Upon reception of the query, the SAS server assembles all pre-generated RECS files into a single binary file per satellite, with each parameter taking the value specified in the query. These aggregated files are created by the byte-by-byte concatenation of the unitary RECS files stored on the server, as described previously. More detailed information can be found in [110].

3.6 Relying on Galileo OSNMA

A fundamental aspect of SAS is the use of TESLA keys provided by Open Service Navigation Message Authentication (OSNMA). Hence, a brief description of OSNMA is provided in this section, but the interested reader can refer to [47] and [101] for more detail. An initial analysis was conducted in [111], highlighting the added value of authentication in the Galileo navigation message.

OSNMA is a Galileo service, the first of its kind, that aims to mitigate GNSS vulnerabilities, by including cryptographic protection of the navigation message. This allows a GNSS receiver to verify the authenticity of the data, but also ensures that the satellite navigation data come from a trusted source, that is, the Galileo system. OSNMA is, therefore, a data-level authentication service.

As its name suggests, this Navigation Message Authentication is implemented for the Open Service of Galileo. This approach is particularly important for civil users of open signals, enabling them to authenticate GNSS data and helping them protect against spoofing, provided that their receivers have the authentic public key.

3.6.1 The TESLA Protocol

Due to the multicast (one-to-many) transmission nature of a GNSS, an authentication mechanism based on asymmetric cryptography seems to be the natural choice. However, using a conventional digital signature appended to the navigation message would imply a prohibitive increase in bandwidth use. Instead, Galileo OSNMA is based on the TESLA protocol, which generates a chain of secret keys using a one-way function and used them in reverse order.

The TESLA protocol requires low computational overhead for the generation and validation of the authentication information, but also requires low communication overhead, which allows efficient use of the available bandwidth. Furthermore, it offers a high tolerance to data loss that suits the GNSS receiver operating in harsh scenarios with reduced visibility [112]. All of this makes TESLA really adapted for GNSS constraints.

To authenticate the plaintext navigation message, TESLA is based on the transmission of a Message Authentication Code (MAC), truncated with a few bit tags. These tags are generated based on symmetric cryptography with a TESLA key yet to be disclosed, so the key is still secret to the receiver. This key belongs to a chain generated through a one-way hash function. The chain starts with a secret random seed key and ends with a TESLA root key that is public and certified as authentic [47]. Due to the one-way nature of the function, each element of the chain can be constructed by hashing the previous element. Furthermore, the hash function cannot be used to predict keys. This requires that the receiver has certain information (specifically, the root key) that is certified as correct, independently of the data transmitted through the Signal-In-Space (SIS).

The processing logic of Galileo OSNMA is summarized below and schematized in Figure 3.8 [113].

- The receiver obtains from the SIS the navigation message, as well as and the corresponding OSNMA data, which includes the tag, the TESLA chain key and the TESLA root key.
- The TESLA root key is authenticated using a public key already available at the receiver.
- The TESLA chain key is authenticated with the TESLA root key or with a previously authenticated key from the TESLA chain.
- The tag is re-generated locally with the verified TESLA chain key and the received navigation data.
- The computed tag is compared with the previously received tag.
- If the comparison is successful, the receiver can authenticate the navigation message.

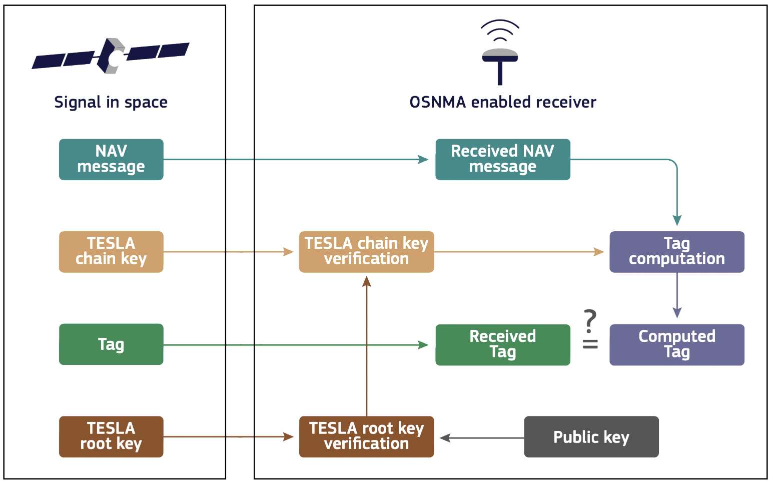


Figure 3.8: Galileo OSNMA processing logic © EUSPA.

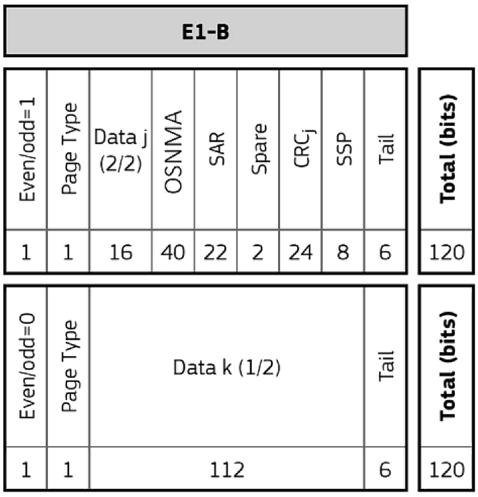
The TESLA key used to compute the MAC is then disclosed after about 30 seconds, corresponding to an I/NAV subframe duration). The receiver can then proceed with the verification of the tag that ensures the authenticity of the navigation message. This authentication mechanism requires the OSNMA receiver to be loosely synchronized at start-up with an external time reference (Galileo System Time (GST)), with an accuracy between some seconds and a few minutes, depending on the mode of operation. This ensures a secure implementation of the TESLA protocol, since it guarantees that an spoofer cannot intercept the key, manipulate the data, and reply the signal. As sources of loose reference time, the receiver can use an internal real-time clock, but also a secure network connection with time transfer capability [113].

3.6.2 Particularization for OSNMA

The TESLA used for OSNMA has been adapted for transmission through Galileo, as detailed in the proposal published by the Galileo program [47]. This optimization includes the use of a shared one-way chain (that is, a single key) by all Galileo satellites, so the GNSS users will be able to receive the key by any satellite in view. This allows the possibility to authenticate non-Galileo GNSS satellites which do not transmit OSNMA data with the data retrieved from satellites transmitting OSNMA. This innovative aspect of Galileo OSNMA is known as cross-authentication. This approach minimizes the Authentication Error Rate (AER) and Time Between Authentications (TBA), crucial for maintaining high navigation performance and robustness against attacks.

Regarding its implementation, OSNMA makes use of 40 reserved bits in the Galileo E1-B data message (I/NAV), as shown in Figure 3.9, extracted from the service's ICD [114]. The 40 bits of the OSNMA field are divided into two parts: the HKROOT section (first 8 bits), which includes the global headers and the Digital Signature Message (DSM), and the Message Authentication Code and Key (MACK) section (next 32 bits), which contain the MACs and associated keys, which are delivered later. The rest of the navigation message bits remain unencrypted.

OSNMA is designed to be fully compatible with the existing GNSS infrastructure, requiring minimal changes to the Galileo system deployed. Indeed, it is conceived to ensure that authenticated users experience navigation performance on par with non-authenticated users, in terms of accuracy, availability, and time to first fix, even under challenging reception conditions. The unpredictability of OSNMA symbols has led also



Even/odd=1	Page Type	Data j (2/2)	OSNMA	SAR	Spare	CRC$_j$	SSP	Tail	Total (bits)
1	1	16	40	22	2	24	8	6	120

Even/odd=0	Page Type	Data k (1/2)	Tail	Total (bits)
1	1	112	6	120

Figure 3.9: Galileo OSNMA field in I/NAV word.

to new techniques for anti-reply [115].

After the proposal published in 2016 [47], the Galileo OSNMA implementation decision was taken in February 2017. The first OSNMA data was broadcast in November 2020, and the final SIS ICD of the service was published in December 2022 —the latest updated available at the time of writing was released in February 2023 [114]. The OSNMA public observation test phase was started in August 2023, and the OSNMA initial service declaration is aimed for early 2024. Furthermore, OSNMA will be offered as a free-of-charge service.

3.7 Cryptographic Operations

In SAS a certain number of cryptographic operations are involved at the receiver, which are

- the generation of the RECS decryption key,
- the generation of the randomization parameter,
- and the RECS decryption.

These operations are briefly described next, but the interested reader can refer to [116] and [1] for more details.

Generation of the RECS decryption key – Once the TESLA key K_j belonging to block j is received and verified by the OSNMA keychain, the acrecs decryption key K'_j is generated as follows: $K'_j = \mathrm{SHA256}(K_j)$, where SHA256 is the one-way hash function Secure Hash Algorithm (SHA)-256 —this allows to decrypt the RECS only when the TESLA related key is disclosed in the E1-B signal.

Generation of the randomization parameter – Since a single TESLA key is used for each 30-second interval, but additional offsets might be needed for each satellite within a RECS period, the (Advanced Encryption Standard (AES) cipher is initially employed to create an adequately large ciphertext. From this ciphertext, the necessary random time offsets are derived. The ciphertext is generated as follows: $(C_1, \ldots, C_N) = \mathrm{AES256_{OFB}}(K'_{j+D_{K'}}, 0, \mathrm{IV})$, where $\mathrm{AES256_{OFB}}$ is the AES cipher in Output Feedback (OFB) mode and configured for 256-bit keys, and $\mathrm{IV} = \mathrm{trunc}(128, \mathrm{SHA256}(\mathrm{GST}_{SF,j}))$ is the initialization vector, where $\mathrm{trunc}(n, p)$ is the truncation function that retains the n Most Significant Bits (MSBs) of the input p and $\mathrm{GST}_{SF,j}$ is the 32-bit GST associated to the TESLA key K_j. Finally, the ciphertext block array $(C_1, \ldots, C_N)$ is assigned to the random time offsets $D\tau$.

RECS decryption – For the decryption key $K'_{j+D_{K'}}$ and for the k-th satellite, the RECS decryption process is performed as follows: $\mathrm{ECS}^k_{j,i} = \mathrm{AES}^{-1}_{\mathrm{CBC}}(K'_{j+D_{K'}}, \mathrm{RECS}^k_{j,i}, \mathrm{IV})$, where $\mathrm{AES_{CBC}}$ is the inverse cipher AES configured for 256-bit keys in Cipher Block Chaining (CBC) mode.

3.8 SAS drawbacks

One of the main advantages of SAS is its ability to provide signal authentication without modifying the current Galileo signal structure. This is made possible because the constellation already includes the key components for this new service: the TESLA keys provided by OSNMA in the E1-B signal and the E6-C pilot component, which will be encrypted in the near future.

However, this also presents a key drawback in implementing SAS in Galileo: the loss of the pilot component on the E6 band, which currently supports the demodulation of HAS data in E6-B through the secondary codes available in the existing E6-C. Once encrypted, E6-B based receivers would need to rely solely on the E6-B component. This limitation could be alleviated with the future introduction of Quasi-Pilot (QP) signals [117], by incorporating a pilot component in the E6 band (referred to as E6-D-P) [118].

Other drawbacks to consider include the storage requirements that a SAS-enabled receiver may face—primarily due to the RECS files downloaded from the server—and authentication latency. These issues are discussed in more detail in Section 4.2.6 and Section 4.5, respectively.

Finally, other limitations must be considered when using the E1-B signal to support the detection of the RECSs on E6-C, as analyzed in Chapter 5.

The majority of these limitations are expected to be addressed in G2G [2].

3.9 Conclusions

In this chapter, we have reviewed SAS, which is designed to provide enhanced protection at the range level by using encrypted spreading codes. Combined with OSNMA, which offers authentication of the navigation message bits, it aims to deliver a fully secure solution for GNSS receivers.

To achieve this without modifying the current signal plan of Galileo and without requiring the storage of any secret keys in the receivers, SAS will make use the E6-C encrypted signal and the TESLA keys already delivered in the E1-B open signal by the OSNMA protocol.

The operation of SAS has been described in detail, involving the use of re-encrypted sequences known as RECS, the recording of E6-C snapshots at the expected times of these RECSs, and subsequent correlation to authenticate the PVT solution. All the parameters involved in the service definition are also described.

Chapter 4

Generic Approach for SAS

4.1 Introduction

As SAS is currently being consolidated and is not yet operational, it is imperative to analyze the impact of the various parameters involved in the service, and to evaluate the performance at the signal level under different scenarios. The subtleties of SAS deserve a thorough examination to determine the benefits and limitations of potential strategies, particularly in the context of the acquisition phase. Certainly, the length and periodicity of the RECS, as well as the time reference used by the receiver, can have a great impact on signal detection capabilities and, consequently, on the authentication of the PVT through SAS. This analysis aims to elucidate these aspects and provide general guidelines for the implementation of SAS receivers. Furthermore, the results provided here can be useful in selecting the configuration of a hardware receiver and as a performance reference baseline for practical implementations.

One of the primary challenges for the receiver is identifying the RECS location within the E6-C signal. As detailed in Chapter 3, the RECS files downloaded by the receiver include the transmission times for these segments, allowing the receiver to theoretically pinpoint their exact location under ideal conditions by considering all relevant (and known) parameters. However, the receiver may not have an accurate time reference; in fact, the lower the accuracy of this time reference, the greater the uncertainty regarding the RECS location.

Depending on the envisioned approach, the receiver can use various time sources, each with its own set of implications. Given that in SAS, the receiver tracks the E1-B signal to obtain the TESLA keys needed for decrypting the RECS, it is plausible to leverage an accurate time reference from E1-B to precisely determine the ECSs' location within the E6-C signal. This approach will be analyzed in detail in Chapter 5.

In this chapter, we introduce a generic approach for SAS that covers all potential time references. For simplicity, it is assumed that the receiver uses its internal clock as the primary time reference. The accuracy of this clock significantly influences the acquisition process. In fact, the case where the E1-B time reference is employed can be considered a particular case of this broader approach, in which the receiver clock has no uncertainty and therefore is perfectly synchronized with the GST reference.

4.2 Generic Approach

Unlike a conventional GNSS acquisition procedure, where the receiver can start correlating the local replica immediately after receiving the broadcast signal of interest, the SAS receiver lacks the local replica required for correlation with the ECS. Therefore, it should record a snapshot of the E6-C samples that (hopefully) contain the corresponding ECS. This snapshot should be retained in the receiver storage until the corresponding key is revealed in the E1-B signal, at which point it can proceed with the decryption of RECS (i.e., the local replica) and consequently the a posteriori correlation.

It is evident that the size and number of these snapshots will be limited by the amount of memory storage available in the receiver. This is analyzed in Section 4.2.6.

The first step for the SAS receiver is, then, determine the expected location of the ECSs to be broadcast, according to the parameters defined in the service and the delays that may affect the transmission and reception of the signal. These will allow later determining the starting and ending point of the snapshots to be recorded.

4.2.1 ECS Location

In the following, we assume, without loss of generality, that each TESLA key is used to encrypt a unique ECS, so each j-th key corresponds to a unique p-th RECS period. The starting point of this period is given by

$$t^k_{\text{period-start},p} = t^k_{\text{RECS-start}} + (p-1)\tau_{\text{RECS}} \tag{4.1}$$

where $t^k_{\text{RECS-start}}$ is the start time of the RECS (extracted from the RECS file header) for the k-th satellite.

For a given period, each ECS could be transmitted at the beginning of this period or delayed by some amount. This delay is the sum of the RECS Offset (δ_{RECS}), and the RECS Random Delay ($\text{D}\tau$). The RECS offset is the same for all periods and satellites, while the RECS Random Delay is chosen from 0 to the RECS Maximum Random Delay ($\text{D}\tau_{\max}$), and could be different for each period and satellite. Thus, for the p-th period and the k-th satellite, the ECS is delayed with respect to the start of the period by

$$\Delta\text{ECS}^k_{\text{tx},p} = \delta_{\text{RECS}} + \text{D}\tau^k_p \tag{4.2}$$

From the receiver point of view, the ECS will be subject to further delays with respect to the start of the RECS period due to propagation delay and clock offsets. However, the latter could introduce either a delay or an advance, depending on its sign. Hence, the sum of the propagation delay and the clock offsets for the k-th satellite, called the reception delay (τ^k), is given by

$$\tau^k = \tau^k_{\text{prop}} - \delta t^k_{\text{sat}} + \delta t_{\text{rx}} \tag{4.3}$$

where τ^k_{prop} is the propagation delay from the k-th satellite, δt^k_{sat} is the k-th satellite clock offset, and δt_{rx} is the receiver clock offset.

However, it is more convenient to express this delay in terms of its span (i.e., the maximum variation it can reach) rather than its absolute magnitude.

$$\tau^k_{\text{prop}} = \tau^k_{\text{prop,min}} + \Delta\tau^k_{\text{prop}} \tag{4.4}$$

$$\delta t^k_{\text{sat}} = \delta t^k_{\text{sat,max}} + \Delta\delta t^k_{\text{sat}} \tag{4.5}$$

$$\delta t_{\text{rx}} = \delta t_{\text{rx,min}} + \Delta\delta t_{\text{rx}} \tag{4.6}$$

Hence, the reception delay can be equivalently expressed as

$$\tau^k = \tau^k_{\text{prop,min}} - \delta t^k_{\text{sat,max}} + \delta t_{\text{rx,min}} + \Delta\tau^k_{\text{prop}} + \Delta\delta t^k_{\text{sat}} + \Delta\delta t_{\text{rx}} \tag{4.7}$$

The previous terms can be grouped as

$$\tau^k_{\text{min}} = \tau^k_{\text{prop,min}} - \delta t^k_{\text{sat,max}} + \delta t_{\text{rx,min}} \tag{4.8}$$

$$\Delta\tau^k = \Delta\tau^k_{\text{prop}} + \Delta\delta t^k_{\text{sat}} + \Delta\delta t_{\text{rx}} \tag{4.9}$$

Finally, the reception delay can be rewritten as

$$\tau^k = \tau^k_{\text{min}} + \Delta\tau^k \tag{4.10}$$

Hence, from the receiver perspective, the delay with respect to the start of the p-th RECS period and a k-th satellite is given by

$$\Delta\text{ECS}^k_{\text{rx},p} = \Delta\text{ECS}^k_{\text{tx},p} + \tau^k = \tau^k_{\text{min}} + \Delta\tau^k + \delta_{\text{RECS}} + \text{D}\tau^k_p \tag{4.11}$$

Actually, it is worth noting that the only term unknown to the receiver is $\Delta\tau^k$, since the term τ^k_{min} could be estimated a priori from the propagation characteristics and the receiver and satellite clocks specifications.

An example of these delays for a given p-th period and the k-th satellite is shown in Figure 4.1, where, without loss of generality, τ_{min} is assumed to be positive and τ^k is assumed to be shorter than the RECS Period.

4.2.2 Snapshot Determination

Once the receiver has calculated the expected location for the ECSs, it can proceed to determine the starting point of the snapshots, as well as their length.

The starting time of the snapshot for the p-th period and the k-th satellite is given by:

$$t^k_{\text{snp-start},p} = t^k_{\text{RECS-start}} + (p-1)\tau_{\text{RECS}} + \delta_{\text{RECS}} + \tau^k_{\text{min}} \tag{4.12}$$

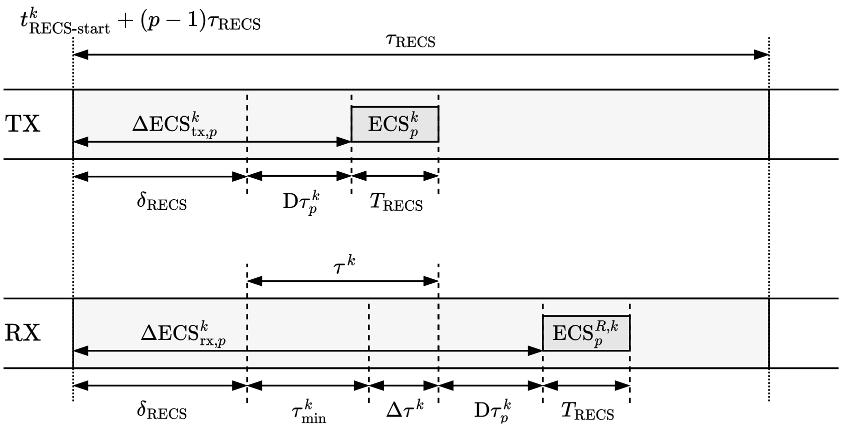


Figure 4.1: Example of ECS delays for a given p-th period and k-th satellite.

where $t^k_{\text{RECS-start}}$ is the start time of the RECS extracted from the RECS file for the k-th satellite.

The last expression takes into account only the k-th satellite. For the receiver to take into account all satellites, we rewrite the previous equation as

$$t_{\text{snp-start},p} = t_{\text{RECS-start}} + (p-1)\tau_{\text{RECS}} + \delta_{\text{RECS}} + \tau_{\text{min}} \tag{4.13}$$

where $t_{\text{RECS-start}} = \min_k(t^k_{\text{RECS-start}})$ and where τ_{min} account for the sum of the minimum (or maximum, depending on the sign) values of the terms included in the aforementioned reception delay for all satellites, that is:

$$\tau_{\text{min}} = \min_k(\tau^k_{\text{min}}) = \min_k(\tau^k_{\text{prop,min}}) - \max_k(\delta t^k_{\text{sat,max}}) + \delta t_{\text{rx,min}} \tag{4.14}$$

To calculate the length of the snapshot, we need to consider the maximum span that we can have on the reception delay. For the k-th satellite, this maximum reception delay span, denoted $\Delta\tau^k_{\text{max}}$, corresponds to the sum of the maximum spans of its terms:

$$\Delta\tau^k_{\text{max}} = \Delta\tau^k_{\text{prop,max}} + \Delta\delta t^k_{\text{sat,max}} + \Delta\delta t_{\text{rx,max}} \tag{4.15}$$

For example, considering that the propagation time in Galileo typically varies between 77 and 97 milliseconds, we could establish that $\tau^k_{\text{prop,min}} \approx 77$ ms, $\Delta\tau^k_{\text{prop}} \approx [0-20]$ ms, and therefore $\Delta\tau^k_{\text{prop,max}} \approx 20$ ms. With respect to the maximum span on the satellite clock, in Galileo, it could be estimated around 10 milliseconds (but it will depend on the status of the clock of each specific satellite), whereas the maximum span on the receiver clock will depend on its status. A perfectly calibrated clock would translate to $\Delta\delta t_{\text{rx,max}} = 0$, but it could increase by several seconds (or more) if the receiver has not been calibrated for a long period of time.

Therefore, the maximum span that we can have on the reception delay for all satellites is given by

$$\Delta\tau_{\max} = \max_{k}(\Delta\tau^{k}_{\max}) \tag{4.16}$$

Finally, the length of the snapshot is obtain as

$$T_{\text{snp}} = T_{\text{RECS}} + \text{D}\tau_{\max} + \Delta\tau_{\max} \tag{4.17}$$

where $T_{\text{RECS}} = N_{c,\text{RECS}}/R_c$ is the length of the ECS/RECS, being R_c is the chip rate of E6-C.

4.2.3 ECS Correlation

Snapshots will be stored in the receiver, waiting for the p-th key, used to encrypt the p-th ECS, to be disclosed. Once this key is revealed, the receiver can compute the corresponding random delay ($\text{D}\tau^{k}_{p}$) that has been applied for the k-th satellite and the p-th period, and shorten the snapshot that will be used for the acquisition, which will be referred to as the acquisition window hereafter.

Therefore, the starting time of the acquisition window that will be used for the correlation with the ECS corresponding to the k-th satellite and p-th period is be given by

$$t^{k}_{\text{acq-start},p} = t_{\text{snp-start},p} + \text{D}\tau^{k}_{p} = t_{\text{RECS-start}} + (p-1)\tau_{\text{RECS}} + \delta_{\text{RECS}} + \tau_{\min} + \text{D}\tau^{k}_{p} \tag{4.18}$$

It should be noted that, whilst the starting point of the snapshot is taken considering the spans for all the satellites, the starting point of the acquisition window will depend on the k-th satellite processed.

Furthermore, since the specific RECS Random Delay applied to the ECS has been computed, we no longer need to account for the RECS Maximum Random Delay ($\text{D}\tau_{\max}$).

Hence, the length of the acquisition window is given by

$$T_{\mathrm{acq}} = T_{\mathrm{snp}} - \mathrm{D}\tau_{\max} = T_{\mathrm{RECS}} + \Delta\tau_{\max} \tag{4.19}$$

It is noteworthy that the lengths of both the snapshot and the acquisition window are constant, regardless of the satellite or period considered.

As can be observed in (4.19), the maximum reception delay span is unknown by the receiver and consequently must be considered as an uncertainty that affects the length of the acquisition window. In fact, if the maximum span tends to zero, the length of the acquisition window coincides with the length of the ECS. Indeed, if there were no uncertainty at all (i.e., no span in the reception delay), one could argue that strictly speaking, the length would be exactly one sample, since the receiver will know perfectly the alignment of the sequence in the E6-C signal.

Therefore, this acquisition window length is the main parameter that will drive the performance of the acquisition procedure in terms of probability of detection and false alarm. Indeed, the larger the window, the larger the search space of the CAF, and so the probability of global false alarm. In addition, the processing time would also increase accordingly.

As described in Chapter 2, the correlation of the ECS with the corresponding decrypted RECS can be achieved with any of the conventional techniques employed in GNSS. The bidimensional search space defined by both the code delay and the Doppler frequency results in the CAF function. The correlation process is considered successful if a peak is found (i.e., the maximum of the CAF) and meets the pre-established conditions. The SAS receiver can start then the authentication procedure to authenticate the PVT.

4.2.4 Generic Acquisition Summary

The generic acquisition procedure for SAS for a given p-th RECS period described previously is summarized next. We assume that the RECSs have already been downloaded by the receiver for the desired autonomy. In addition, we assume that the receiver knows the list of satellites to be tracked at each period.

First, for the offline mode, where the TESLA keys have not yet been disclosed:

1. Estimate the minimum reception delay and maximum reception delay span according to (4.14) and (4.16), respectively. This is not necessarily done for each period, depending on the RECS Period used.
2. Obtain RECS-related parameters (from the header of the RECS file) and calculate the starting point and the length of the snapshot to be recorded, according to (4.13) and (4.17).
3. Wait until the E6-C signal is broadcast and record the snapshot in receiver storage.

Second, for the online mode, where the keys are transmitted:

4. Wait until the E1-B signal is broadcast and obtain the key related to the period.
5. Use the key to decrypt the corresponding RECS and obtain the ECS that will be used as the local replica for the correlation.
6. Use the key to obtain the corresponding RECS Random Delay and, for each k-th satellite to be processed, calculate the starting point of the acquisition window according to (4.18). Also, calculate the length of the acquisition window according to (4.19).
7. For each k-th satellite to be processed, perform the correlation of the acquisition window with the local replica for the configured search space. As a result, a CAF is obtained for each processed satellite.
8. For each k-th satellite to be processed, obtain the maximum for the CAF (i.e., the correlation peak), in addition to the corresponding code delay and Doppler frequency estimates.

An example of the acquisition procedure for a given p-th period and the k-th, $k + 1$-th, and $k + 2$-th satellites is shown in Figure 4.2, where, as in the Example shown in Figure 4.1, $\tau_{\min}$ is assumed to be positive and τ^k is assumed to be shorter than the RECS Period, without loss of generality.

In this instance, the reception delay linked to the $k+1$-th satellite is set to the minimum value (i.e., $\tau^{k+1} = \tau_{\min}$), whereas the delay for the $k+2$-th satellite is set to the maximum value (i.e., $\tau^{k+2} = \tau_{\min} + \Delta\tau_{\max}$). Consequently, the ECS associated with the $k + 1$-th satellite will be located at the start of the acquisition window, while the ECS linked to the $k + 2$-th satellite will be placed at the end.

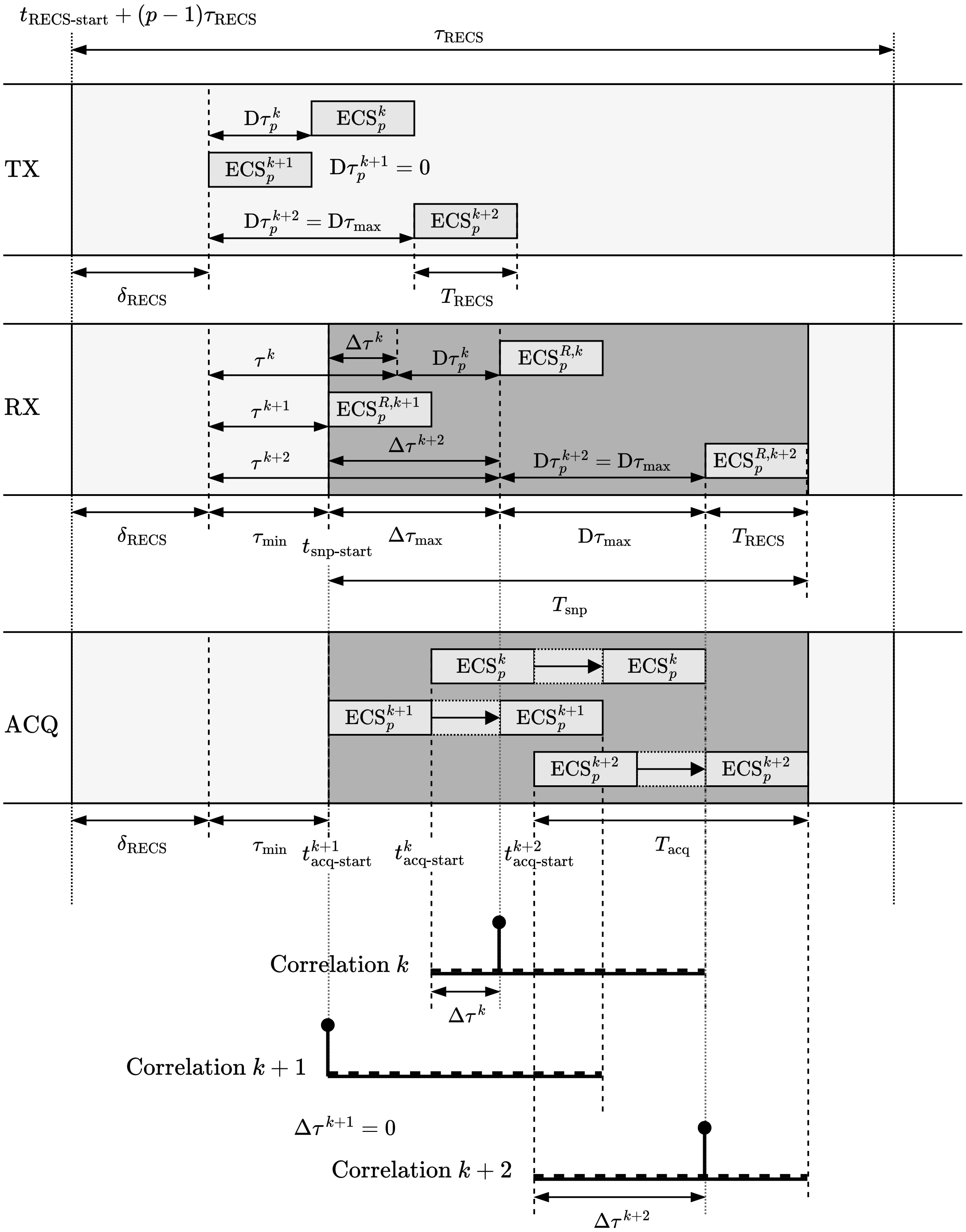


Figure 4.2: Example of SAS acquisition procedure for a given p-th period.

For the sake of clarity, a summary of all the parameters defined in this chapter and the following ones is given in Table 4.1.

Notation	Definition
$\mathrm{D}\tau_p^k$	RECS Random delay for p-period and k-th satellite.
τ_{prop}^k	Propagation delay from k-th satellite to receiver specified in seconds.
$\tau_{\text{prop,min}}^k$	Minimum propagation delay (from nearest satellite to receiver) specified in seconds.
$\Delta\tau_{\text{prop}}^k$	Propagation delay span specified in seconds.
δt_{sat}^k	Satellite clock offset for the k-th satellite specified in seconds.
$\delta t_{\text{sat,max}}^k$	Maximum satellite clock offset (from worst-case satellite) specified in seconds.
$\Delta\delta t_{\text{sat}}^k$	Satellite clock offset span for k-th satellite specified in seconds.
δt_{rx}	Receiver clock offset specified in seconds.
$\delta t_{\text{rx,min}}$	Minimum receiver clock offset specified in seconds.
$\Delta\delta t_{\text{rx}}$	Receiver clock offset span specified in seconds.
τ^k	Reception delay for k-th satellite specified in seconds.
τ_{min}^k	Minimum reception delay specified in seconds.
$\Delta\tau^k$	Reception delay span for k-th satellite specified in seconds.
$\Delta\tau_{\text{max}}^k$	Maximum reception delay span specified in seconds.
$t_{\text{RECS-start}}^k$	Starting point of the RECS for k-th satellite specified in seconds.
$t_{\text{period-start},p}^k$	Starting point of the p-the RECS period for k-th satellite specified in seconds.
$t_{\text{snp-start},p}^k$	Starting point of the E6-C snapshot for k-th satellite specified in seconds.
T_{snp}	Length of the E6-C samples snapshot specified in seconds.
$t_{\text{acq-start},p}^k$	Starting point of the acquisition window for p-period and k-th satellite specified in seconds.
T_{acq}	Length of the acquisition window specified in seconds.
T_{RECS}	RECS length in seconds.
R_c	Chip rate of the E6-C signal, unless specified otherwise.

Table 4.1: Definition and notation of parameters

4.2.5 Post-Detection

SAS is designed to operate with a single snapshot by default, from which an authenticated PVT can be obtained if the correlation with the local replica is successful (i.e., a peak is found). However, integrating a single ECS may be insufficient under harsh conditions, where the adverse effects of noise can hinder the detection of the correlation peak.

Typically, most GNSS receivers need to perform several integrations—coherent, non-coherent, or a combination of both—to decide whether the desired signal is present. In SAS, since the fragments to be combined are separated by the RECS Period (τ_{RECS}), particular attention should be paid. This separation not only complicates the computation of a PVT solution by making it harder to assign the corresponding reception time but also hampers the use of coherent integrations, as we analyze below.

The maximum coherent integration time is mainly limited by the effects of frequency Doppler and phase noise caused by receiver clock instabilities. When processing a conventional GNSS signal, which involves a consecutive concatenation of spreading codes, these effects are considered over the total coherent integration time, spanning N_c code repetitions. However, in SAS, the Doppler frequency and phase noise effects extend over N_c periods, i.e., $N_c\tau_{\text{RECS}}$, while the total coherent integration time is given by

$$T_{\text{int,coh}} = T_{\text{RECS}}N_c \tag{4.20}$$

Regarding the Doppler frequency, the frequency bin size in acquisition is typically chosen to be half the inverse of the coherent integration time. In the context of SAS, this translates to half the inverse of the total span time, given by $1/(2N_c\tau_{\text{RECS}})$. To keep this bin size practical, the RECS Period should be minimized. For instance, with $N_c = 3$ and $\tau_{\text{RECS}} = 100$ ms, the frequency bin size would be approximately 5 Hz. Given a typical frequency search range of ±5 kHz, this configuration would require 10,000 frequency bins, leading to a prohibitive acquisition time.

Regarding phase noise, the type of clock used in the receiver determines how fast the phase varies over time. As detailed in Section 2.2.7, the clock can be characterized by its short-term and long-term stabilities. If the sequence to be integrated coherently lasts only a few milliseconds (e.g., common primary spreading codes), the impact of the receiver's clock phase noise can be neglected. However, when a larger number of sequences are combined coherently (e.g., 100 or more), the phase noise effects can become visible but still limited.

In SAS, the *distance* between RECSs is the factor that determines the possible phase variations between them. Therefore, even a combination of just 2 or 3 RECS can lead to significant phase variations, making coherent integration unfeasible. Shortening the

RECS periods will limit these phase variations but at the expense of increasing the storage requirements for the receiver (see Section 4.2.6).

Therefore, if the receiver relies solely on the E6-C signal, multiple coherent integrations cannot be achieved when RECSs are transmitted very far apart from each other. In such cases, non-coherent integrations must be considered. However, to achieve performance equivalent to that provided by coherent integration, the receiver may need to combine a large number of sequences. In SAS, this could be problematic since, for a given RECS period, the receiver would require a prohibitive amount of time to provide a PVT solution. Moreover, as mentioned earlier, this would make it more challenging to assign the corresponding reception time accurately.

An example of performing multiple integrations that combine both coherent and non-coherent techniques is illustrated in Figure 4.3. In this example, a total of four ECSs are used. The RECS period is assumed to be small enough to allow for the coherent combination of two consecutive ECSs ($N_c = 2$), but not small enough to combine four of them coherently. Therefore, the results of these coherent integrations are then combined non-coherently ($N_i = 2$).

To prevent combining multiple RECS in SAS, larger correlation sequence, i.e., a RECSs/ECS with larger NChips, could be used. However, one of the downsides of such approach is the increase of the file size to be downloaded from the GSC.

4.2.6 Receiver's Storage Considerations

In SAS, the receiver must download and store the RECS files, which will determine its capacity to operate autonomously without relying on server communication. The size of these files depends on different parameters, as

- the RECS Length, determined by T_{RECS},
- the number of satellites to be used, defined by N_{sats},
- and the RECS Period, determined by τ_{RECS}, which will define the time between authentications.

Considering that one bit per chip is used, the approximate size of the RECS files for

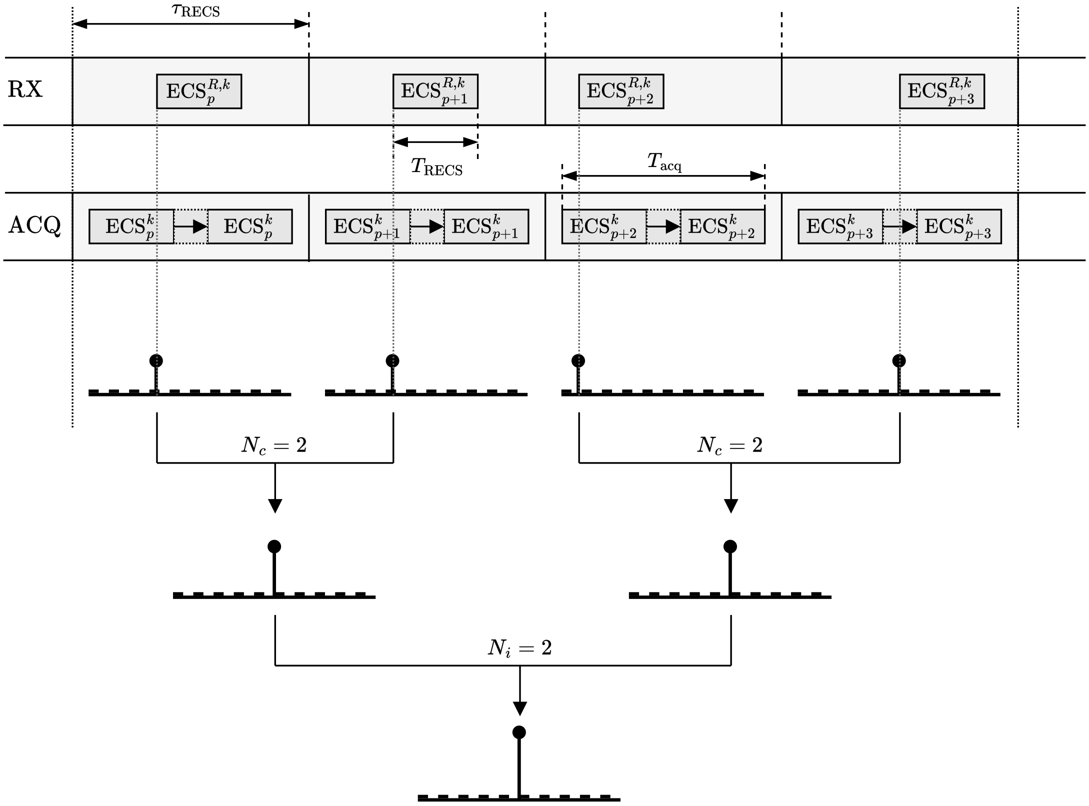


Figure 4.3: Multiple integrations in SAS.

the required autonomy, denoted T_{autonomy}, is calculated as

$$N_{\text{bits,RECS}} = N_{\text{sats}} \frac{T_{\text{autonomy}}}{\tau_{\text{RECS}}} T_{\text{RECS}} R_c \tag{4.21}$$

For example, we assume that an SAS receiver must authenticate its position every 30 seconds, using RECSs for 18 satellites and using 8 ms per correlation. If such a receiver required an autonomy of 24 hours, the size occupied by the RECS files will be approximately 252.9 MB (where 1 MB stands for MegaByte and equals to 2^{20} bytes). Other uses cases are shown in Table 4.2.

Parameter	**Use case 1**	**Use case 2**	**Use case 3**
T_{autonomy}	1 h	24 h	168 h
N_{sats}	12	18	24
τ_{RECS}	300 s	30 s	15 s
T_{RECS}	2 ms	8 ms	16 ms
$N_{\text{bits,RECS}}$	0.2 MB	252.9 MB	9440.8 MB

Table 4.2: Examples of RECS files storage needs.

It should be noted that, in practice, a small overhead for each RECS files should be considered.

Additionally, the receiver must also pre-record the required samples from the E6-C signal corresponding to the time of authentication, which is related to the corresponding TESLA key for a given period. This will allow the receiver to perform the a posteriori correlation between these samples and the corresponding ECS. The number of pre-recorded samples required has been determined in Section 4.2.2 where the length of the E6-C snapshots is calculated, but depends mainly on two parameters:

1. The maximum span assumed for the reception delay.
2. The RECS Maximum Random Delay, since this parameter is only known when the RECS is decrypted with the disclosed TESLA key.

A large RECS Maximum Random Delay will increase the randomization of the ECSs in the E6-C signal, but would increase the duration of the snapshot. Therefore, it is recommended to keep this parameter as low as possible to avoid increasing the storage requirement of the receiver.

Hence, for a given sampling rate f_s, the size in bits of a single E6-C snapshot will be given by

$$N_{\text{bits,snp}} = T_{\text{snp}} f_s \tag{4.22}$$

where T_{snp} is provided by (4.17).

Assuming that no Maximum RECS Random Delay is applied ($\mathrm{D}\tau_{\mathrm{max}} = 0$) and that the receiver has access to a perfectly-accurate time reference (i.e., $\Delta\delta_{\mathrm{rx}} = 0$), the size occupied by the E6-C snapshot will be roughly 0.9 MB for 8-ms RECS and 10 MHz sampling rate. For this calculation we have considered a reasonable value of 20ms for both the maximum span of the propagation delay and satellite clocks (i.e., $\Delta\tau_{\mathrm{prop}} = \Delta\delta_{\mathrm{sat}} = 20$ ms), which leads to 48 ms snapshot duration. Other uses cases are shown in Table 4.3.

Parameter	**Use case 1**	**Use case 2**	**Use case 3**
T_{RECS}	8 ms	8 ms	16 ms
$\Delta\tau_{\mathrm{prop}}$	20 ms	20 ms	20 ms
$\Delta\delta_{\mathrm{sat}}$	20 ms	20 ms	20 ms
$\Delta\delta_{\mathrm{rx}}$	0 s	0 s	30 s
$\mathrm{D}\tau_{\mathrm{max}}$	0 s	2.5 s	0 s
f_s	10 MHz	10 MHz	10 MHz
$N_{\mathrm{bits,snp}}$	0.9 MB	48.6 MB	573.3 MB

Table 4.3: Examples of E6-C snapshots storage needs.

Of course, this size could increase rapidly if the location of the RECS is randomized. For example, if the RECS Random Delay is configured to 2.5 seconds, the size occupied by the E6-C snapshot will be 48.6 MB. If the receiver does not have an accurate reference, the size could increase even more dramatically. If the receiver must search for the RECS throughout an entire I/NAV frame, this size could reach up to 573.3 MB.

It should be noted that the number of snapshots to be stored in the receiver will depend on the delay between the delivery of the TESLA key and its related RECS.

From the analysis conducted above, we can conclude that:

- The memory required to record the E6 snapshot can significantly increase if the receiver lacks an accurate time reference. However, typically only a single snapshot needs to be stored by the receiver, particularly in cases where the disclosed TESLA key corresponds to the same RECS period. As a result, the total memory required remains manageable by current standards.

- The memory required to store downloaded RECS files can also increase significantly if the receiver is expected to operate autonomously over an extended period, potentially reaching several gigabytes. Additionally, the cost of the data downloaded by the receiver must be taken into account. For example, in a typical scenario (corresponding to use case 2 of Table 4.2) where approximately 250 MB per day are needed, an SAS receiver would require roughly 7.5 GB of data per month.

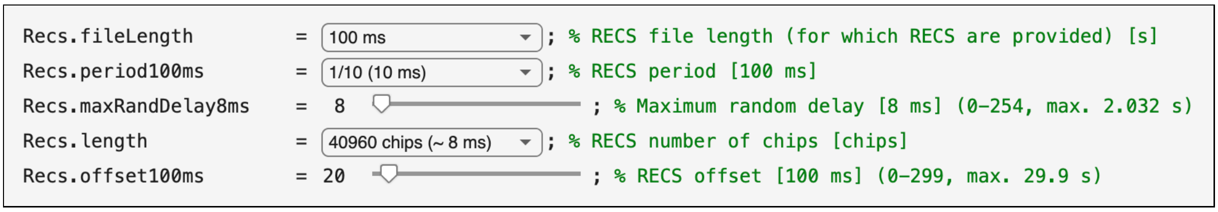
4.3 MATLAB Simulator

To assess the impact of the different parameters involved in SAS from a theoretical perspective, a tailor-made simulator has been developed. It focuses on simulating the acquisition procedure for the generic approach as described in Section 4.2. In this section, we provide a description on how the simulator works and the parameters that are considered.

The cryptographic operations involved in the encryption and decryption of the RECS are not considered here, as they do not affect the performance at signal-level. Therefore, in the simulator, the RECSs are already considered decrypted.

The simulator is implemented in MATLAB™ from MathWorks, a programming language widely used as the de facto standard in the academic and scientific community due to its excellent balance between the effort of coding and the results obtained.

Files are coded using the latest MATLAB live scripts/functions (.mlx), which, compared to legacy files (.m), offer enhanced input of parameters (such as defining predetermined ranges) and a built-in visual interface. An example is shown in Figure 4.4. The code follows the official MATLAB style guidelines [119].



```
Recs.fileLength      = 100 ms                ; % RECS file length (for which RECS are provided) [s]
Recs.period100ms     = 1/10 (10 ms)          ; % RECS period [100 ms]
Recs.maxRandDelay8ms = 8                     ; % Maximum random delay [8 ms] (0–254, max. 2.032 s)
Recs.length          = 40960 chips (~ 8 ms)  ; % RECS number of chips [chips]
Recs.offset100ms     = 20                    ; % RECS offset [100 ms] (0–299, max. 29.9 s)
```

Figure 4.4: Configuration of RECS parameters in the MATLAB simulator.

The main SAS simulator comprises four primary modules: the SIS generator, the RECS generator, the received signal generator, and the acquisition module. Simulation parameters are consolidated in a structure passed as input argument to the different modules and their included functions. Additionally, a Monte Carlo module is utilized on top of them to derive statistical results by iterating through the main simulator. The following sections describe these modules. Figure 4.5 illustrates a high-level block diagram of the MATLAB simulator and Table 4.4 summarizes their corresponding inputs and outputs (simulation parameters aside).

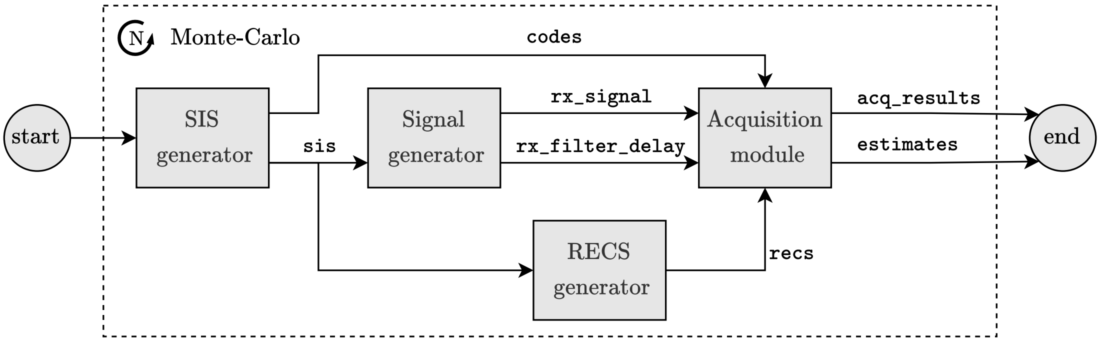


Figure 4.5: Block diagram of the MATLAB SAS simulator.

Module name	Inputs	Outputs
SIS generator	–	`Codes`, `Sis`
RECS generator	`Sis`	`Recs`
RX signal generator	`Sis`	`RxSignal`, `rxFilterDelay`
Acquisition module	`RxSignal`, `Codes`, `Recs`, `rxFilterDelay`	`AcqResults`, `SnrEstimates`

Table 4.4: MATLAB simulator modules

4.3.1 SIS Generator

It is responsible for generating the SISs to be transmitted by the satellites. It obtains the primary codes for the E6-C band required for SAS, according to the specifications published in the corresponding ICD [120]. The E6-C sequence is then generated by concatenating these primary codes for the desired simulation time. Simulated code delay is also applied at this stage. Finally, the E6-C chips can also be encrypted using a randomly generated key. Other bands can also be generated if required, as well as the corresponding secondary code sequences if applicable. A block diagram of the module is shown in Figure 4.6.

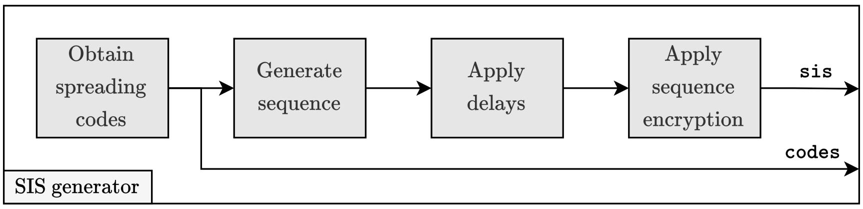


Figure 4.6: Block diagram of the SIS generator.

The input parameters that can be configured in this module are provided below.

- SVIDs to generate.
- Observation time to be simulated.
- Starting primary code chip per SV (to simulate the propagation delay).
- Starting secondary code chip per SV.

4.3.2 RECS Generator

This module generates the required RECS. Since encryption/decryption operations are not considered, this module directly obtains the ECSs (i.e., the decrypted RECSs). To achieve this, the module calculates the location of the ECSs based on the configured RECS parameters, including their length and periodicity. Once calculated, the module extracts the corresponding fragments from the E6-C SIS provided by the SIS generator. A block diagram of the module is shown in Figure 4.7.

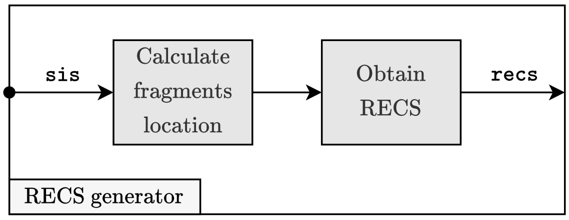


Figure 4.7: Block diagram of the RECS generator.

The input parameters that can be configured in this module are provided below.

- RECS File Length
- RECS Length
- RECS Period
- RECS Maximum Random Delay
- RECS Offset

4.3.3 RX Signal Generator

The signal is then modulated according to its specifications (BPSK/BOC) and filtered by the simulated channel coefficients (AWGN or LMS). Next, all the common impairments that affect the transmission from satellites to the receiver are added to the signal: the Doppler effect, the phase noise, the thermal noise (AWGN) and the interference (to account for the MAI effects produced by other satellites in view). Finally, the resulting received signal is filtered by the front-end RX filter, which is generated using a FIR low-pass model.

This module commands the generation of all necessary signals at the receiver. First, it samples the SIS chips provided by the SIS generator using fractional sampling. This method, which employs a noncommensurate sampling rate, prevents degradation of the code delay estimation [24], [70]. Subsequently, the signal is modulated according to its specifications (BPSK/BOC) and filtered by simulated channel coefficients (AWGN or LMS). Next, common impairments, including those that affect transmission from satellites to the receiver, are incorporated into the signal: the Doppler effect, phase noise, thermal noise (AWGN), and interference (to account for multiple-access effects produced by other satellites in view). Finally, the resulting received signal is filtered by the front-end RX filter, generated using a FIR low-pass model.

Special attention is devoted to phase noise (see Section 4.3.6), as it can significantly impact ACAS depending on the RECS periodicity. The simulator facilitates the simulation of various types of receiver clocks, encompassing the Temperature-Compensated Crystal Oscillator (TCXO) commonly found in typical handheld devices, as well as the more advanced Oven-Controlled Crystal Oscillator (OCXO) and Chip Scale Atomic Clock (CSAC) variations.

A block diagram of the RX signal generator module is shown in Figure 4.8.

The input parameters that can be configured in this module are provided below.

- SVIDs in view.
- Reference SVID.
- C/N_0 for reference satellite.
- C/N_0 difference with respect to the reference satellite (per SV).
- Doppler frequency (per SV).
- Satellite propagation time (per SV).

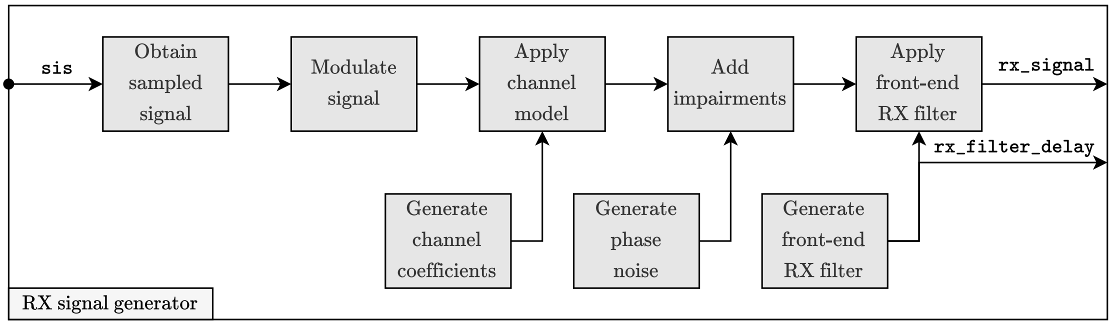


Figure 4.8: Block diagram of the RX signal generator.

- Satellite clock offset (per SV).
- Maximum uncertainty of the RX clock offset.
- RX clock type (ideal, TCXO, OCXO, CSAC).
- RX front-end low-pass filter bandwidth.
- Channel model (AWGN, LMS).
- Sampling frequency.
- Number of samples per chip (forced or frequency-dependent).

4.3.4 Acquisition module

This is the largest module of the simulator. It encompasses the acquisition procedure described in Section 4.2, which aims to detect the ECSs in the simulated received signal. The main acquisition sub-module consists mainly of correlating the received signal provided by the signal generator module with the decrypted RECS (i.e., the local replica) provided by the RECS generator module.

The bi-dimensional search space is configured according to the size of the Doppler frequency bin and the length of the acquisition window, as specified in (4.19). The correlation is performed using the efficient Parallel Code Phase Search (PCS) acquisition method, which translates the correlation in time to the frequency domain through the FFT, as shown in (4.23).

$$R(\tau, f_d) = \mathcal{F}^{-1}\{\mathcal{F}\{r[n]\}\mathcal{F}^{*}\{c[n]e^{j2\pi f_d n}\}\} \tag{4.23}$$

where the Fourier Transform $\mathcal{F}$ and its inverse $\mathcal{F}^{-1}$ are performed using the `fft` and `ifft` commands, respectively.

The overlap-save method is specifically employed, as detailed in [24], to correlate the received samples using blocks of samples whose length equals two primary code periods. This approach effectively mitigates the aliased replicas that typically arise when performing circular correlation, which result from the Inverse Fast Fourier Transform (IFFT), by accounting for the discrepancy between the number of samples used per code and the number of samples employed in performing the FFT.

As a result, a CAF is obtained, which contains the integration values for all cells evaluated. The CAF is then used to obtain the different results required to assess the acquisition performance, including the maximum peak value and the Doppler frequency and the code phase delay that reach this maximum. To compute the correct code delay, the module needs to know the delay of the simulated receiver front-end filter. In addition, to obtain a more precise estimation of the code phase delay, the correlation peak can be interpolated using a piecewise linear interpolator [24], [121]. The Primary Peak to Secondary Peak (PPSP) relationship is also calculated and compared to a predefined threshold to decide whether the signal is present or not.

An auxiliary acquisition sub-module is also included. It can be used to perform the acquisition of an auxiliary signal to assist the main acquisition module, providing, for example, estimates of the Doppler frequency.

The module also provides estimates of the output SNR and C/N_0 of the received signal. Finally, the acquisition results and the estimates obtained can be displayed for graphical comparison. A block diagram of the module is shown in Figure 4.9.

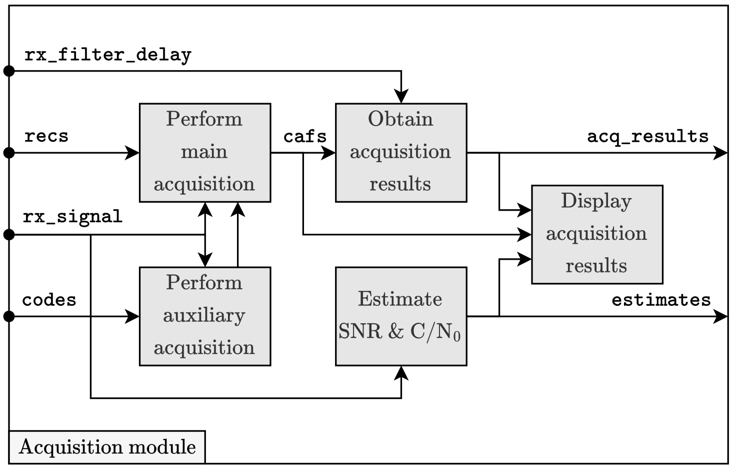


Figure 4.9: Block diagram of the acquisition module.

The input parameters that can be configured in this module are provided below.

- SVIDs to search.
- Number of coherent/non-coherent integrations.
- Doppler frequency search space (range, bin size).

The results of the acquisition are listed below.

- Code phase delay estimate.
- Doppler frequency estimate.
- Maximum peak value.
- PPSP.
- Output SNR estimate.
- C/N_0 estimate.

4.3.5 Monte-Carlo module

To obtain statistical performance, the simulator can execute the required number of Monte-Carlo iterations. At each iteration, the random seed is updated, leading to the generation of new instances of AWGN or phase noise. This module integrates various tools for graphical comparison, such as the useful ROC curves. As explained in Section 2.2.6, these curves allow one to compare the global probability of detection against the global probability of false alarm. This comparison can be conducted as a function of C/N_0, the RECS Length, the receiver clock type, or the maximum reception delay uncertainty.

The input parameters that can be configured in this module are provided below.

- Number of Monte-Carlo iterations.
- Parameter to iterate (C/N_0, RECS Length, $\Delta\tau_{\max}$, RX clock type).
- Range of value of the parameter to iterate.
- Configuration of the input parameters of the main simulator module.

In Figure 4.10, we present an example of ROC curves for single cell probabilities obtained using the Monte-Carlo module of the MATLAB simulator, averaging 5000 iterations for different C/N_0 values. Ideal conditions have been assumed (i.e., no Doppler frequency, no phase noise, no impairments). The acquisition metric used is the squared absolute value of the CAF output. As observed, the curves estimated by the module

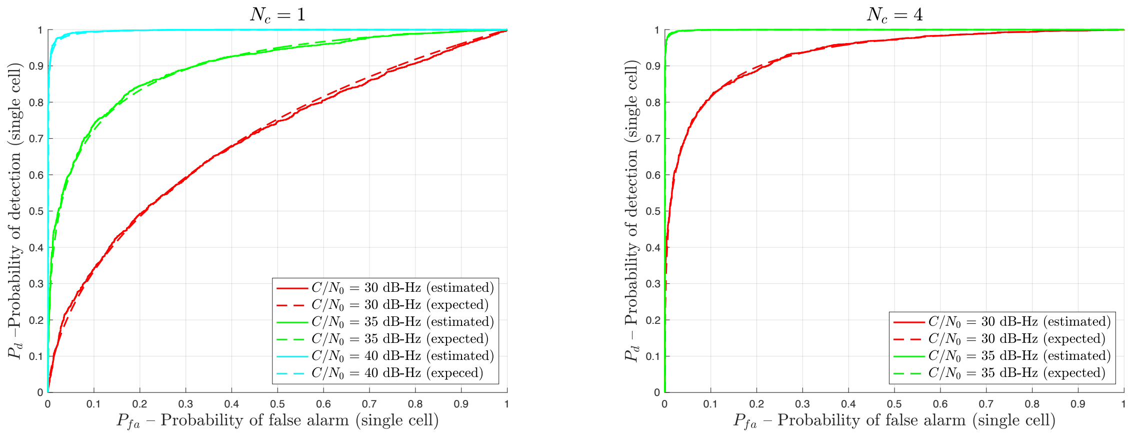


Figure 4.10: ROC curves for single cell evaluation using RECS of 5120 chips (∼ 1 ms) for different coherent integrations.

match the theoretically expected ones (see Section 2.2.5). This is done for when a single RECS is coherently integrated ($N_c = 1$) but also for when four RECSs are coherently combined ($N_c = 4$).

The module can also provide the estimated probability of detection in terms of the C/N_0 for a given target probability of global false alarm. In Figure 4.11, an example is shown for a typical $P_{\text{FA}} = 0.01$ value. As expected, the difference is approximately 3 dB.

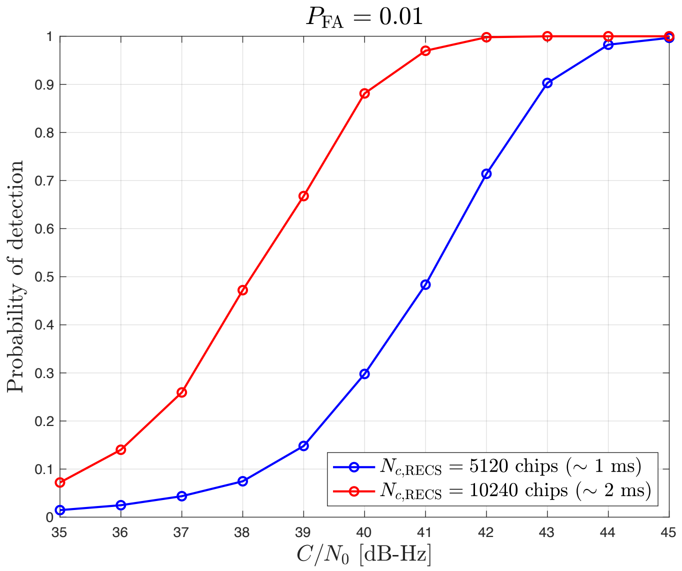


Figure 4.11: Probability of detection vs C/N_0 for RECSs of 5120 (∼ 1 ms) and 10240 chips (∼ 2 ms).

4.3.6 Phase noise generator

The phase noise generator is based on the two-state clock model described in Section 2.2.7. The equivalent discrete model implemented in the simulator is illustrated in Figure 4.12.

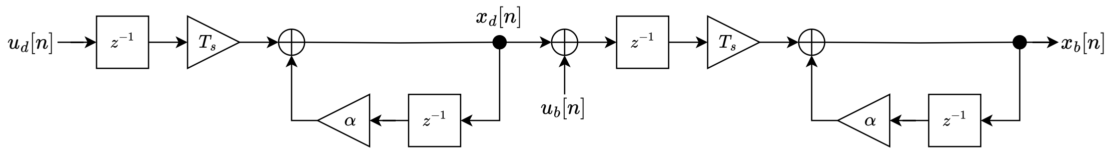


Figure 4.12: Discrete version of the two-state-clock model.

From the output of the model, the clock bias x_b, we obtain the phase noise, denoted ϕ, as in (4.24), which is added to the received signal generated by the simulator by multiplying it by the term $e^{j\phi_n}$.

$$\phi_n = 2\pi f_c x_b[n] \tag{4.24}$$

Table 4.5 show the values for the h-coefficients (see Section 2.2.7) used for simulation [38], [39].

Clock Type	h_0	h_{-2}
TCXO	$9.4 \cdot 10^{-20}$	$3.8 \cdot 10^{-21}$
OCXO1	$8.0 \cdot 10^{-20}$	$4.0 \cdot 10^{-23}$
OCXO2	$3.4 \cdot 10^{-22}$	$1.3 \cdot 10^{-24}$
OCXO3	$2.6 \cdot 10^{-22}$	$4.0 \cdot 10^{-26}$
OCXO4	$2.5 \cdot 10^{-26}$	$2.5 \cdot 10^{-22}$
CSAC	$7.2 \cdot 10^{-21}$	$2.7 \cdot 10^{-27}$

Table 4.5: Clock Type Coefficients

A comparison of the phase noise of different types of receiver clock models is shown in Figure 4.13. Two observation times, 1 second and 3600 seconds, have been simulated to evaluate the short-term and long-term stabilities, respectively. No phase wrapping has been applied.

As we can observe, the TCXO shows the worst short-term stability due to its larger h_0 value, while the OCXO4 type offers better short-term stability due to its lower h_0 value.

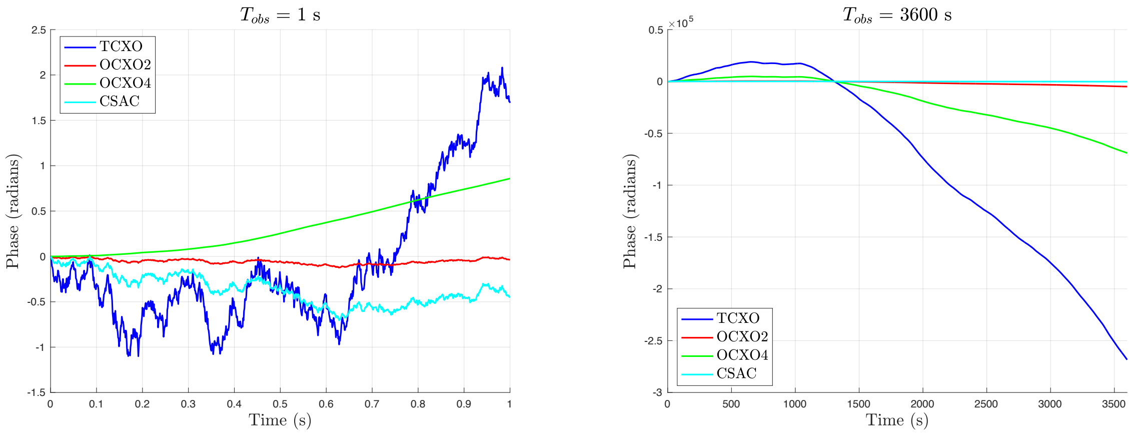


Figure 4.13: Comparison of phase noise for different types of receiver's clock models and for different observation times.

However, it is the CSAC type that offers the best long-term stability, thanks to its lower h_{-2} value, followed closely by the OCXO2 type.

4.4 Determining (Theoretical) Minimum Requirements

In this section, we evaluate the performance of the SAS acquisition in terms of the probability of detection (P_{D}) versus the probability of false alarm (P_{FA}) using ROC curves. For a given C/N_0 and assuming a single integration, this performance is mainly determined by the size of the acquisition search space [33], which depends on the number of Doppler cells (frequency dimension) and code delay cells (time dimension) evaluated.

The number of cells in the frequency dimension depends on several factors, such as the speed of the user vehicle. However, it does not depend on any specific parameter of SAS. Therefore, in this section, its effect is removed by assuming that the Doppler effect has been perfectly estimated, allowing the receiver to avoid any search in the frequency space.

The number of cells in the time dimension depends on the length of the acquisition window, which is specified in (4.19). This dependence arises from two parameters:

1. The length of the RECS/ECS (i.e., T_{RECS}).
2. The maximum reception delay span (i.e., $\Delta\tau_{\max}$).

In case of multiple integrations, the RECS periodicity can play a significant role and therefore must also be considered, as depicted in Section 4.2.5. The rest of the RECS parameters (the RECS Offset and RECS Maximum Random Delay) are not considered, as they do not affect the size of the search space and consequently the probability of detection.

In the next section, we assess the impact of the relevant parameters separately. In the following, the value of the RECS Period (τ_{RECS}) is specified only when multiple integrations are performed; otherwise, it is chosen long enough to perform the correlation during the length of the ECS. It should be noted that these parameters will still have an impact on the processing time needed to compute the PVT solution.

For this evaluation, a scenario with ideal simulation conditions is assumed to maximize the probability of detection. This including using an AWGN channel, an infinite receiver front-end filter bandwidth and a Doppler frequency perfectly estimated.

4.4.1 Impact of RECS Length

To focus on the impact of ECS length on the performance, we assume that only one integration is performed and that the maximum reception delay span is set to a practical minimum, which is considered the length of the sequence to be correlated, that is

$$\Delta\tau_{\text{max}} = T_{\text{RECS}} \tag{4.25}$$

Therefore, the length of the acquisition window, according to (4.19), is set to

$$T_{\text{acq}} = 2T_{\text{RECS}} \tag{4.26}$$

Different C/N_0 scenarios have been simulated, ranging from 30 dBHz, which typically corresponds to a dense urban scenario, to 40 dBHz, which typically corresponds to an open sky scenario. The results obtained in Figure 4.14, which averages 2000 Monte-Carlo realizations, are summarized in Table 4.6.

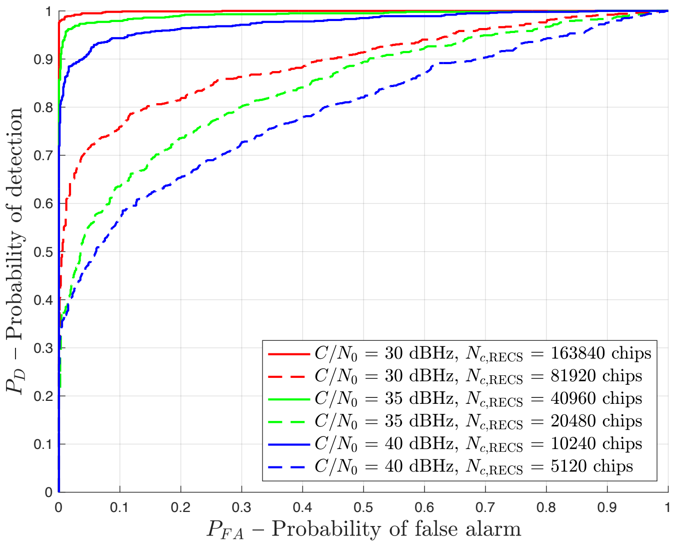


Figure 4.14: ROC curves showing the impact of the RECS/ECS length.

$\mathbf{C/N_0}$	**Minimum ECS length**
40 dBHz	10240 chips (∼ 2 ms)
35 dBHz	40960 chips (∼ 8 ms)
30 dBHz	163840 chips (∼ 32 ms)

Table 4.6: Minimum RECS/ECS length as a function of C/N_0.

4.4.2 Impact of reception delay uncertainty

As stated in (4.15)-(4.16), the maximum reception delay span ($\Delta\tau_{\text{max}}$) depends on three parameters: the receiver clock offset, the satellite clock offset, and the propagation delay. However, the maximum spans for the last two are bounded in practice, so it is the receiver clock offset span, which account as an uncertainty, that can have the largest impact on the acquisition performance.

In the simulations carried out, we consider $\Delta\delta t_{\text{sat,max}} = \Delta\delta t_{\text{prop,max}} = 10$ms, which is consistent with the Galileo specifications. The maximum receiver clock offset span ($\Delta\delta t_{\text{rx,max}}$) ranges from zero to several seconds, according to the receiver clock calibration assumptions made.

Therefore, the length of the acquisition window, according to (4.19), is set to $T_{\text{acq}} = 20\text{ ms} + \Delta\delta t_{\text{rx,max}} + T_{\text{RECS}}$. For the ECS/RECS lengths, the minimum required values are used for each of the C/N_0 scenarios considered in Section 4.4.1, which are provided in Table 4.6.

As shown in Figure 4.15(a), which averages 2000 Monte Carlo realizations, the degradation in terms of P_{D} is starting to become noticeable even with a maximum span of 1 second, especially for the configuration of 10240 chips and $C/N = 40$ dBHz, which is clearly the worst combination. In Figure 4.15(b), the maximum span is extended to 10 seconds for the worst combination stated previously. As expected, the probability of detection is severely degraded. Since the ECS can be located at any position within the acquisition window, the search space of the CAF is expanded, leading to an increased probability of false alarm, mainly due to the higher probability of missed detection.

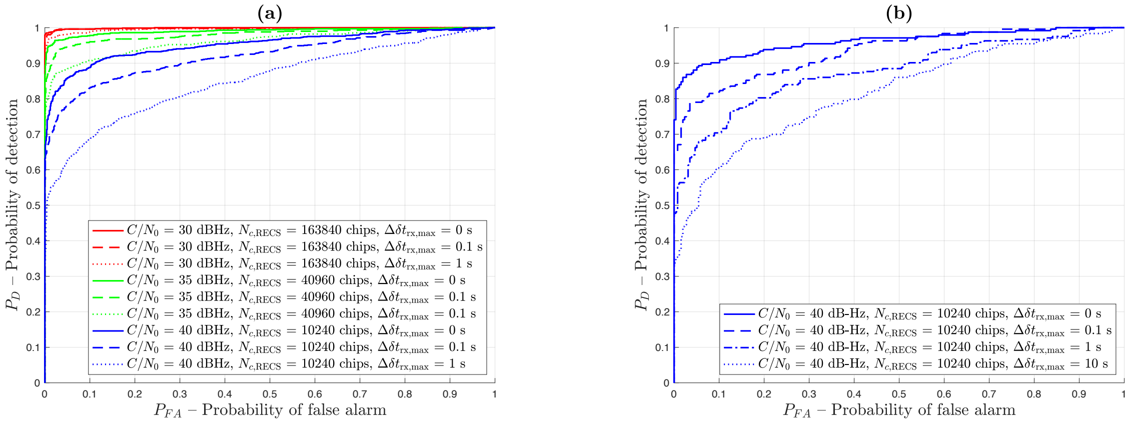


Figure 4.15: ROC curves showing the impact of the maximum reception delay uncertainty.

4.4.3 Impact of RECS Period

For low signal to noise ratios, it may be necessary to perform multiple combinations of the RECS sequences. As detailed in Section 4.2.5, the maximum coherent integration time that can be achieved will be mainly determined by the separation between the first and last sequences to be (coherently) combined. In the case of the phase noise, according to the aforementioned results, this turns in the order of some hundreds of ms, depending on the clock type used in the receiver.

In Figure 4.16, a simulation is performed to assess how this separation affects the probability of detection when using coherent integration. In this case, 500 Monte Carlo iterations are averages, using RECSs sequences of 10240 chips ($\sim$ 2 ms), a TCXO-type receiver clock, and $C/N_0 = 35$ dBHz. The simulation shows the impact of different RECS Periods when $N_c = 3$ coherent integrations are performed. Also, a reference using an ideal clock is provided, which consequently does not depend on the RECS Period simulated.

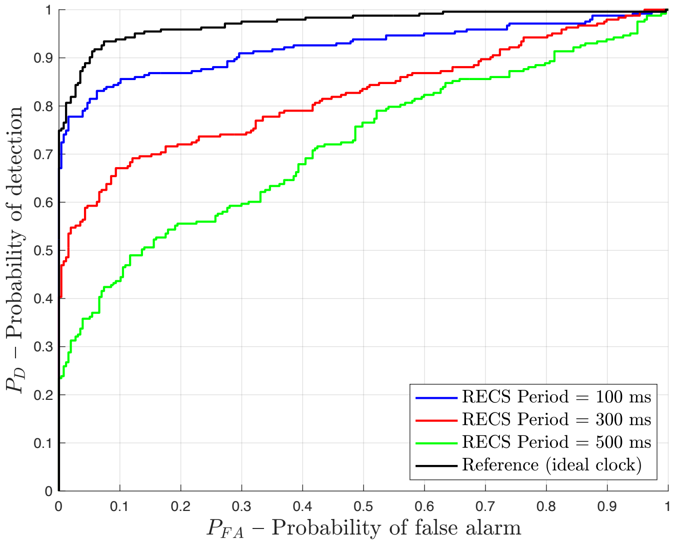


Figure 4.16: ROC curves showing the impact of the RECS Period.

As we can observe, the probability of detection is rapidly degraded when the RECS Period is increased when only coherent integrations are used, taking only into account the effect of phase noise. This degradation could be larger, of course, in the case of considering the Doppler frequency effect. In any case, the RECS Period severely limits the ability to perform coherent integrations.

4.5 PVT Authentication

The ultimate goal of SAS is to authenticate the PVT solution. The detection (or lack thereof) of the RECSs in the E6-C signal, according to the acquisition procedure presented in Section 4.2, is crucial but only the first step in verifying the authenticity of the received signal. In this section, we will discuss how a receiver could proceed to authenticate the PVT and outline the minimum conditions that need to be met.

4.5.1 Threats Identification and Mitigation

SAS is aimed at providing an additional layer of protection against spoofing attacks. Therefore, it is meaningful to identify the threat levels that a receiver might encounter and to deduce the assumptions that can be made for each case. The following threat levels are considered:

- Threat level 1 (T1): E6-C signal cannot be spoofed.
- Threat level 2 (T2): E6-C signal can be spoofed.

Clearly, if the receiver does not find the ECS where expected (signal is not detected), the authentication process fails no matter the threat level considered. However, if the receiver finds the ECS where expected (signal is detected), the assumptions that can be made will depend on the threat level considered. Under T1, the authentication will be considered successful; but, under T2, an additional check to discard the presence of the vestigial signal (the “true” one) is advisable. Only if this check is passed, the authentication can be considered successful.

Of course, even when the ECS is not found where expected, the search for the vestigial signal can be useful, for example, to determine the delay of a replay attack. Moreover, not detecting the signal at the first attempt, does not necessarily imply that the signal has been spoofed. Indeed, the receiver could be operating in a low C/N_0 scenario: in such case, the received E6-C signal could be too weak, and the receiver could try using a longer sequence (if available) to compensate this situation. This ‘E6-C weak signal’ assumption can be also confirmed or discarded by checking the E6-B component: if the signal level of the last one is suspiciously higher than the E6-C level, there are high chances that the E6-C could be spoofed.

Note that the receiver can implement T1 or T2 logic in different circumstances. For example, it may implement T2 logic at startup or with a certain periodicity, in background, while implementing only T1 logic for the regular authentication verifications. This will depend on the level of robustness sought by the manufacturer based on the intended application, and the receiver’s capabilities. It is also worth noting that, depending on its implementation, the search for the vestigial signal could imply recording quite long snapshots and, hence, using large amounts of memory in the receiver.

In Figure 4.17, we provide a flow diagram illustrating the receiver procedure for signal authentication using a generic approach. In the figure, VS stands for vestigial signal.

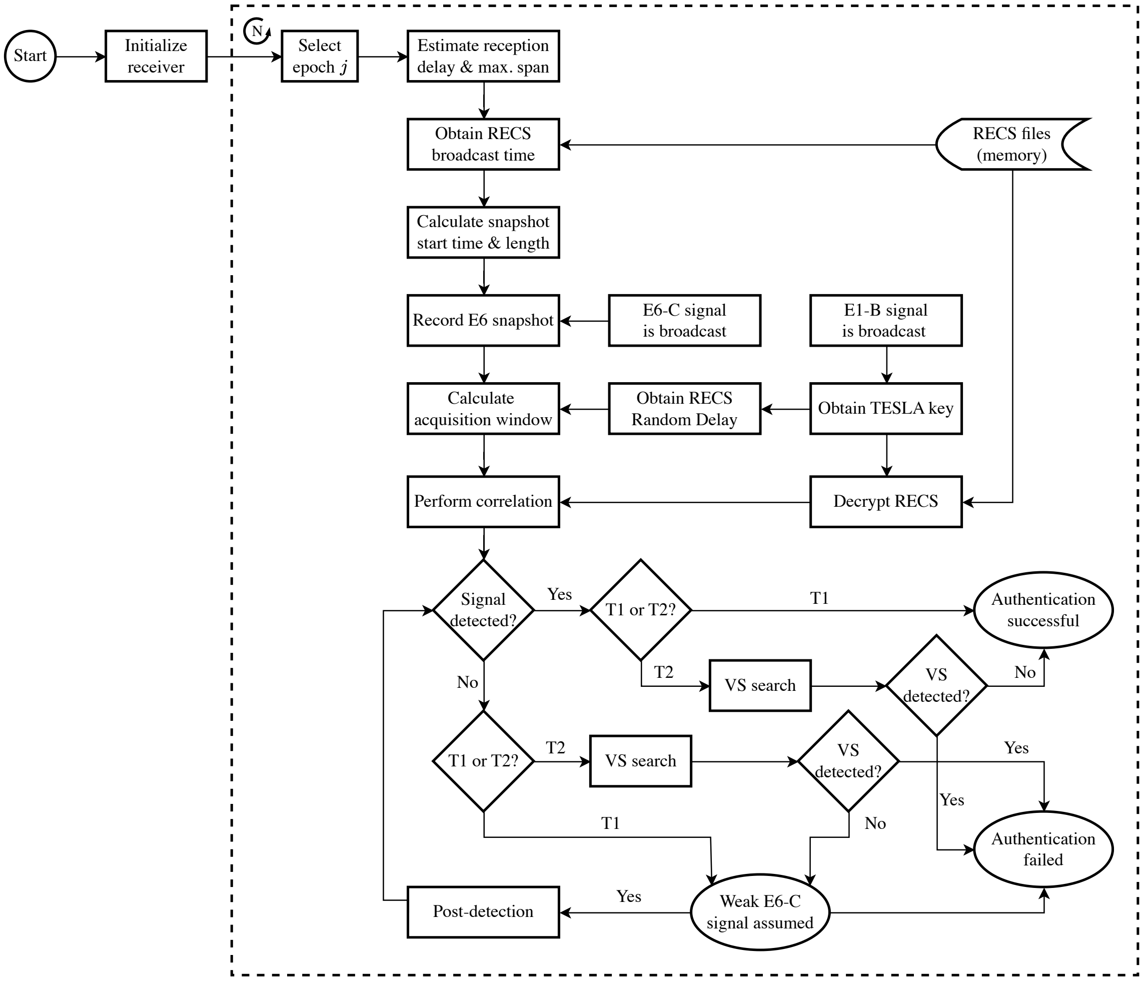


Figure 4.17: Flow diagram for SAS generic approach.

In Table 4.7 we summarize the possible outcomes and authentication assumptions that can be made depending on the threat level considered and the signals detected.

4.5.2 Vestigial Signal Search

In addition to the primary SAS algorithms implemented for signal detection and authentication, auxiliary algorithms can be employed to enhance resilience against spoofing attacks. These auxiliary methods may include techniques such as monitoring the AGC, analyzing the deformation of the correlation peak [122], assessing the receiver's dynamics through an IMU, or using the VSS.

The VSS algorithms are particularly important, as they are not only capable of de-

Threat level	**ECS found in expected location**	**Vestigial ECS found in another location**	**Authentication**	**Assumptions**
T1	Yes	N.A.	Successful	-
T1	No	N.A.	Failed	Weak E6-C signal
T2	Yes	Yes	Failed	E6-C spoofed [1]
T2	Yes	No	Successful	-
T2	No	Yes	Failed	E6-C spoofed
T2	No	No	Failed	Weak E6-C signal

[1] Multipath considerations may affect the assumptions made in this case.

Table 4.7: Authentication assumptions for the generic approach.

tecting spoofed signals but can also identify authentic signals under certain conditions, as described in [99]. In the event of a spoofing attack, both spoofed (inauthentic) and authentic signals might be present in the E6-C band. A VSS algorithm would be able to detect both signals and likely determine the authentic one, which is expected to be the earliest signal.

Therefore, it becomes evident that the VSS plays a crucial role in mitigating the possible threats the receiver may face. By default, it is assumed that the search for the vestigial signal is performed for the maximum reception delay uncertainty assumed by the receiver. However, in practice, depending on memory limitations and other factors, this search could be reduced, which will have implications on the assumptions that can be made.

Clearly, the VSS could be very time-consuming and may not be feasible for simple receivers with resource limitations. One possible approach to implement the VSS is to start by searching within a reduced window of samples, equivalent to assuming a smaller uncertainty for the receiver clock, and iteratively increasing this window by assuming larger uncertainties, up to the maximum reception delay uncertainty assumed.

This approach is shown in Figure 4.18, where we provide a flow diagram detailing the VSS.

It is important to note that the VSS requires working with larger E6 snapshots than those used for the acquisition of SAS. These snapshots must be pre-recorded by the receiver before initiating the search, and the available memory will constrain the VSS's ability to detect the vestigial signal.

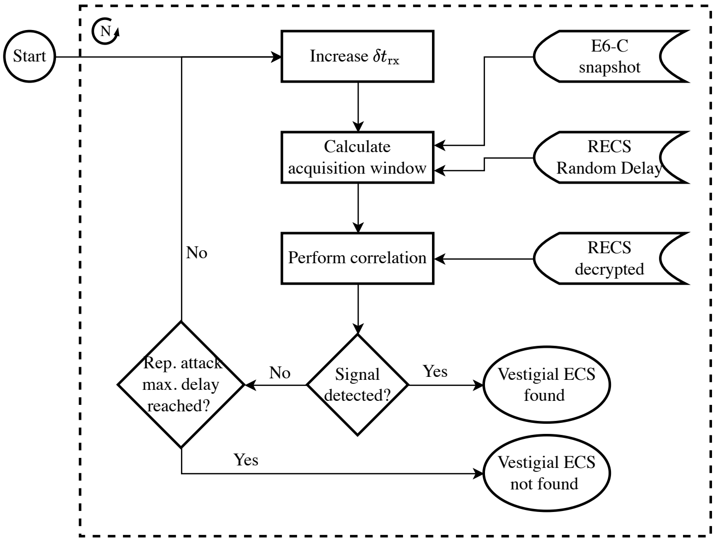


Figure 4.18: Flow diagram of the Vestigial Signal Search (VSS) procedure.

Moreover, although the vestigial search in E6-C can be executed as a non-priority background algorithm, it may still consume substantial receiver resources due to its exhaustive nature. Since RECSs typically last only a few milliseconds and are available during specific RECS periods (e.g., every 30 seconds), the probability of detecting them with an acceptable probability of false alarm diminishes significantly. Consequently, the vestigial search will be restricted by the resources available at any given time.

4.5.3 Authentication delay

One last aspect to consider when authenticating the solution through SAS is the time required to provide this authentication. Beyond the time taken by the receiver to process the signal, the authentication delay primarily depends on the TESLA key used to decrypt the associated RECS. Originally, this was defined by the RECS Key Delay parameter (see Section 3.4.1), which corresponds to the Key Delay Index (KDI) parameter in the latest published specification (see Section 3.5). Essentially, this parameter defines the number of I/NAV subframes between the subframe containing the OSNMA TESLA key used to generate the RECS and the subframe containing the RECS [110].

In the expected nominal case for a SAS receiver, the TESLA key from the next I/NAV subframe is used to decrypt the RECS of a given subframe (corresponding to KDI=1), which results in an authentication delay of over 30 seconds. To reduce this delay to just a few seconds, the receiver could use the TESLA key provided in the same I/NAV subframe containing the RECS (KDI=0), although this would require tighter synchronization. Finally, the specification also considers a "slow MAC" case (KDI=2), in which the receiver uses the key provided 11 I/NAV subframes later, leading to a delay of over 330 seconds. This last case is appropriate for loosely synchronized receivers.

4.6 Conclusions

This chapter has addressed the implementation of the SAS from a signal processing perspective, focusing on the acquisition process. A generic approach has been presented, which relies solely on the E6-C signal for detecting the RECSs. First, the receiver needs to determine the location of the related ECSs before the E6-C signal is broadcast. Once broadcast, the receiver will be able to record a snapshot of samples around this ECS. It has been shown that both the length of this sequence and the uncertainty assumed for the receiver clock offset are the main parameters to be considered at the acquisition level.

When an accurate time reference is available, the receiver will be able to precisely determine such location, and the acquisition search space will be significantly reduced, facilitating a high probability of detecting the corresponding RECS. In this case, the probability of detection will be primarily driven by the length of the RECSs and the C/N_0.

However, this probability of detection will be degraded if some uncertainty is assumed for the time reference. Indeed, the acquisition search space needs to be extended to account for this uncertainty, increasing the probability of false alarms and, consequently, reducing the probability of detecting the ECSs in the E6-C signal.

In addition, when dealing with low C/N_0 scenarios, multiple ECSs may need to be combined. In SAS, since the ECSs are separated by the RECS Period, the phase variations produced by the receiver clock phase noise or the Doppler frequency may negatively impact the coherent combinations. When the maximum coherent integration time is reached, non-coherent integrations may need to be considered. However, this requires combining a larger number of sequences, which may be unfeasible in SAS due to the time required to provide a PVT solution. In general, considering these limitations, post-detection techniques are not recommend in SAS, as they also make it more difficult to assign the corresponding reception time.

To analyse the impact of both the RECS Length and RECS Period in different signal-to-noise ratios scenarios, as well as the uncertainty assumed regarding the receiver clock, a MATLAB simulator has been implemented. By means of Monte Carlo iterations, it is able to determine which are the minimum requirements for such SAS parameters. This could be useful to select the configuration of a hardware's receiver and as a performance reference baseline for practical implementations.

The results show the minimum lengths required for the RECSs under ideal conditions for typical C/N_0 scenarios. A length of around 32 ms turns out to be enough for most scenarios where the SAS is intended to operate (C/N_0 not lower than 30 dB). This length may also avoid the need to use multiple integrations and their drawbacks. Indeed, as the results show, the length of the RECS Period is very limited for coherent integrations when considering a typical TCXO for the receiver.

Additionally, the impact of the receiver clock uncertainty is also analysed. On the basis of the results obtained, it becomes evident that having an accurate time source is crucial to ensure satisfactory acquisition performance. In SAS, where only a relatively small fragment of the received signal is used for correlation, having a bounded search space is highly desirable. Otherwise, the receiver's probability of detection is severely degraded. This uncertainty can be, however, reduced drastically if the satellite broadcast navigation is trustable, and even more, if a trustable time reference can be obtained from another source.

Finally, the mechanisms to authenticate the PVT, the ultimate goal of SAS, have been investigated. This authentication will depend on the assumed threats, specifically whether the E6-C signal could be spoofed. If the ECS is not found where expected, a vestigial signal search could help the receiver mitigate these threats. However, this search can be very resource-intensive, making it prohibitive for simple receivers.

Chapter 5

Nominal Operating Mode for SAS

5.1 Introduction

The acquisition implementation presented in Section 4.2 can be considered a generic procedure that relies on the reference time obtained from the RECS file and the receiver clock. The assumptions made about the calibration of this clock largely determine the maximum uncertainty in reception delay ($\Delta\tau_{\text{max}}$), as the uncertainties related to propagation delay and satellite clock offsets are typically bounded in practice. SAS If the receiver is equipped with a perfectly calibrated clock without any uncertainty (i.e., $\Delta\delta t_{\text{rx,max}} = 0$), the duration of the acquisition window would be only a few milliseconds, depending on the length of the ECS used. With these conditions, one could expect a performance similar to that of conventional GNSS signals (i.e., a concatenation of spreading codes).

However, in cases where the receiver clock is not calibrated and cannot rely on any accurate time reference, the acquisition window may need to be extended to several seconds or more. This would result in a significant degradation in performance in terms of probability of detection and false alarm. The simulations conducted in Section 4.4.2 illustrate this severe degradation.

Nevertheless, in SAS, where the receiver must track the E1-B signal to obtain the TESLA keys, it is possible to use the time reference obtained from the E1-B signal. This approach, that we name nominal operating mode, eliminates the dependence on the receiver clock and avoids the need to extend the acquisition window based on its calibration. Essentially, this approach reduces drastically the uncertainty on the reception delay, resulting in significantly improved acquisition performance in terms of probability of detection. However, it also exposes the receiver to potential malicious attacks on the E1-B signal, in addition to possible attacks on the E6-C signal.

The implications of using a time reference based on the E1-B signal are further examined in Section 5.4.2, where various approaches are considered.

5.2 Nominal Operating Mode

In this section, we analyze the nominal operating mode for SAS, as proposed in [1], using the generic approach presented in Chapter 4 as a baseline for the signal model. This mode is based on using the E1-B signal to enhance the acquisition of the ECSs of the E6-C band. Instead of directly searching for the ECSs in the E6-C signal, which would typically entail a large search space, the receiver leverages the estimates obtained from the E1-B signal to reduce this space in the E6-C band. This handover from E1 to E6-C is detailed in the following.

5.2.1 E1-handover

In the nominal operating mode, the receiver can estimate all the terms that affect the signal delay, including the propagation delay and the receiver and satellite clock offsets, from the E1-B signal solution. In this case, the reception delay that applies to the E6-C signal as defined in (4.3), is also affected by the estimated time bias between the E1 and E6 signals, which is denoted as $\delta^{k}_{\text{E1-E6}}$ for the k-th satellite. Therefore, the reception delay can be rewritten as

$$\tau^{k} = \tau^{k}_{\text{prop-E1}} - \delta t^{k}_{\text{sat-E1}} + \delta t_{\text{rx-E1}} + \delta^{k}_{\text{E1-E6}} \tag{5.1}$$

where $\tau^{k}_{\text{prop-E1}}$ is the estimated propagation delay, $\delta t^{k}_{\text{sat-E1}}$ is the estimated satellite clock offset (for the k-th satellite) and $\delta_{\text{rx-E1}}$ is the estimated receiver clock offset, all based on the solution provided by the E1-B signal.

Following the same line of reasoning as done in (4.4)–(4.10) for the generic approach in Section 4.2.1, we can obtain the reception delay as the sum of a minimum value plus its uncertainty, which in this case is given only by the estimated time bias between E1 and E6 bands, denoted $\Delta\delta^k_{\text{E1-E6}}$ for the k-th satellite:

$$\tau^k = \tau^k_{\text{min}} + \Delta\tau^k = \tau^k_{\text{min}} + \Delta\delta^k_{\text{E1-E6}} \tag{5.2}$$

where

$$\tau^k_{\text{min}} = \tau^k_{\text{prop,min}} - \delta t^k_{\text{sat,E1,max}} + \delta t_{\text{rx,E1}} + \delta^k_{\text{E1-E6,min}} \tag{5.3}$$

$$\tag{5.4}$$

Thanks to the time reference obtained from the E1-B, the receiver can be synchronized with the GST. Thus, for each RECS period, we can now associate a GST_p. Therefore, the snapshot start time at the receiver for the p-th period expressed by (4.13) for the generic approach detailed in Section 4.2.2 can be now rewritten as

$$t_{\text{snapshot-start},p} = \text{GST}_p + \delta_{\text{RECS}} + \tau_{\text{min}} \tag{5.5}$$

which account for all the satellites.

The length of the snapshot no longer depends on the maximum span of the reception delay, as in (4.17), since this parameter is no longer an uncertainty for the receiver, as the necessary information is extracted from the E1-B signal. Therefore, the length of the snapshot for the nominal approach can be expressed as

$$T_{\text{snp}} = T_{\text{RECS}} + \text{D}\tau_{\text{max}} \tag{5.6}$$

Finally, the length of the acquisition window is determined after extracting the RECS Maximum Random Delay from (5.6), which simplifies to

$$T_{\text{snp}} = T_{\text{RECS}} \tag{5.7}$$

As can be deduced from 5.7, in the nominal operating mode, a single correlation would suffice in an ideal case where no inter-frequency bias applies (i.e., $\delta_{\text{E1-E6,min}} = 0$). However, in practice, some additional correlations are necessary to account for this bias. In any case, the length of the acquisition window is further reduced compared to the generic approach, which relies solely on the E6-C signal. This allows for an increased probability of detecting the corresponding RECS, thereby enhancing the probability of successfully authenticating the signal, which is the ultimate goal of SAS [1].

5.2.2 E6-band Doppler Frequency Estimation from E1

For the correlation process, the receiver must also account for the Doppler shift search. However, in SAS mode, since the E1-B signal is also processed, the receiver can exploit the Doppler estimate from the E1 band to approximate the Doppler shift in the E6 band.

The E6-C Doppler frequency can be estimated from E1-B considering the carrier frequency ratios of both bands:

$$\hat{f}_{d,6} = \hat{f}_{d,1}\frac{f_{c,6}}{f_{c,1}} \tag{5.8}$$

where $\hat{f}_{d,i}$ and $f_{c,i}$ are, respectively, the Doppler frequency estimates and carrier frequencies for the Ei-th band ($i = \{1, 6\}$).

Ideally, the relationship between the Doppler shifts in both bands is a scale factor based on their respective carrier frequencies. However, when considering the effect of the ionosphere, which is inversely proportional to the carrier frequency, the frequency deviation in the E6-C band cannot be perfectly estimated using only the E1-B signal [3].

In any case, in the SAS nominal operating mode, the same frequency bin used for E1-B can be applied to E6-C, allowing the frequency search to be generally omitted. As a result, the acquisition search is effectively reduced to the time domain only, significantly simplifying the process.

5.2.3 Post-Detection

The contribution to the delay suffered by the signal is twofold: the effect of ionosphere and the Doppler effect. Thus, the total delay introduced in the signal for the Ei band, $i = \{1, 6\}$, expressed in cycles, is given by

$$\Delta\tau_i = \frac{40.3 \text{ TEC}}{cf_{c,i}} - \frac{f_{c,i}}{c}vt \tag{5.9}$$

where TEC is the Total Electron Content, $f_{c,i}$ is the carrier frequency for the Ei-band, c is the speed of light in empty space, v the radial velocity between the satellite and the user. The reference of time t is irrelevant because it does not affect the rate of change of $\Delta\tau_i$.

The estimate for the E6 signal from the E1 signal, denoted $\Delta\tau_6'$, expressed in cycles of $f_{c,6}$, will be given by:

$$\Delta\tau_6' = \Delta\tau_1 \frac{f_{c,6}}{f_{c,1}} = \frac{40.3 \text{ TEC}}{c f_{c1}^2} f_{c,6} - \frac{f_{c,6}}{c} vt \tag{5.10}$$

From the comparison of the previous expression and the estimate obtained directly from E6, we obtain that:

$$\Delta\tau_6 - \Delta\tau_6' = \frac{40.3 \text{ TEC}}{c f_{c,6}} \left(\frac{f_{c,1}^2 - f_{c,6}^2}{f_{c,1}^2} \right) = \frac{40.3 \text{ TEC}}{c f_{c,6}} I_6 \tag{5.11}$$

where $I_6 \approx 0.3412$ is the ratio which determines the residual error of the doppler frequency estimate for E6.

It is important to note that we are not interested in the absolute magnitude of the ionospheric error of (5.11), but in its variation over time, i.e., the slant ionospheric delay rate, since the first one can be considered within the acquisition search.

Next, in Table 5.1, we compute the maximum integration times for some typical values of the ionospheric delay rate using the worst-case scenario shown in [123], which have been obtained for the L1-E1 band; the slant ionospheric delay rate in E6 band has been obtained by multiplying the E1 band slant rate by the $f_{c,1}^2/f_{c,6}^2$ ratio. The maximum integration time is computed considering that the maximum acceptable delay, to sum coherently, is around one quarter of the wavelength, which for the E1 and E6 frequency band, corresponds to approximately 4.8 cm and 5.9 cm, respectively.

Slant iono. delay rate (E1)	**Confidence interval**	**Max. int. time**
0.8 cm/s	95 %	6 s
3.5 cm/s	99.9 %	1.4 s
10 cm/s	99.999 %	0.5 s
Slant iono. delay rate (E6)	**Confidence interval**	**Max. int. time**
1.2 cm/s	95 %	4.9 s
5.3 cm/s	99.9 %	1.1 s
15 cm/s	99.999 %	0.4 s

Table 5.1: Maximum integration time (E1 & E6 bands).

Finally, in Table 5.2, we compute the maximum integration times for the residual slant ionospheric delay rate in E6 band after doppler correction of E6 band from E1 band, which has been obtained by multiplying the slant ionospheric delay rate in E6 band by the correction ratio I_6 computed in (5.11).

Slant iono. delay rate (E6)	**Confidence interval**	**Max. int. time**
0.4 cm/s	95 %	14.2 s
1.8 cm/s	99.9 %	3.3 s
5.2 cm/s	99.999 %	1.1 s

Table 5.2: Maximum integration time (E6 band after correction).

Therefore, for large slant ionospheric rates, multiple coherent integrations among different RECS periods are only feasible if these periods are very small, e.g. around 300 ms if $N_c = 3$ integrations are envisaged. For slower rates, larger periods can be assumed, but still cannot exceed a few seconds.

It is worth noting that the maximum integration times obtained previously consider that no Doppler frequency search is performed; if it was done, these times could be increased. In any case, the correction in E6 based in E1 is still useful to reduce the uncertainty and, therefore, to reduce the number of frequency bins of the acquisition search space.

5.3 Determining (Practical) Minimum Requirements

In Section 4.4, the impact of the RECS Length and Period on acquisition performance, in terms of probability of detection, is analyzed for typical scenarios where SAS is intended to operate (C/N_0 not lower than 30 dB). The results presented are based on ideal conditions (i.e., infinite receiver bandwidth, perfectly estimated Doppler frequency, perfectly stable receiver, etc.) and within the generic framework (without leveraging any aid from the E1-B signal). In this section, we extend the analysis to more realistic scenarios, considering the impact of the RECS Length, and evaluate the case where the E1-B signal is used as a time reference, which is expected to be the default operating mode for SAS.

5.3.1 Impact of RECS Length

As expected, the length of the sequence to be correlated, which is determined by the number of chips used for the RECS/ECS, is one of the main SAS parameters that drive this performance at acquisition level. In Section 4.4.1, the minimum recommended lengths for these sequences are determined as a function of the carrier to noise density ratio, for a generic approach (non E1-B-aided).

However, as discussed in Section 5.2, utilizing the E1-B signal can significantly reduce the effective uncertainty to just a few samples. Figure 5.1 illustrates the impact of the RECS length in such a case, assuming an uncertainty of 20 samples, meaning that the acquisition window is approximately the length of a single RECS/ECS. For these simulations, ideal conditions have been assumed to evaluate the minimum required RECS lengths, which are summarized in Table 5.3.

The results presented are averaged over 2000 Monte Carlo realizations. The RECS Period is chosen long enough to perform the correlation during the length of the ECS.

$\mathbf{C/N_0}$	**Minimum ECS length**
40 dBHz	10240 chips ($\sim$ 2 ms)
35 dBHz	20480/40960 chips ($\sim$ 4/8 ms)
30 dBHz	81920 chips ($\sim$ 16 ms)

Table 5.3: Minimum ECS length as a function of C/N_0.

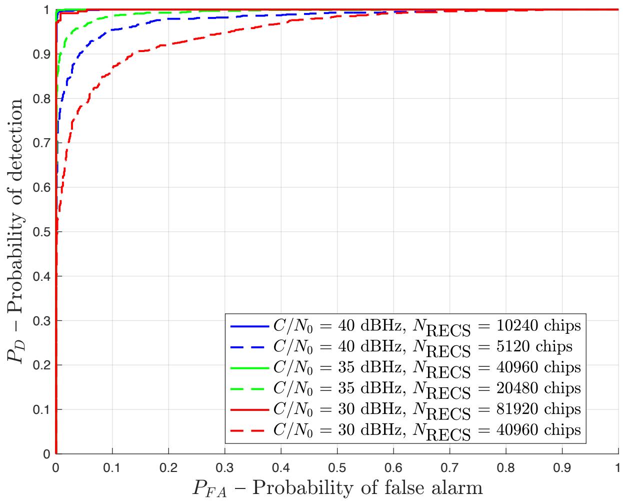


Figure 5.1: ROC curves showing the impact of of the RECS/ECS length, considering that the E6 parameters have been estimated from the E1-B signal to reduce the effective uncertainty.

In Figure 5.2, we compare these results with a more realistic scenario to assess the degradations relative to the ideal scenario. The realistic scenario simulates a TCXO-type receiver clock, a two-sided bandwidth receiver set to half the sampling frequency (with a sampling rate of 20 Msps), multiple satellite interference (7 satellites in view with equal power levels), and a Land-Mobile Satellite (LMS) channel (at 50 km/h), instead of the AWGN model used under ideal conditions. As in the previous simulation, we assume that the E6 parameters have been estimated from the E1-B signal. As expected, compared to the ideal scenario, larger RECSs may be required to compensate for the losses at a given C/N_0 to achieve a satisfactory probability of detection.

5.3.2 Impact of RECS Period

For low signal-to-noise ratios, it may be necessary to combine multiple RECS sequences. As detailed in Section 4.2.5, the maximum coherent integration time achievable is primarily determined by the separation between the first and last sequences that are coherently combined.

Regarding phase noise, the aforementioned results indicate that the integration time can reach several hundred milliseconds, depending on the type of clock used in the receiver. In Figure 5.3, a simulation is performed to evaluate how this separation affects the probability of detection when using coherent integration. In this case, ideal conditions

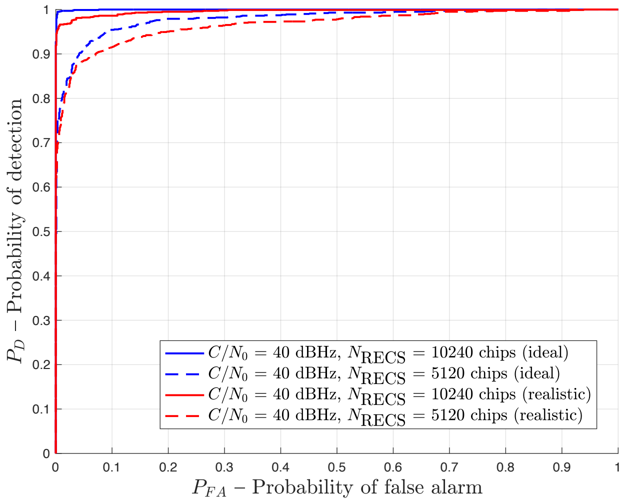


Figure 5.2: ROC curves showing the comparison of ideal (blue lines) and realistic scenarios (red lines).

are assumed to isolate the impact of the RECS Period. Additionally, a reference scenario using non-coherent integration is provided, which does not depend on the RECS Period being simulated.

The results presented are averaged over 500 Monte Carlo realizations for RECSs sequences of 10.240 chips ($\sim$ 2 ms), with a C/N_0 of 35 dBHz, and assuming a TCXO-type receiver clock.

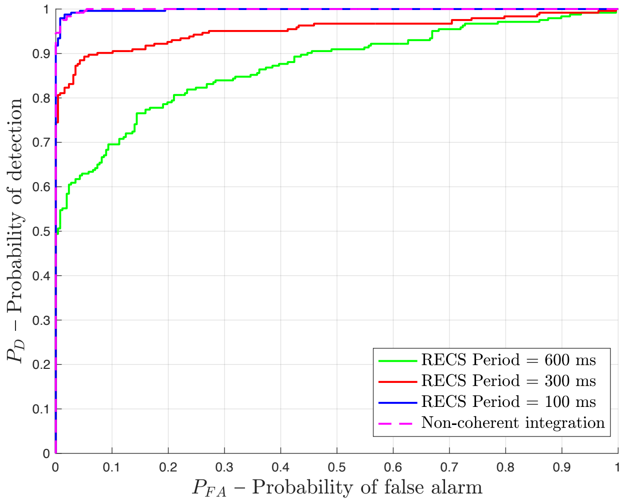


Figure 5.3: ROC curves showing the impact of different RECS Periods when $N_c = 3$ coherent integrations are performed ($N_i = 3$ non-coherent integrations shown as reference).

As observed, the probability of detection rapidly degrades as the RECS Period increases when only coherent integrations are used, considering solely the effect of phase noise. This degradation could be even more pronounced if the Doppler frequency effect is considered. In any case, the RECS Period significantly restricts the ability to perform coherent integrations.

5.4 PVT Authentication

5.4.1 Authentication Mechanism

The authentication mechanism proposed for the nominal operating mode of SAS relies on using one trusted ranging signal as an anchor for another ranging signal transmitted by the same satellite, either in the same or a different frequency band. The underlying principle is that an unauthenticated measurement may be considered reliable and usable for PVT if the difference between the measurements of the anchor and the other signal falls below a specific threshold, as outlined in [124]. Mixing authenticated and non-authenticated signals to verify the authenticity of the PVT solution is also investigated in [125].

The measurements involved in this process are the code phase delay estimations obtained from both E1-B and E6-C samples, meaning their accuracy directly influences the effectiveness of the authentication mechanism. Specifically, the measurement for the k-th satellite is authenticated only if the difference between the code phase delay observed on E6-C (τ^k_{E6C}) and the code phase delay estimated for E6-C based on E1-B ($\hat{\tau}^k_{E6C-E1B}$) is less than a predefined threshold, denoted as γ_{auth} [1]:

$$|\tau^k_{E6C} - \hat{\tau}^k_{E6C-E1B}| = |\tau^k_{E1B} - \hat{\tau}^k_{E6C-E1B} + \delta^k_{E1-E6}| \leq \gamma_{\text{auth}} \tag{5.12}$$

This threshold depends on the characterization of the error contributions in the estimation process. Quantifying the natural errors that contribute to the range difference expressed in (5.12), including all inter-frequency biases between E1 and E6 ($\hat{\tau}^k_{E6C-E1B}$), is crucial for precise modeling of these contributions. Accurate modeling enables the establishment of a consistent mechanism for authenticating the measurements. A comparison of the code phase delays estimated from E1-B and E6-B is presented in Section 6.4.1.

Finally, the position derived from E1-B can be authenticated if the condition expressed in (5.12) holds true for all the K satellites processed.

It is worth mentioning that other authentication mechanisms are possible. In [126], an authenticated timing protocol is proposed, which relies solely on SAS and authenticated navigation messages. This contrasts with the method mentioned earlier, which checks the consistency between E6 and E1 measurements, as the E1 signal is not authenticated at the ranging level.

5.4.2 Time Reference Source and Operating Modes

Due to the characteristics of SAS, multiple operating modes can be envisaged, depending on how the receiver obtains the time reference to record the snapshot of E6-C samples and performs the a-posteriori correlation with the decrypted RECS.

In the generic approach presented in Section 4.2, the receiver relies solely on its own clock to obtain a time reference. In the nominal operating mode, however, the E1-B signal can be used to derive a time reference that is independent of the receiver's clock uncertainty.

One possible approach is to use the PVT solution derived from the E1-B signal. In this case, since the time reference comes from the E1-B PVT, the only uncertainties the receiver needs to account for are the propagation delay and the satellite clock offset. The model presented in [1] follows this approach, which is referred to as the PVT-based approach.

An alternative method, also based on the E1-B signal, is to use the transmitted GST contained within the E1-B samples. In this so-called signal-based approach, the time reference is directly obtained from the samples, and the only uncertainty to account for is the satellite clock offset. Notably, in this approach, the receiver operates on a satellite-by-satellite basis, working at the signal level without the need to compute a position.

Indeed, each approach corresponds to different hardware setups and is designed to authenticate distinct outputs. In the generic approach, the receiver solely relies on the TESLA keys obtained from the E1-B signal. Therefore, if the ECS is found in the expected location, the receiver clock time could be authenticated. Afterward, the receiver can compute the E6-C pseudoranges and the E6-C PVT.

In the PVT-based approach, the receiver typically has access only to the E1-B observables or PVT solution. If the ECS is found in the expected location, the time derived from the E1-B PVT can be authenticated. To further authenticate the PVT, the receiver would then need to compute the E6-C pseudoranges. These pseudoranges can be compared with the E1-B ones or used to compute the E6-C PVT, which can subsequently be compared to the E1-B PVT.

In the signal-based approach, the receiver has direct access to the E1-B signal samples. If the ECS is found where expected, the transmitted GST can be authenticated, indicating that the E1-B signal has not been delayed by a spoofer. Consequently, the E1-B pseudorange can be trusted. When this check is applied to multiple satellites, the E1-B PVT can be authenticated directly, without the need to compute the E6-C PVT, as is necessary in the PVT-based approach.

In Table 5.4, we summarize the characteristics of the presented approaches, where $\delta_{\mathrm{E1,E6}}$ represents the time bias between the E1 and E6 pseudoranges. This bias includes the estimation of satellite bias, the offset due to ionospheric effects, and the receiver hardware bias, as detailed in [1].

	Generic	**PVT-based**	**Signal-based**
Time reference source	Receiver clock	Time derived from E1-B PVT	Transmit GST from E1-B
RX clock delay (δt_{rx})	Uncertainty	Not needed	Not needed
Propagation delay (τ_{prop})	Uncertainty	To be computed	Not needed
Sat. clock delay (δt_{sat})	Uncertainty	To be computed	To be computed (E1-E6 difference)
Interfrequency bias ($\delta_{\mathrm{E1,E6}}$)	Not needed	Uncertainty	Uncertainty

Table 5.4: Operating modes approaches

5.4.3 Threats Identification and Mitigation

Certainly, using a time reference based on the E1-B signal (either the PVT-based or the signal-based approach) eliminates the receiver's dependence on the reliability of its own clock and allows for a reduced acquisition window, which increases the probability of detection. However, this approach can also make the receiver more vulnerable to certain threats, such as spoofing attacks. In fact, the E1-B signal is more susceptible to spoofing than the E6-C signal, primarily due to the lack of encryption in its spreading codes.

Therefore, it is important to revise the definition of threats outlined for the generic approach in Section 4.5.1. For a receiver operating in the SAS nominal mode, the following threat levels are considered:

- Threat level 1 (T1): E1-B signal can be spoofed; E6-C signal cannot be spoofed.
- Threat level 2 (T2): E1-B and E6-C signals can be spoofed.

In Figure 5.4, we present a schematic flow diagram illustrating the receiver's procedure for signal authentication using the nominal operating mode, updated from the diagram for the generic approach in Figure 4.17. The additions and modifications relative to the generic approach are highlighted in grey boxes.

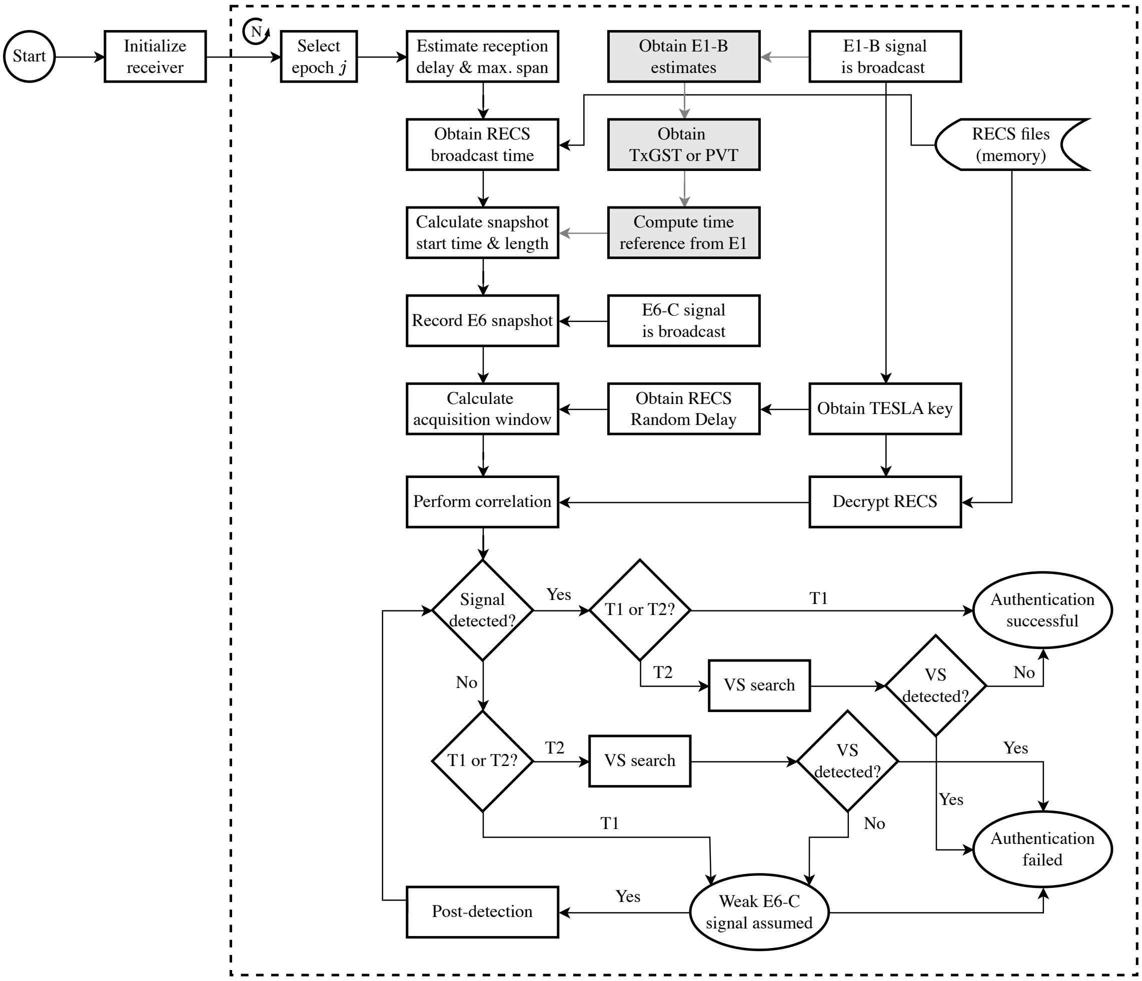


Figure 5.4: Flow diagram for SAS nominal operating mode.

The outcomes and authentication assumptions that derive from the threat levels considered above, summarized in Table 4.7 for the generic approach, are correspondingly

Threat level	ECS found in expected location	Vestigial ECS found in another location [1]	Authentication	Assumptions
T1	Yes	N.A.	Successful	-
T1	No	N.A.	Failed	E1-B spoofed, or weak E6-C signal
T2	Yes	Yes	Failed	E1-B and E6-C spoofed [1]
T2	Yes	No	Successful	-
T2	No	Yes	Failed	E1-B spoofed, or E1-B and E6-C spoofed
T2	No	No	Failed	E1-B spoofed, or weak E6-C signal

[1] Multipath considerations may affect the assumptions made in this case.

Table 5.5: Authentication assumptions for the nominal operating mode.

updated for the nominal operating mode in Table 5.5.

5.4.4 Vestigial Signal Search

Compared to the generic approach, the receiver can also rely on the E1-B signal for the VSS when required. Instead of performing an exhaustive search in the E6-C signal, the receiver can perform a handover from E1-B, narrowing the search to specific locations in the E6-C, as illustrated in [99] and [127]. This process is schematized in Figure 5.5.

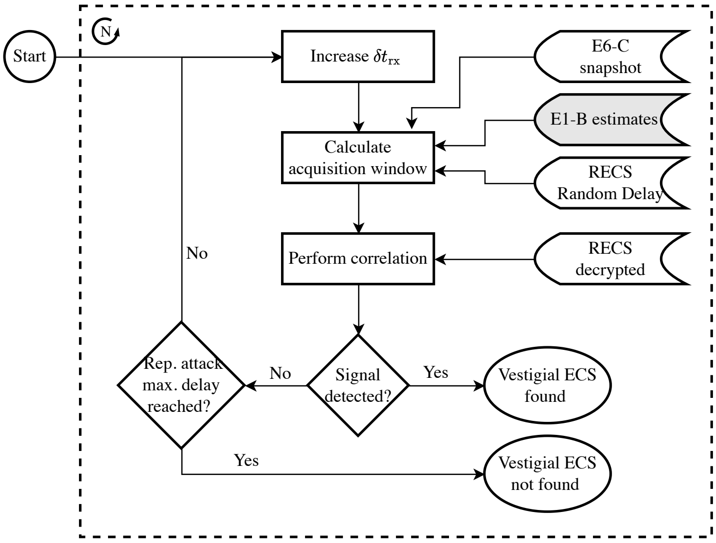


Figure 5.5: Flow diagram of the Vestigial Signal Search (VSS) procedure for the nominal operation mode.

Indeed, in the nominal operating mode, the spoofer may need to spoof both the E1-B and E6-C components to succeed, which enables the VSS algorithm to exploit both components. Consequently, a less resource-intensive VSS process could involve initially searching for both authentic and inauthentic signals in the E1-B component, leveraging its periodicity. After calculating the code phase biases, these signals could then be verified in the E6-C component [99].

However, in the event of a spoofing attack where the receiver cannot detect the second peak (i.e., the earlier peak corresponding to the vestigial/authentic signal) in E1-B, it would be impossible to hand over from the E1-B signal to the E6-C signal. In such cases, an exhaustive search in the E6-C band would need to be performed. If the receiver's time reference error is bounded to less than 30 seconds (i.e., within a single I/NAV frame), the VSS search could be constrained to this range.

Furthermore, the receiver must search for not just a single RECS but also for RECS from other periods in cases where the spoofer induces a delay greater than 30 seconds (assuming one RECS per I/NAV subframe).

It is also important to note that nothing prevents an SAS receiver from performing similar checks across multiple signals. Since a typical SAS receiver processes both E1 and E6 frequencies, the VSS process could be extended to other components, such as E1-C and E6-B. The latter would provide a more accurate handover to E6-C, as it does not introduce inter-frequency biases like the handover from E1-B.

5.5 Enhancing SAS Using E6-B Aiding

5.5.1 Detecting RECS Using Handover from E6-B

The non-repeating nature of E6-C has important implications for the receiver. Without further assistance, the acquisition search space could be prohibitively large. This is why, in SAS, the use of an auxiliary signal is assumed, which helps to reduce this search space. The natural choice for this auxiliary signal is the E1-B component, as it is the signal that provides the TESLA keys required for the RECS decryption.

In an ideal scenario, transitioning from the E1-B to the E6-C would provide to the receiver the exact location of the $\mathrm{ECS}^{\mathrm{R}}$ in the recorded snapshot. However, in practice, due to the ionospheric and multipath effects, in addition to the potential errors in the previous estimates, the receiver must account for an uncertainty.

Using the E6-B component as the auxiliary signal further reduces the uncertainty associated with E6-C measurements. This does not require additional hardware within the receiver, as E6-B shares the same frequency as E6-C. The main advantage of using E6-B over E1-B is that a handover from E6-B does not involve any inter-frequency biases, leading to more precise estimates, particularly in relation to the ionosphere contributions. This is illustrated in Figure 5.6.

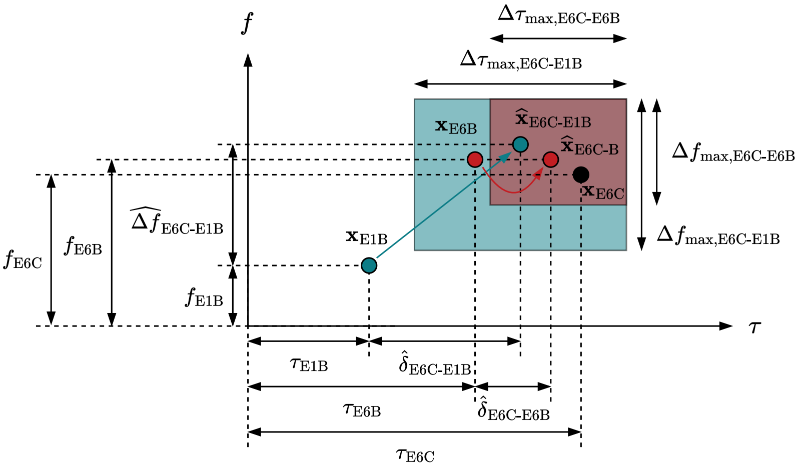

f
$\Delta\tau_{\text{max,E6C-E6B}}$
$\Delta\tau_{\text{max,E6C-E1B}}$
$\mathbf{x}_{\text{E6B}}$
$\hat{\mathbf{x}}_{\text{E6C-E1B}}$
$\hat{\mathbf{x}}_{\text{E6C-B}}$
$\mathbf{x}_{\text{E6C}}$
$\Delta f_{\text{max,E6C-E6B}}$
$\Delta f_{\text{max,E6C-E1B}}$
$\widehat{\Delta f}_{\text{E6C-E1B}}$
f_{E6C}
f_{E6B}
$\mathbf{x}_{\text{E1B}}$
f_{E1B}
τ
τ_{E1B}
$\hat{\delta}_{\text{E6C-E1B}}$
τ_{E6B}
$\hat{\delta}_{\text{E6C-E6B}}$
τ_{E6C}

Figure 5.6: Estimating E6-C code phase and Doppler frequency from E1-B and E6-B estimates.

It should be noted that the figure provided is not to scale. In the figure, $\mathbf{x}_i = (\tau_i, f_i)$ denotes the pair consisting of the code phase delay and Doppler frequency bins used to for the acquisition correlation. In practice, the search in the frequency dimension could be avoided.

Additionally, the estimation of the E6-C code phase delay from E6-B could further enhance the authentication mechanism described in Section 5.4.1. Since the threshold γ_{auth} is dependent on the characterization of the error contributions in the estimation process, incorporating an estimate of E6-C from E6-B could reduce this threshold ($\gamma'_{\text{auth}} \leq \gamma_{\text{auth}}$), thereby improving the overall authentication process. A comparison of the code phase delays estimated from E1-B, E6-C, and E6-B is provided in Section 6.4.1.

5.5.2 Snapshot Positioning using E6-B

The main goal of SAS is to authenticate the PVT, but the specific method of obtaining this PVT will vary depending on the receiver's implementation. In the nominal operating mode of SAS, the E1-B signal can be used to obtain the PVT, which could subsequently be authenticated with E6-C if the corresponding ECS is found where expected in the recorded snapshot. This approach can yield high accuracy in position determination since the receiver tracks the E1-B signal. However, this method implies using dual-frequency receivers that process both E1-B and E6-C signals, taking into account the inter-frequency biases between these bands for accurate estimation of the code phase delay and Doppler frequency, as discussed in Section 2.

Single-frequency SAS receivers can also be considered, where the E6-B signal is used instead of E1-B. In this case, it is assumed that the TESLA keys required to decrypt the RECS are provided through alternative means, such as a remote server connected via the internet, rather than processing the E1-B signal (see Figure 5.7). Under this setup, the receiver can rely on the E6 snapshot it records to compute the PVT, as this snapshot contains both E6-B and E6-C components.

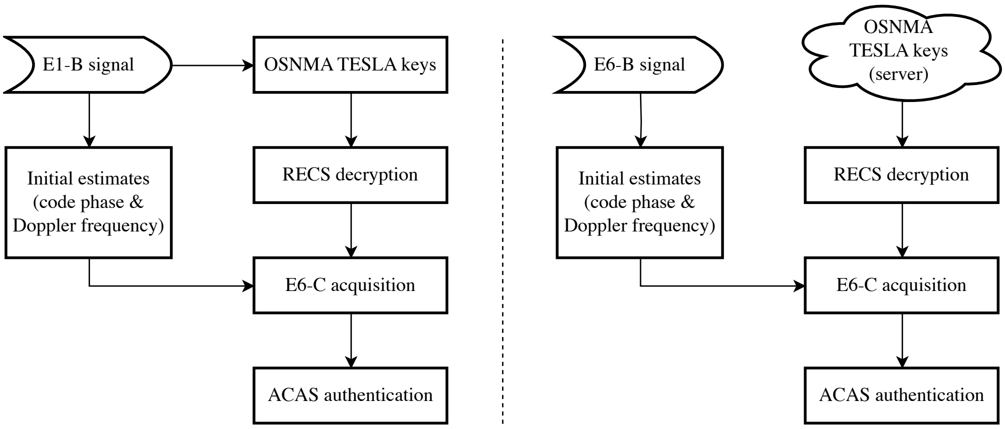


Figure 5.7: The TESLA keys could be retrieved from E1-B signal (left side) or from a remote server (right side).

Snapshot positioning involves certain subtleties compared to the conventional "acquisition and tracking" approach, such as peak interpolation and block-wise implementation. Interested readers are referred to Chapter 9 of [24] for more details. The accuracy of the PVT obtained from the snapshot will largely depend on the size of the snapshot and the carrier bandwidth, which in the case of E6-B is relatively large —approximately 10 MHz two-sided bandwidth for the main lobe.

Snapshot size

The size of the E6 snapshot is a crucial factor for the accuracy of snapshot-based positioning. In SAS, this size depends primarily on two parameters: the RECS configuration and the accuracy of the time reference available at the receiver. According to the latest SAS specification published [110], the RECS parameters that influence the snapshot size include the number of satellites to be tracked and the selected randomization parameters. Since RECS are provided on a per-satellite basis, a higher number of satellites will increase the required snapshot size to account for variations in propagation delays and satellite clock offsets. For Galileo, the span of propagation delays is approximately 20 milliseconds.

Regarding randomization parameters, the ECSs may either be transmitted at the specified broadcast time or delayed by a few milliseconds, depending on the selected values. Considering these factors, snapshot sizes between approximately 25 and 150 milliseconds are anticipated. Table 5.6 provides examples of snapshot sizes for different SAS configurations. The definitions of the parameters RAND and KDI are available in the latest specification [110] (see Section 3.5).

RECS length	RAND	KDI	Snapshot length
4 ms	0	0	$\sim$25 ms
4 ms	0	2	$\sim$50 ms
16 ms	1	0	$\sim$100 ms
16 ms	1	2	$\sim$150 ms

Table 5.6: Examples of snapshots sizes for different SAS configurations.

Comparison of E1-B tracking with E6-B snapshot acquisition accuracy

Depending on the receiver processing, the accuracy that can be expected using only the limited samples available in the E6-B snapshot may be lower than that achieved through continuous tracking on the E1-B signal. However, from an SAS perspective, the accuracy obtained through snapshot acquisition from E6-B may be sufficient for a wide range of applications, where having an authenticated solution is more critical than achieving highly precise PVT.

Next, we provide a preliminary analysis of the expected accuracies for both approaches. For continuous tracking of the E1-B signal, we refer to the simplified model expressed in (5.13). The selected parameters are typical for a conventional commercial receiver: 0.1 chips for early-late spacing, C/N_0 of 40 dB-Hz, and a DLL bandwidth of 1 Hz (refer to Chapter 4 of [24]). The K factor represents the sharpness of the correlation peak relative to a BPSK signal. For BOC (1,1)-based processing, as used for E1-B, this factor equals 3. The chip period, denoted by T_c, is approximately 1 µs for E1-B [114]. The speed of light is represented by c.

$$\sigma_{\mathrm{DLL}} \approx cT_c\sqrt{\frac{B_{\mathrm{DLL}}d_{\mathrm{E-L}}}{2KC/N_0}} \approx 0.5[\ \mathrm{m}] \tag{5.13}$$

A more refined model is discussed in Chapter 5 of [19], which provides a general expression for the thermal noise code tracking jitter for a non-coherent DLL discriminator.

Regarding the snapshot acquisition on E6-B, the well-known Cramer-Rao Lower Bound (CRLB) for time delay estimation can be applied (see Chapter 3 of [128]). This bound establishes the minimum possible variance of the time delay estimation using a Maximum Likelihood Estimation (MLE), which is equivalent to estimating the code phase delay by testing all possible delay values during the receiver's acquisition stage. The CRLB depends on the SNR at the output of the correlators, determined by the C/N_0 and the coherent integration time used for the correlation, denoted T_{int}, as well as the quadratic mean square bandwidth, represented by the Gabor bandwidth or Mean Square bandwidth, denoted B_{ms}. Therefore, based on this CRLB, the ranging accuracy for snapshot-based positioning can be expressed as:

$$\sigma_{\mathrm{snp}} \geq c\sqrt{\frac{1}{2C/N_0T_{\mathrm{int}}B_{\mathrm{ms}}}}[\mathrm{m}] \tag{5.14}$$

Alternative approaches can be considered to derive particularized expressions of the CRLB for GNSS signals (see Appendix D of [129]). Table 5.7 summarizes the attainable accuracies for various combinations of snapshot sizes and C/N_0 ratios. A 10 MHz receiver bandwidth is assumed, capturing the entire main lobe of the E6-B signal spectrum.

Snapshot size	$\mathbf{C/N_0}$	Accuracy (σ_{snp})
25 ms	45 dB-Hz	0.7 m
25 ms	40 dB-Hz	1.3 m
50 ms	40 dB-Hz	0.9 m
100 ms	40 dB-Hz	0.6 m
100 ms	35 dB-Hz	1.1 m
150 ms	35 dB-Hz	0.9 m

Table 5.7: Maximum code phase delay accuracies attainable for various E6 snapshot sizes.

In Section 6.4.3, an estimation of these accuracies is provided based on real data samples.

It should be noted that in the absence of data symbol information, non-coherent integration must be used for E6-B, which accounts for certain squaring losses compared to coherent integration when computing the previous accuracies. However, these losses are negligible for the C/N_0 values used in this analysis and only have a significant impact in indoor scenarios. Despite this, the accuracy values obtained remain comparable to those achieved using a DLL, primarily due to the high Gabor bandwidth of the E6-B signal.

Of course, in snapshot positioning, the receiver can only estimate the code phase delays from the acquisition process, which is not sufficient to obtain absolute pseudorange values. Therefore, a snapshot receiver must resolve this code phase ambiguity to achieve a full positioning solution. Several techniques to address this challenge are explored in Chapter 4 of [130]. A method for instantaneously computing a snapshot position and time solution without any reference time or position is detailed in [131].

5.6 Conclusions

In this chapter, we introduced the nominal operating mode for SAS, which leverages the E1-B signal to enhance RECS detection on E6-C. This is achieved by first obtaining initial estimates for the code phase delay and Doppler frequency from the E1-B signal and subsequently using these estimates to compute the corresponding values for E6-C. As a result, the search space for E6-C is reduced to just a few samples. In contrast, the generic approach presented in Chapter 4 relies on the receiver's clock uncertainties to define the search space. However, in the nominal approach, a time reference can be extracted from E1-B, effectively eliminating this dependence.

The MATLAB simulator used to determine the impact of the RECS Length and RECS Period in the generic approach is here used to determine such impact when using the estimates provided by E1-B. Also, additional simulations are provided with a more realistic scenario that considers typical impairments for a GNSS receiver.

The MATLAB simulator, previously used to assess the impact of RECS Length and RECS Period in the generic approach, is now employed to evaluate these effects when utilizing the estimates provided by E1-B. Additionally, further simulations are conducted under a more realistic scenario that accounts for typical impairments encountered in a GNSS receiver.

The authentication mechanism proposed for the nominal operating mode is also analyzed. This mechanism is based on computing the range difference between the code phase delay obtained from E6-C and the one estimated from E1-B. When such difference is less than a predefined threshold, the measurement can be authenticated.

The threats that an SAS receiver operating in this nominal mode may face are also examined. In this case, the E1-B signal could be spoofed as well, requiring the receiver to account for additional threat scenarios. However, the VSS algorithm can leverage the E1-B signal to narrow down the search for the authentic signal in the E6-C band, improving detection accuracy and mitigating the impact of potential spoofing attacks.

Finally, a proposal for using the E6-B signal to enhance SAS is introduced. Since E6-B does not exhibit the inter-frequency bias present between E1-B and E6-C, it allows for more accurate estimates of both the code phase delay and Doppler frequency. Consequently, the authentication mechanism—based on the range difference between these estimates—can be refined and improved. Additionally, if the TESLA keys are obtained through an alternative path rather than the E1-B signal, an SAS receiver could operate solely on the E6 band, in a snapshot-based approach. This would eliminate the need for dual-frequency receivers that process both E1 and E6 bands.

Chapter 6

Evaluating SAS Performance with Experimental Platform

6.1 Introduction

In Chapters 4 and 5, the performance of SAS at the signal level is evaluated using a custom-built MATLAB simulator, detailed in Section 4.3. This simulator provides a high degree of control over all processes, particularly in configuring the parameters involved. However, the synthetic data it generates is only an approximation of the impairments affecting real signals. Using highly accurate models can minimize the disparities between synthetic and real signals, albeit at the cost of increased complexity or expense.

This chapter expands the performance evaluation to real signals using a custom-built evaluation platform based on low-cost SDRs. This approach enables testing SAS in real-world scenarios and supplements the results obtained with synthetic data. The use of low-cost SDRs allows for relatively rapid development while keeping costs low.

The first part of this chapter describes the platform in detail, including all configuration aspects. Following this, the real datasets obtained with the platform are discussed. Finally, the results obtained from these datasets are presented.

It should be noted that, at the time of writing, the Galileo E6-C signal is broadcast unencrypted. Therefore, the results presented here have been obtained using the existing open signals and should be considered preliminary. Encryption of the E6-C signal is expected by the end of 2024 or beginning of 2025.

6.2 SDR Evaluation Platform

As discussed in Chapter 5, the proposed nominal operating mode for SAS utilizes the E1-B signal to reduce the uncertainties associated with the E6-C signal, enabling precise localization of RECSs. Therefore, the alignment between E1-B and E6-C estimates, particularly concerning code phase and Doppler frequency, is critical for this operation.

This section outlines the evaluation platform and provides instructions on configuring it to facilitate synchronous acquisition of E1 and E6 samples. This setup aims to validate the assumptions made in the proof-of-concept of the SAS nominal operating mode [1] using real measurements.

6.2.1 Hardware Description

To shorten the development time and reduce costs, we implemented our evaluation platform using Commercial Off-The-Self (COTS) SDR devices from the market. Given that both the E1-B and E6-C signals are integral to SAS processing, we first review the main characteristics of these signals, which are described in Section 3.3 and summarized in Table 6.1.

Signal	**Carrier Frequency**	**Reference Bandwidth**	**Code Chip Rate**
E1-B	1575.42 MHz	24.552 MHz	1.023 Mcps
E6-C	1278.75 MHz	40.920 MHz	5.115 Mcps

Table 6.1: Main Galileo E1 & E6 signals characteristics.

In light of the range of existing technical solutions, we first determine a list of minimum requirements to narrow the search among the available SDR boards on the market:

- Frequency range encompassing the L-band (1 to 2 GHz).
- Minimum sampling rate of 60 MSps (to ensure sufficient accuracy for the envisaged tests).
- Minimum ADC resolution of 8 bits (to provide adequate dynamic ranges for the envisaged tests).
- Low-cost board (less than $500 per board).
- External clock reference input.

After a preliminary selection process that considered non-technical factors such as purchasing convenience, the bladeRF 2.0 micro board from Nuand was chosen for our platform, due to its focus on developing cost-effective SDR platforms. The main specifications of this board are summarized in Table 6.2, along with those of other candidates (LimeSDR [132] and Analog Devices [133]) that met all the requirements listed above. A comprehensive evaluation of available SDR boards can be found in [134], within the SDR Makerspace project [135]. For the latest information on GNSS SDR, including different receiver architectures and front ends, see [24], [136]. Additionally, for comparison purposes, [137] presents a signal authentication device based on snapshot processing for SAS.

Board Model	Transceiver	Max.Rate[1]	Max.BW[2]	ADC Res.[3]	USB Ver.[4]	Price[5]
bladeRFmicro	AD9361	61 MSps [6]	56 MHz	12 bits	USB 3.0	$400
LimeSDR	LMS7002M	61 Msps	61 MHz	12 bits	USB 3.0	$300
ADRV9364	ADS9364	61 Msps	56 MHz	12 bits	USB 2.0 [7]	$800

[1] Maximum sampling rate. [2] Maximum radio-frequency bandwidth. [3] Analog to Digital Conversion resolution. [4] Device connection (USB version). [5] Approximative prices obtained in 2022, in USD.
[6] This rate can be extended up to 122.88 MHz using the 8-bit mode support of the AD9361 transceiver integrated in the board, a feature released in February 2023 [138].
[7] PCIe and Ethernet interfaces with faster speeds are also available.

Table 6.2: Candidate SDR boards specifications.

Going into further detail, the bladeRF 2.0 micro boards (see Figure 6.2) are 2 × 2 Multiple-Input Multiple-Output (MIMO) SDR boards offering a frequency range of 47 MHz to 6 GHz and a maximum sampling rate of 61.44 MHz with a resolution of 12 bits per sample. The core of the board is the latest generation Cyclone V Field Programmable Gate Array (FPGA) from Intel (formerly Altera). The FPGA size, measured in Logic Elements (LEs), varies depending on the bladeRF board model. The models used for our platform are the ×A4, with 49 K LEs, and the ×A5, 72 K LEs. The complete list of specifications can be found in [139] and the schematics provided by the manufacturer is shown in Figure 6.1.

All the boards include an on-board PLL which allows for controlling its Voltage-Controlled Temperature-Compensated Crystal Oscillator (VCTCXO) to a 10MHz reference signal, but they can also use an external reference clock source through a dedicated surface-mounted U.FL connector. The board can be powered solely from a Universal Serial Bus (USB) (3.0 type), but an external power source can be supplied to ensure maximal linear performance of bias-tee peripherals if needed.

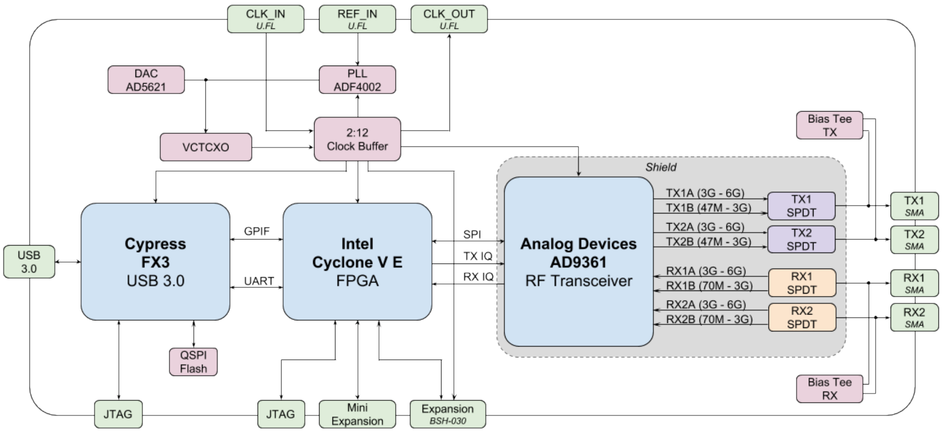


Figure 6.1: bladeRF SDR board schematics © Nuand.

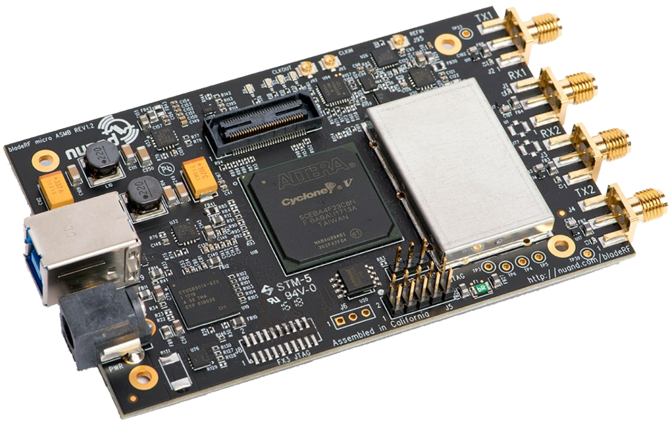


Figure 6.2: bladeRF SDR board © Nuand.

6.2.2 Platform Architecture

Even though the bladeRF micro 2.0 boards include 2 × 2 MIMO capabilities, featuring two transmitters and two receiver modules that share the same oscillator, they cannot be used to acquire samples from both the E1 and E6 bands simultaneously. This limitation arises because the E1 band is located in the upper L-band, while the E6 band is in the lower L-band. Consequently, two separate boards are required to record the E1 and E6 samples synchronously.

The two boards are connected to a non-powered Drotek multi-band antenna [140], which covers Galileo E1 and E6 frequency bands, using a power splitter. One of the boards powers the antenna via its built-in bias-tee, while a DC-block is used on the other board to prevent any damages.

Each board has its own clock reference operating at 34 MHz. However, to avoid mismatches arising from using different clock sources, both boards are connected to a common external reference via the J93/J95 test points on the bladeRF boards [141]. Using an external reference also allows for a high degree of control over clock stability, especially when using very stable sources like those provided by an OCXO. The model used in our platform is the Citrine version from Wenzel [142], which provides a 10 MHz clock with an aging rate of just 5 nanoseconds per day. It is worth noting that, before running any tests with the evaluation platform, the OCXO should be warmed up for a few minutes to achieve temperature stability.

Finally, the J51 test points of each board are connected using a jumper wire to facilitate communication between the boards, enabling synchronization tasks. A block diagram of the platform is shown in Figure 6.3, and a prototype is depicted in Figure 6.4.

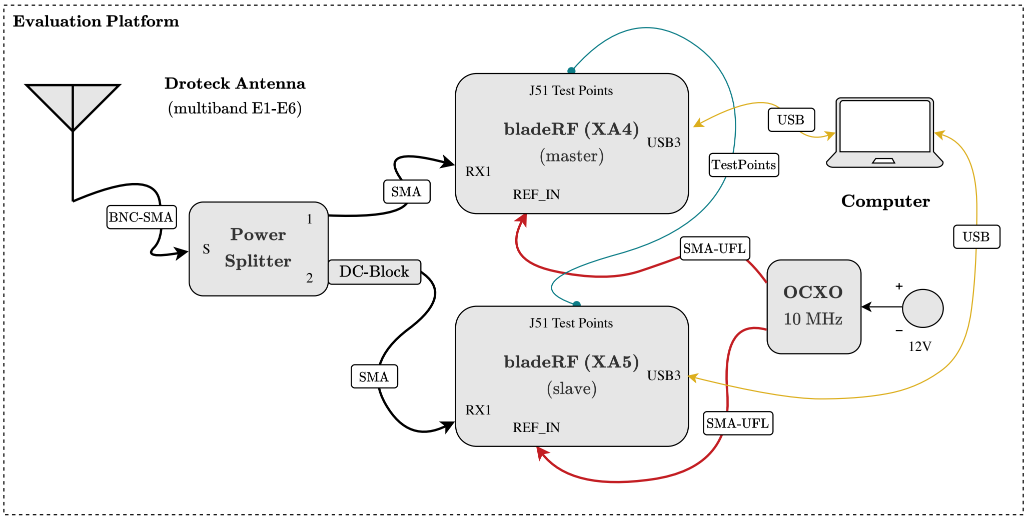


Figure 6.3: Block diagram of the SDR evaluation platform.

6.2.3 Platform Configuration

The bladeRF boards in the platform were configured using the native libbladeRF library from Nuand. Higher-level tools, such as the open-source SoapySDR Application Programming Interface (API) [143], which includes a library for interfacing with various SDR devices, can also be used. Nuand provides a basic installation guide [144] for the

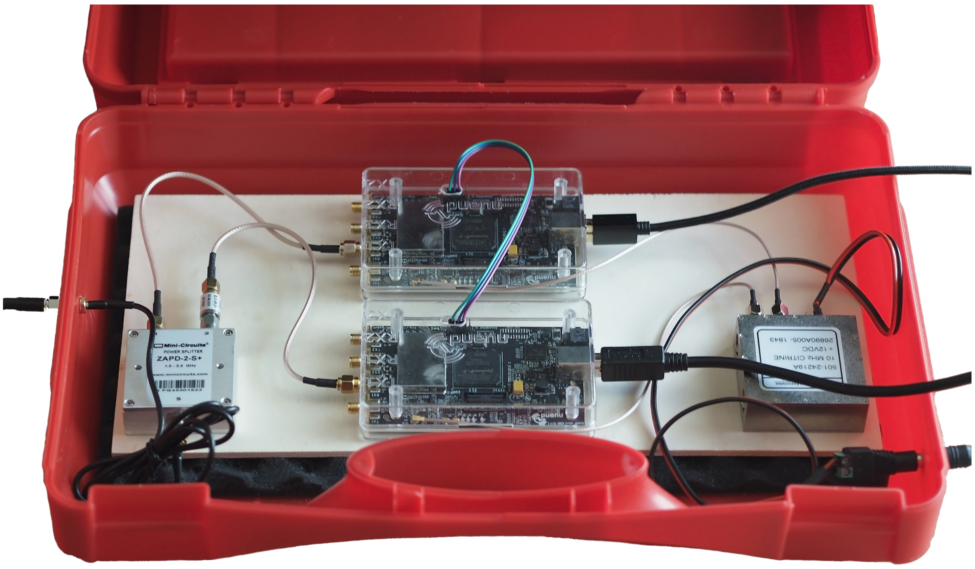


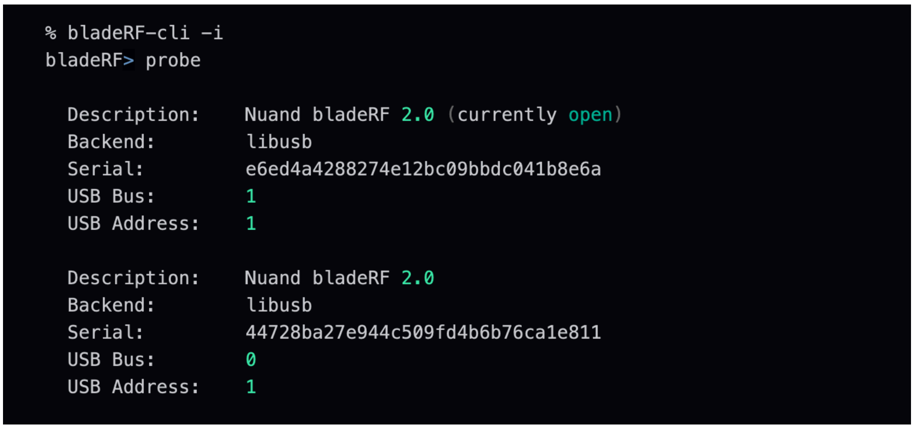
Figure 6.4: Prototype of the SDR evaluation platform.

libbladeRF library, which can be installed on multiple computer platforms. On the macOS platform chosen in this thesis, the MacPorts utility [145] was used. Both Intel- and ARM-based platforms have been successfully tested.

The bladeRF library supports two modes of operation: interactive and non-interactive. The non-interactive mode allows for batch-processing configuration commands from the Command Line Interface (CLI). However, we have encountered inconsistencies with the latest version at the time of writing, where some parameters were repeatedly misconfigured. Consequently, the more reliable interactive mode has been used, which can be activated using the command `` `bladeRF-cli --interactive` `` from the host computer terminal.

The first step is to individually configure the boards. In our setup, the xA5-model board, labeled as the master, is assigned to acquire the E6 samples, while the xA4-model board, labeled as the slave, acquires the E1 samples. After successfully connecting the boards to the host, we can verify the connection details using the `` `probe` `` command, as shown in Figure 6.5.

Now we can check the status of the FPGA by using the `` `version` `` command, which displays the size of the FPGA and whether it is loaded. If the FPGA is not loaded, the corresponding image can be downloaded from [146]. To load the FPGA, the command



```
% bladeRF-cli -i
bladeRF> probe

  Description:    Nuand bladeRF 2.0 (currently open)
  Backend:        libusb
  Serial:         e6ed4a4288274e12bc09bbdc041b8e6a
  USB Bus:        1
  USB Address:    1

  Description:    Nuand bladeRF 2.0
  Backend:        libusb
  Serial:         44728ba27e944c509fd4b6b76ca1e811
  USB Bus:        0
  USB Address:    1
```

Figure 6.5: bladeRF boards connections.

`` `load-fpga [filename.rbf]` `` is used [1].

To avoid the need for manually loading the FPGA every time the system is powered on, Nuand provides an autoloading mechanism [147]. This mechanism can be either host-software-based or firmware-based. In the former, the libbladerf library checks if the FPGA image file is available in a list of predefined folders in the host, and loads it if found. In the latter, which is host-independent but slightly slower, the FPGA bitstream needs to be written into the device's SPI flash. This is accomplished with the command `` `flash-firmware [filename.img]` ``. The latest FX3 firmware can also be downloaded from [148].

We can now configure the receiver parameters on the boards, either by inputting the commands individually or by saving them in a script and running it with the command `` `run [filename.script]` ``. The contents of the scripts for both the master and slave boards are provided in Listing 6.1 and Listing 6.2, respectively. The results of applying these scripts is shown in Figure 6.6.

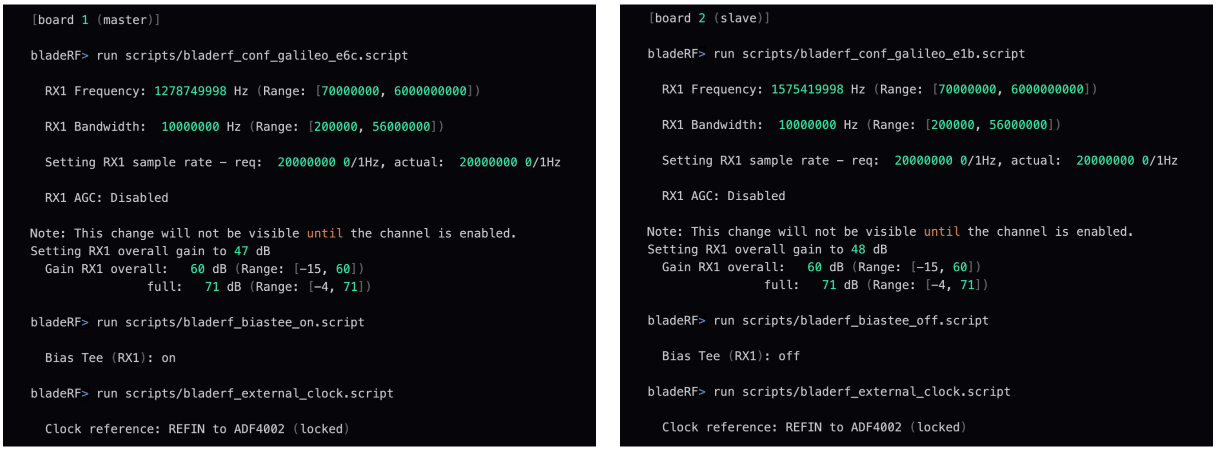
[1] The version used in this thesis is the v0.14.0, which was released on April 4, 2021. At the time of writing, the latest version available is v0.15.0 (released on February 13, 2023), which mainly adds support for oversampling and, therefore, is not expected to affect the results shown later.

```
set frequency rx 1278.75 M
set bandwidth rx 10 M
set samplerate rx 20 M
set agc rx off
set gain rx 45
set biastee on
set clock_ref enable
```

Listing 6.1: bladerf_conf_master.script

```
set frequency rx 1575.42 M
set bandwidth rx 10 M
set samplerate rx 20 M
set agc rx off
set gain rx 46
set biastee off
set clock_ref enable
```

Listing 6.2: bladerf_conf_slave.script

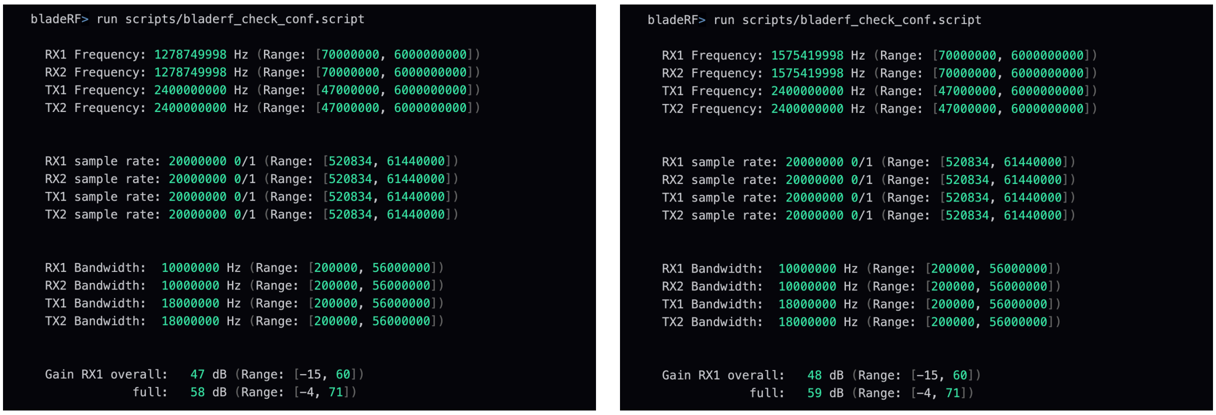


```
[board 1 (master)]

bladeRF> run scripts/bladerf_conf_galileo_e6c.script

  RX1 Frequency: 1278749998 Hz (Range: [70000000, 6000000000])

  RX1 Bandwidth:  10000000 Hz (Range: [200000, 56000000])

  Setting RX1 sample rate - req:  20000000 0/1Hz, actual:  20000000 0/1Hz

  RX1 AGC: Disabled

Note: This change will not be visible until the channel is enabled.
Setting RX1 overall gain to 47 dB
  Gain RX1 overall:   60 dB (Range: [-15, 60])
                full:   71 dB (Range: [-4, 71])

bladeRF> run scripts/bladerf_biastee_on.script

  Bias Tee (RX1): on

bladeRF> run scripts/bladerf_external_clock.script

  Clock reference: REFIN to ADF4002 (locked)
```

```
[board 2 (slave)]

bladeRF> run scripts/bladerf_conf_galileo_e1b.script

  RX1 Frequency: 1575419998 Hz (Range: [70000000, 6000000000])

  RX1 Bandwidth:  10000000 Hz (Range: [200000, 56000000])

  Setting RX1 sample rate - req:  20000000 0/1Hz, actual:  20000000 0/1Hz

  RX1 AGC: Disabled

Note: This change will not be visible until the channel is enabled.
Setting RX1 overall gain to 48 dB
  Gain RX1 overall:   60 dB (Range: [-15, 60])
                full:   71 dB (Range: [-4, 71])

bladeRF> run scripts/bladerf_biastee_off.script

  Bias Tee (RX1): off

bladeRF> run scripts/bladerf_external_clock.script

  Clock reference: REFIN to ADF4002 (locked)
```

Figure 6.6: bladeRF boards configuration scripts.

It is also possible to verify that the boards have been correctly configured, as shown in Figure 6.7.



```
bladeRF> run scripts/bladerf_check_conf.script

  RX1 Frequency: 1278749998 Hz (Range: [70000000, 6000000000])
  RX2 Frequency: 1278749998 Hz (Range: [70000000, 6000000000])
  TX1 Frequency: 2400000000 Hz (Range: [47000000, 6000000000])
  TX2 Frequency: 2400000000 Hz (Range: [47000000, 6000000000])


  RX1 sample rate: 20000000 0/1 (Range: [520834, 61440000])
  RX2 sample rate: 20000000 0/1 (Range: [520834, 61440000])
  TX1 sample rate: 20000000 0/1 (Range: [520834, 61440000])
  TX2 sample rate: 20000000 0/1 (Range: [520834, 61440000])


  RX1 Bandwidth:  10000000 Hz (Range: [200000, 56000000])
  RX2 Bandwidth:  10000000 Hz (Range: [200000, 56000000])
  TX1 Bandwidth:  18000000 Hz (Range: [200000, 56000000])
  TX2 Bandwidth:  18000000 Hz (Range: [200000, 56000000])


  Gain RX1 overall:   47 dB (Range: [-15, 60])
              full:   58 dB (Range: [-4, 71])
```

```
bladeRF> run scripts/bladerf_check_conf.script

  RX1 Frequency: 1575419998 Hz (Range: [70000000, 6000000000])
  RX2 Frequency: 1575419998 Hz (Range: [70000000, 6000000000])
  TX1 Frequency: 2400000000 Hz (Range: [47000000, 6000000000])
  TX2 Frequency: 2400000000 Hz (Range: [47000000, 6000000000])


  RX1 sample rate: 20000000 0/1 (Range: [520834, 61440000])
  RX2 sample rate: 20000000 0/1 (Range: [520834, 61440000])
  TX1 sample rate: 20000000 0/1 (Range: [520834, 61440000])
  TX2 sample rate: 20000000 0/1 (Range: [520834, 61440000])


  RX1 Bandwidth:  10000000 Hz (Range: [200000, 56000000])
  RX2 Bandwidth:  10000000 Hz (Range: [200000, 56000000])
  TX1 Bandwidth:  18000000 Hz (Range: [200000, 56000000])
  TX2 Bandwidth:  18000000 Hz (Range: [200000, 56000000])


  Gain RX1 overall:   48 dB (Range: [-15, 60])
              full:   59 dB (Range: [-4, 71])
```

Figure 6.7: bladeRF boards configuration verification.

We can observe that the bias-tee is activated in the master board, as it is responsible for feeding the antenna. The AGC is turned off, and the receiver gain is manually adjusted to a predetermined value that maximizes the board ADC's dynamic range. Additionally, the gain is slightly increased in the slave board to compensate for the attached DC-block's losses. It is worth noting that the bladeRF micro 2.0 board converter has a native format of signed, complex 16-bit Q11, which implies that the values are within the range of $(-2048, 2047)$. Finally, the last command in the script allows the use of the OCXO as the external clock reference.

The next step is to configure the boards so that they can acquire samples synchronously. To do so, we start defining the test points that will be used for triggering, first in the master board and then in the slave board. This is accomplished using the scripts provided in Listings 6.3 and 6.4.

If the `timeout` option is omitted, as in our scripts, a timeout of 1 second is used by default. This is typically enough when the commands are automated in a routine, but if the if the commands are executed manually, a timeout error is reported if the trigger on the master board is executed more than one second later than the slave receiver's configuration command. In such case, a larger timeout can be configured (e.g. 10 s), appending the option `` `timeout=10s` `` in the `` `rx config` `` command.

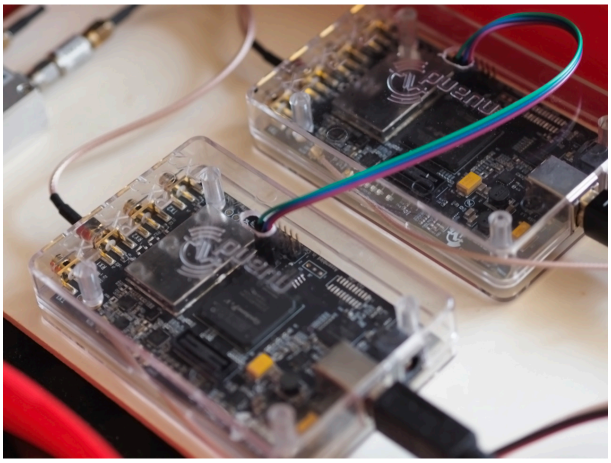


Figure 6.8: Detail of the connection of J51 pins between the two SDR boards that allow the synchronous capture of samples using the specified triggering commands.

Finally, we need to fire the trigger back into the master board to start acquiring samples synchronously, which is done with the command `` `trigger j51-1 tx fire` ``. The execution of these commands in the macOS terminal application is illustrated in Figure 6.9.

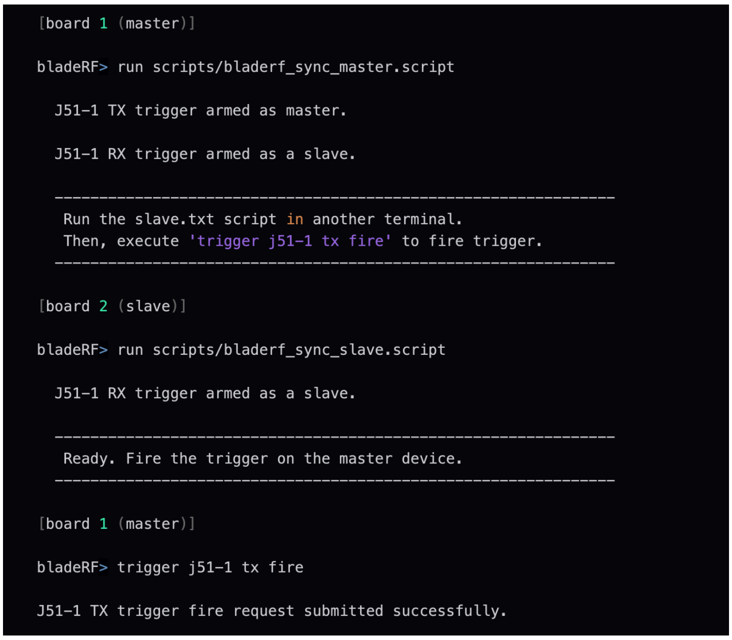
```
rx config file=master_e6.bin n=200M
trigger j51-1 rx master
rx start
```

Listing 6.3: bladerf_rx_sync_master.script

```
rx config file=slave_e1.bin n=200M
trigger j51-1 rx slave
rx start
rx wait
```

Listing 6.4: bladerf_rx_sync_slave.script

The `` `rx config` `` command allows for the specification of the filename where the samples will be recorded, as well as the total number of samples to be recorded. If not specified, the native binary format will be used. However, it is also possible to use the Comma-Separated Values (CSV) format using the option `` `format=csv` `` in the `` `rx config` `` command. In our case, since we have configured the boards with a sampling rate of 20 MHz, the 200 million samples (`` `n = 200 M` ``) to be acquired correspond to a 10 s snapshot. Therefore, two files (one for E6, one for E1) will be generated, each approximately 800 MB in size, taking into account that each sample (I and Q) is coded in 2 bytes.



```
[board 1 (master)]

bladeRF> run scripts/bladerf_sync_master.script

  J51-1 TX trigger armed as master.

  J51-1 RX trigger armed as a slave.

  ----------------------------------------------------------------
   Run the slave.txt script in another terminal.
   Then, execute 'trigger j51-1 tx fire' to fire trigger.
  ----------------------------------------------------------------

[board 2 (slave)]

bladeRF> run scripts/bladerf_sync_slave.script

  J51-1 RX trigger armed as a slave.

  ----------------------------------------------------------------
   Ready. Fire the trigger on the master device.
  ----------------------------------------------------------------

[board 1 (master)]

bladeRF> trigger j51-1 tx fire

J51-1 TX trigger fire request submitted successfully.
```

Figure 6.9: Execution of bladeRF triggering commands in macOS terminal application.

It is important to note that the synchronization feature used in this evaluation platform was not documented originally in the official Nuand documentation for the bladeRF micro 2.0 boards. A similar feature is available for other boards from the same manufacturer [149], which has been used to infer the existence of this feature for the board used in our platform. Later, at the time of writing, the official documentation has been updated accordingly.

6.3 Test Datasets

In this section we describe the test datasets used to evaluate the SAS performance under real world scenarios. These datasets have been obtained using the SDR-based evaluation platform described in Section 6.2.

6.3.1 Real Datasets using Existing Open Signals

The real datasets used in this thesis were obtained in a rural area with clear-sky scenario. The recording spot is located near Girona (Spain), at a latitude of 41º59’35” N (41.9932) and a longitude of 2º47’43” E (2.7954). Each of the datasets recorded comprises two files, corresponding to the snapshots of E1 and E6 samples captured synchronously with the evaluation platform.

Two batches of datasets are used in the thesis for testing purposes. The first batch was recorded during February 20th and 22nd of 2023. To accommodate the E6-C BPSK(5) main lobe (10 MHz two-sided bandwidth), all the datasets were recorded at a sampling rate of 20 MHz, excepting the first dataset, which was recorded near at the maximum sampling rate of the bladeRF SDR boards (i.e., 60 MHz at full resolution). The datasets B1-B2-B3-B4 were recorded every roughly three minutes, the time of switching off and on the power, so that different power cycles were tested. The specific data for each dataset comprised in the first batch is summarized in the Table 6.3.

Dataset ID	**Recording Date/Time**	**Duration**	**Sampling rate**
A0	2023-02-20 16:24 (GMT)	10 seconds	60 MHz
B1	2023-02-22 12:26 (GMT)	10 seconds	20 MHz
B2	2023-02-22 12:29 (GMT)	10 seconds	20 MHz
B3	2023-02-22 12:31 (GMT)	10 seconds	20 MHz
B4	2023-02-22 12:34 (GMT)	10 seconds	20 MHz

Table 6.3: Information about the real datasets used in the thesis (first batch).

To emulate scenarios with lower carrier-to-noise ratios, a second batch of datasets were recorded in the same spot during April 12th and 14th of 2023. In this case, the antenna was covered under building blocks of wood or concrete with different set-ups. The specific data for each dataset comprised in the seconds batch is summarized in the Table 6.4.

Dataset ID	Recording Date/Time	Duration	Sampling rate
C0	2023-04-12 11:54 (GMT)	8 seconds	20 MHz
D0	2023-04-14 14:21 (GMT)	3.2 seconds	20 MHz

Table 6.4: Information about the real datasets used in the thesis (second batch).

The sky plots of visible Galileo satellites for the C0 and D0 datasets with an elevation not lower than 20º are shown in Figure 6.10 and Figure 6.11, which has been obtained with the web tool "GNSS-Radar" [150]. In Table 6.5 and Table 6.6 we list these satellites with their corresponding azimuth and elevation. The orbital and technical parameters for the Galileo satellites can be found in [151].

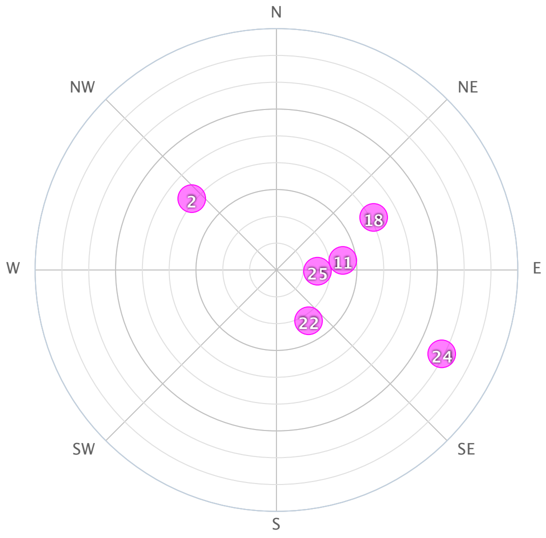


Figure 6.10: Visible Galileo satellites for dataset C0.

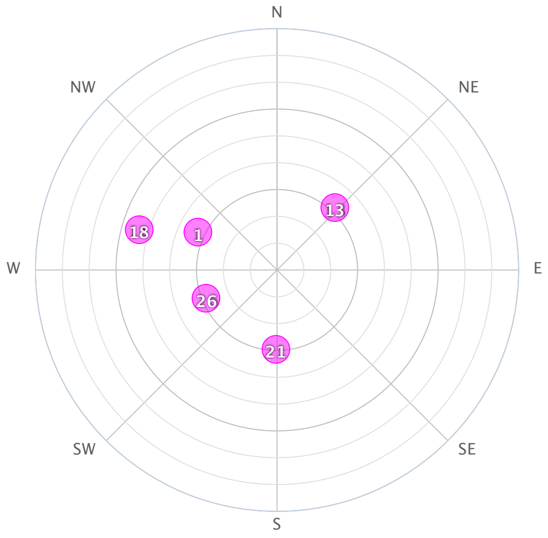


Figure 6.11: Visible Galileo satellites for dataset D0.

SVID	Azimut	Elevation
E02	310.1º	48.7º
E11	81.8º	65.0º
E18	61.6º	48.8º
E22	147.7º	67.6º
E24	113.4º	24.3º
E25	75.1º	77.0º

Table 6.5: Nominal Galileo satellites in dataset C0.

SVID	Azimut	Elevation
E01	295.6º	57.2º
E13	42.9º	58.2º
E18	286.3º	36.6º
E21	180.8º	60.4º
E26	248.4º	61.6º

Table 6.6: Nominal Galileo satellites in dataset D0.

It is worth noting that both SVID 18 and SVID 22 are auxiliary Galileo satellites, which have been omitted hereafter for our analysis. All the datasets of real samples recorded with the SDR platform used for this thesis can be downloaded from UAB SPCOMNAV [152].

6.3.2 Snapshot C/N_0 Estimation

To evaluate the SAS performance in different scenarios, we first perform an estimation of the carrier-to-noise-density ratio of the E6-C signal for the snapshots recorded with the SDR platform. The estimator used is the following non-coherent post-correlation estimator presented in [22] and proposed in [24] for snapshot receivers:

$$\left(\frac{\hat{C}}{N_0}\right)_{\text{NC}} \doteq \frac{R_{\text{NC}}\left(\hat{\tau}_0, \hat{f}_{d,0}\right) B_n - T_{\text{int,coh}} F_s^2 \hat{P}}{\left(T_{\text{int,coh}} F_s\right)^2 \hat{P} - R_{\text{NC}}\left(\hat{\tau}_0, \hat{f}_{d,0}\right)} \tag{6.1}$$

where $R_{\text{NC}}\left(\hat{\tau}_0, \hat{f}_{d,0}\right)$ is the value of the (non-coherent) CAF for the estimated code phase delay $\hat{\tau}_0$ and the estimated Doppler frequency $\hat{f_{d,0}}$, B_n is the receiver noise equivalent bandwidth, $T_{\text{int,coh}}$ is the coherent integration time, F_s is the sampling rate, and $\hat{P}$ is an estimate of the input signal power.

The non-coherent CAF can be expressed as:

$$R_{\text{NC}}(\tau, f) \doteq \frac{1}{N_I} \sum_{k=0}^{N_I-1} |R_{\text{C}}(\tau, f; k)|^2 \tag{6.2}$$

where N_I is the number of non-coherent integrations and R_{C} the (coherent) CAF.

This estimator incorporates a pre-correlation estimate of the noise power, rendering it less sensitive to errors in code delay and Doppler frequency estimation typically encountered during the acquisition stage. Consequently, this attribute makes it more suitable for snapshot receivers compared to other traditional estimators.

As highlighted in [153], this estimator demonstrates good performance when the probability of acquisition is high, specifically for large C/N_0 ratios or extended integration times. Indeed, under such conditions, it exhibits a bias smaller than 1 dB, and the variances become very close to the Cramer-Rao Bound (CRB).

In Table 6.7 and Table 6.8 we show the C/N_0 estimates of the E6-C signal for all the visible satellites found in the real datasets used. The Secondary Code Index (SCI) for E6-C is also indicated, as it will be later used to emulate the required length for the RECS. The satellites selected for the preliminary evaluation are highlighted in red.

SVID	**C/N$_0$**	**SCI**
E02	48 dB-Hz	20
E11	43 dB-Hz	24
E24	35 dB-Hz	13
E25	43 dB-Hz	23

Table 6.7: Estimates for dataset C0.

SVID	**C/N$_0$**	**SCI**
E01	43 dB-Hz	12
E13	42 dB-Hz	12
E21	40 dB-Hz	12
E26	45 dB-Hz	13

Table 6.8: Estimates for dataset D0.

As anticipated, both the C/N_0 and SCI estimates are consistent with the positions of the visible satellites for each snapshot, where the lower C/N_0 values generally correspond to lower elevations. It is important to note that, due to the non-uniformity of the blocks covering the antenna, some satellites may exhibit different attenuations than expected with respect to their specific elevation.

6.3.3 Emulating RECS

As the E6-C signal is broadcast unencrypted at the time of writing, the RECSs need to be emulated to evaluate SAS performance with the existing open signals. To achieve this, an initial acquisition is performed on the E6-C samples contained in the corresponding snapshot recorded with the evaluation platform. This step allows for the estimation of the code phase delay by performing a standard correlation with the primary E6-C 1-ms spreading code sequences.

Next, to emulate an encrypted E6-C signal, the received E6-C samples are encrypted using a pseudo-random binary sequence, except for the samples where the RECSs are supposed to be, which depends on the selected SAS configuration. The start of the RECSs for a given period can be determined based on the code phase delay estimated in the previous acquisition.

As the RECS can last several milliseconds, each one will contain a given number of 1-ms E6-C spreading codes. Since the samples corresponding to the RECS are not encrypted, a coherent integration for the required 1-ms codes should be performed. However, the primary codes of the E6-C open signal are tiered with a secondary code that lasts 100 ms. Therefore, the E6-C secondary code must be acquired first. Once the SCI is estimated, the coherent integration can be extended to the effective length of the RECS, as per the selected configuration. Finally, the required estimates can be obtained, allowing for the performance evaluation of SAS at the signal level.

This is accomplished by adding a specific emulation function to the MATLAB SAS simulator presented in Section 4.3. This function performs the following tasks:

- Perform standard acquisition of E6-C open signal using primary E6-C codes to estimate the code phase delay.
- Encrypt the received E6 samples, except where the RECSs are supposed to be.
- Obtain the secondary code of E6-C.
- Perform acquisition on E6-C encrypted samples by integrating coherently the required 1-ms primary codes to match the configured RECS length.
- Obtain the required estimates and acquisition metrics for evaluating the SAS.

Technically speaking, the definition of the RECS length may not be an exact multiple of 1-ms blocks, but the difference could be neglected as there is no real impact in the performance evaluation. Additionally, it is worth noting that this emulation is done by encrypting the received samples, and not the transmitted chips, as it would be done if a real E6-C signal was broadcast already encrypted. This could led to slightly variations in the performance evaluation with respect testing the real encrypted signal.

6.4 Experimental Results

In this section, we provide a preliminary evaluation of the Galileo SAS using the existing open signals and the real datasets captured with the evaluation platform. First, we analyze the alignment of both E1 and E6 estimates, which is crucial for the SAS nominal operating mode, as it allows for a reduced acquisition search space. Then, we assess the performance of the acquisition at the signal level by computing the ROC curves for different SAS configurations.Finally, we provide an estimation of the code phase delay accuracies that can be achieved using snapshot positioning based on E6-B, complementing the analysis performed in Section 5.5.2.

6.4.1 E1-E6 Estimates Alignment

The goal of this analysis is to check the consistency the E1-B and E6-C estimates, for different C/N_0's and RECS lengths. To accomplish this, we use the SAS MATLAB simulator implemented in Section 4.3, that first divides the recorded snapshots into smaller chunks to be processed individually.

For each of these chunks, the simulator first performs an acquisition of the E1-B signal. The estimates obtained, specifically the code phase and Doppler frequency, are then used as the initial estimates for the acquisition of the E6-C signal. Thanks to this handover from E1-B, the search space is reduced to just a few correlations in the time domain. For the results presented in this section, the length of the acquisition window has been reduced to 20 samples, although fewer samples could be envisaged.

Next, we obtain the code phase and Doppler frequency estimates obtained in the E6-C and we compare them with the estimates obtained from E1-B. The difference obtained in samples is then converted to the equivalent in meters, in order to obtain the Range Error

(RE), which is calculated follows:

$$\mathrm{RE} = \frac{c}{F_s}\left[(\hat{\tau}_{0,1})_{\mathrm{mod}\, N_{\mathrm{scode}}} - \hat{\tau}_{0,6}\right] \quad [\mathrm{m}] \tag{6.3}$$

where F_s is the sampling rate, N_{scode} is the number of samples in a primary spreading code of E6-C, and $\hat{\tau}_{0,i}$ is the estimated code phase delay in samples for the Ei-th band. That is, for a sampling rate of 20 Msps, a difference of one sample corresponds to a Range Error of approximately 15 meters.

A comparison of the E1-B and E6-C correlation peaks for a given chunk extracted from B-datasets is shown in Figure 6.12. As expected, we observe that the width of the BOC (1,1) central peak used in E1-B is slightly wider but comparable to the BPSK(5) peak used in E6-C.

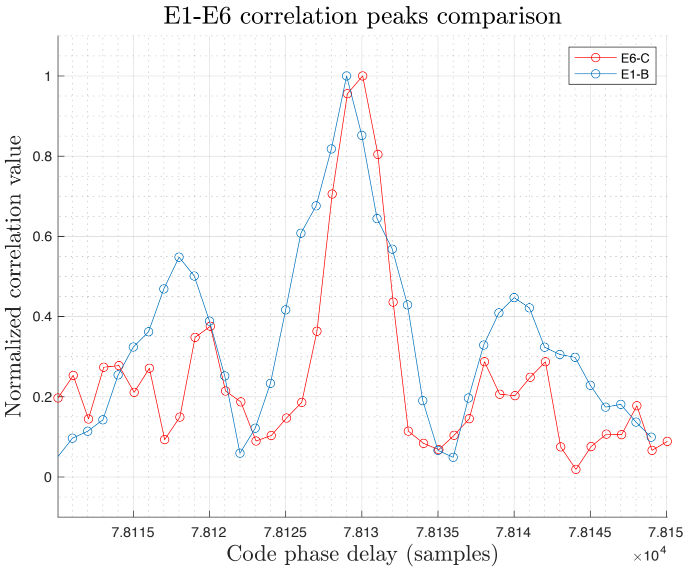
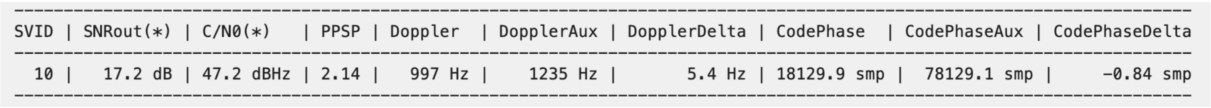


Figure 6.12: E1-E6 correlation peaks comparison.

The corresponding estimates provided by the MATLAB simulator are shown in Figure 6.13. As observed, the code phase difference (`CodePhaseDelta`) between the E6-C estimate (`CodePhase`) and the E1-B estimate (`CodePhaseAux`) is approximately one sample in this case. This value has been obtained by considering that the 1-ms E6-C spreading code lasts for 20,000 samples, so that 78,129.1 modulo 20,000 equals 18,129.1 samples, resulting in a difference of approximately 0.8 samples compared to the 18,129.9 samples that indicate the position of the E6-C correlation peak. In the case of the Doppler frequency, the difference (`DopplerDelta`) between the E1-B and E6-C estimates is obtained by considering the ratio of their respective carrier frequencies.



```
-----------------------------------------------------------------------------------------------------------------------
SVID | SNRout(*) | C/N0(*)   | PPSP | Doppler  | DopplerAux | DopplerDelta | CodePhase  | CodePhaseAux | CodePhaseDelta
-----------------------------------------------------------------------------------------------------------------------
  10 |   17.2 dB | 47.2 dBHz | 2.14 |   997 Hz |    1235 Hz |       5.4 Hz | 18129.9 smp |  78129.1 smp |     -0.84 smp
-----------------------------------------------------------------------------------------------------------------------
```

Figure 6.13: Estimates provided by the MATLAB simulator.

Once the simulator obtains the estimates for all the chunks of the snapshots comprised in the processed dataset, we can estimate a statistical distribution for both the code phase delay (or, equivalently, the range error) and the Doppler frequency.

It is worth noting that, in certain instances, one of the boards of the evaluation platform misses some samples in the recording process, so that the code phase difference is affected by this amount, which should be removed in the statics computation to obtain a fair comparison of the code phase estimates.

The first set of results is performed with the B-datasets [5]. In Figure 6.14, we show the results obtained with the four B-datasets, allowing for an analysis of the alignment with different power cycles. In this setup, each snapshot is divided into chunks of 4 ms, so that a total of 2000 estimates are obtained from the 10-seconds snapshots. Additionally, both the E6-C and E1-B estimates are obtained by processing a single spreading code, with durations of 1 ms and 4 ms, respectively.

The obtained distribution exhibits, as expected, a non-centered Gaussian-like shape, the variance of which is related to the sample noise. A Gaussian curve (in red) is fitted to the experimental data (in blue) to interpolate the mean and the standard deviation of the range error for each snapshot.

The results demonstrate a high level of consistency with respect to the spread of the estimates, with a standard deviation of around 2.5 m when only one E6-C spreading code is used for the acquisition. The observed bias primarily originates from the ionospheric effects and the hardware biases after power cycles resulting from the use of two distinct SDR boards in the platform.

In Figure 6.15(a), we compare the code phase delay between E1-B and E6-C using dataset A0. The higher sampling rate used in this dataset (60 Msps) allows for more precise estimates. In this case, we use the same coherent integration time of 4 ms for both E1-B and E6-C bands. Consequently, this involves performing a secondary code acquisition on E6-C to coherently combine four 1-ms primary codes. As a result, the

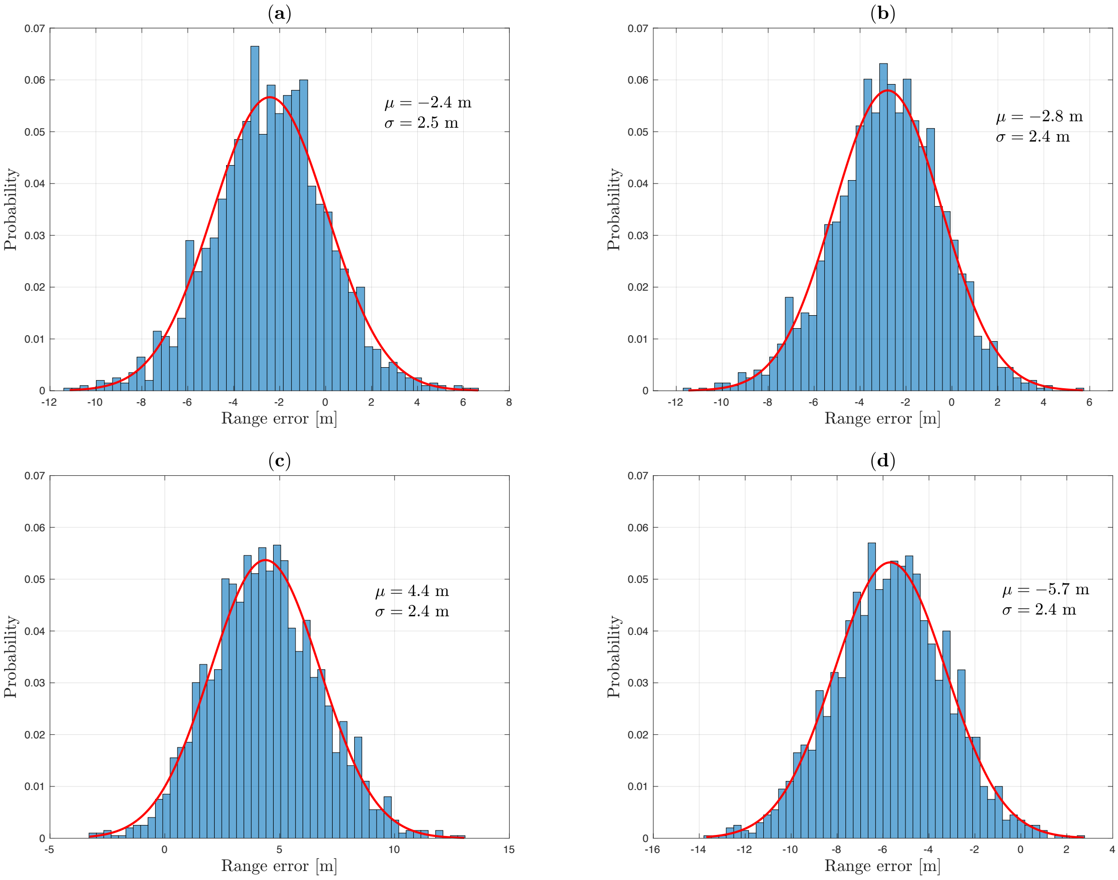


Figure 6.14: Code phase delay comparison (E1-B vs. E6-C) for datasets B1 (**a**), B2 (**b**), B3 (**c**), and B4.

standard deviation of the range error is reduced to less than 2 meters.

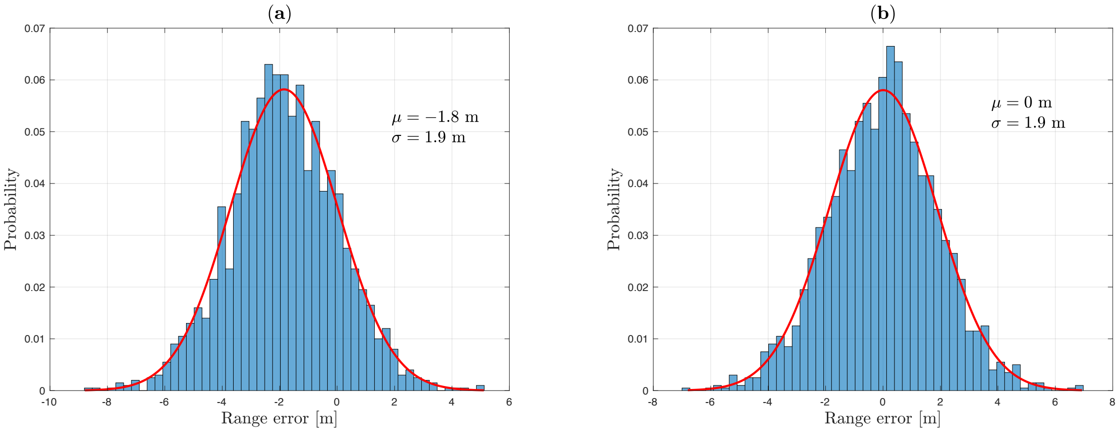


Figure 6.15: Code phase delay comparison for dataset A0: E1-B vs E6-C (**a**), and E6-B vs. E6-C (**b**).

To diminish the hardware bias produced by using two distinct boards, we also provide a comparison of the code phase delay between the E6-B and E6-C using the latest dataset, but now processing only the E6 snapshot from the same board. As we can observe in Figure 6.15(b), the obtained distribution exhibits a centered Gaussian-like shape in this case.

Regarding the Doppler frequency, a comparison of the estimates for both E6-C and E1-B bands is shown in Figure 6.16, using the real samples of dataset A0. For this comparison, we consider the relationship between both carrier frequency bands; hence, the Doppler frequency used for E6 is computed from the estimate obtained in E1 multiplied by the ratio f_{c_6}/f_{c_1}. The coherent integration time is 4 ms for both bands. The obtained results indicate that the differences are sufficiently small to justify using a reduced frequency search space in E6 based on the E1 estimate.

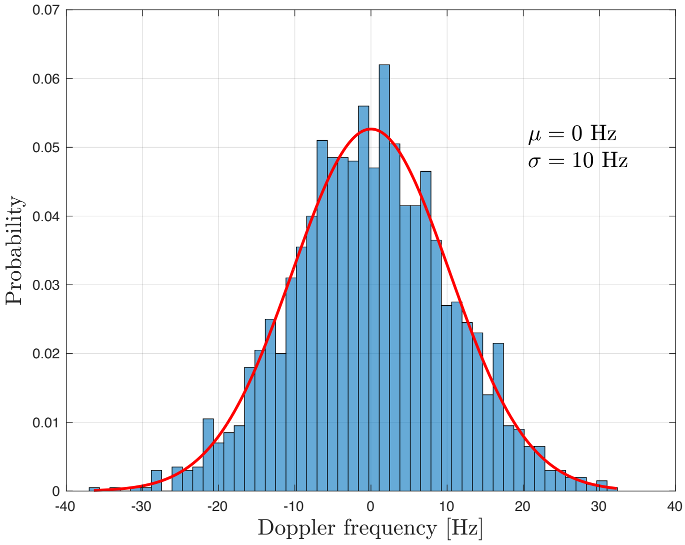


Figure 6.16: Doppler frequency comparison (E1-B vs. E6-C) for dataset A0.

The second set of results replicates the previous experiments under different C/N_0 scenarios, using datasets C0 and D0. Table 6.9 outlines the relationship between the figures, corresponding satellite/dataset used, and the estimated C/N_0 values, as determined by the estimator described in Section 6.3.2. For this analysis, the snapshots are divided into 16 ms chunks. Alongside the histogram of range error estimates, we also present the evolution of these errors over the snapshot duration.

Figure No.	Dataset ID	SVID	Estimated C/N_0	# chunks
Figure 6.17	D0	E26	45 dB-Hz	500
Figure 6.18	D0	E21	40 dB-Hz	500
Figure 6.19	C0	E24	35 dB-Hz	200

Table 6.9: E1-E6 alignment results.

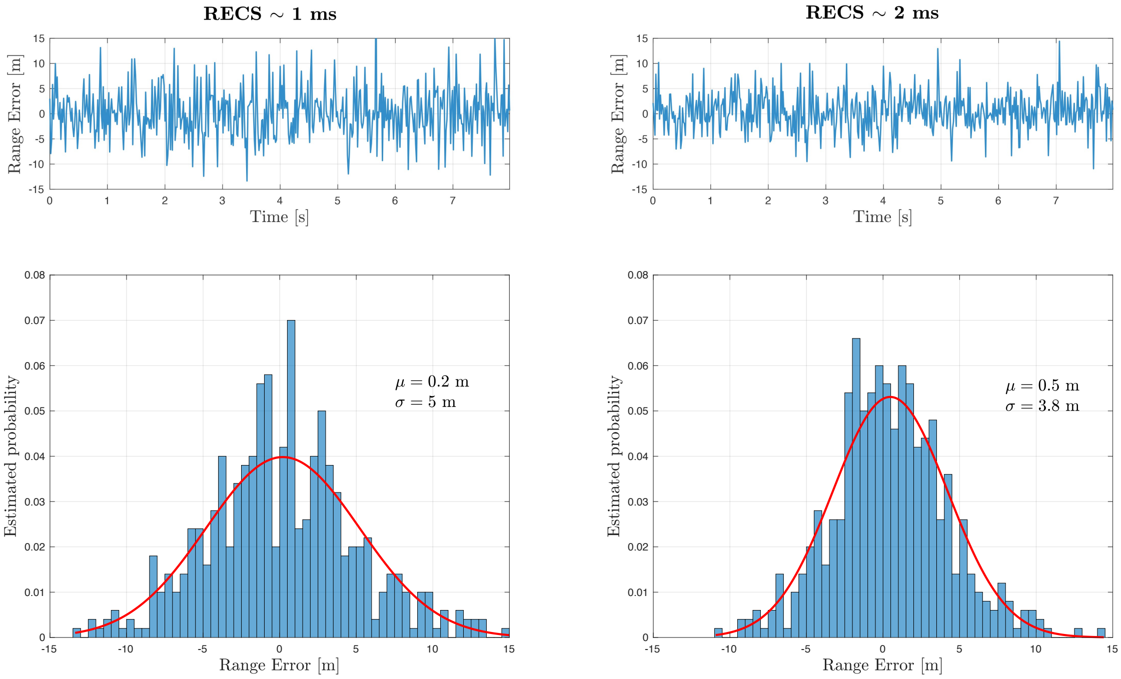


Figure 6.17: RE estimation (E1-B vs E6-C) for SVID 26 in dataset D0 using 1/2-ms RECS.

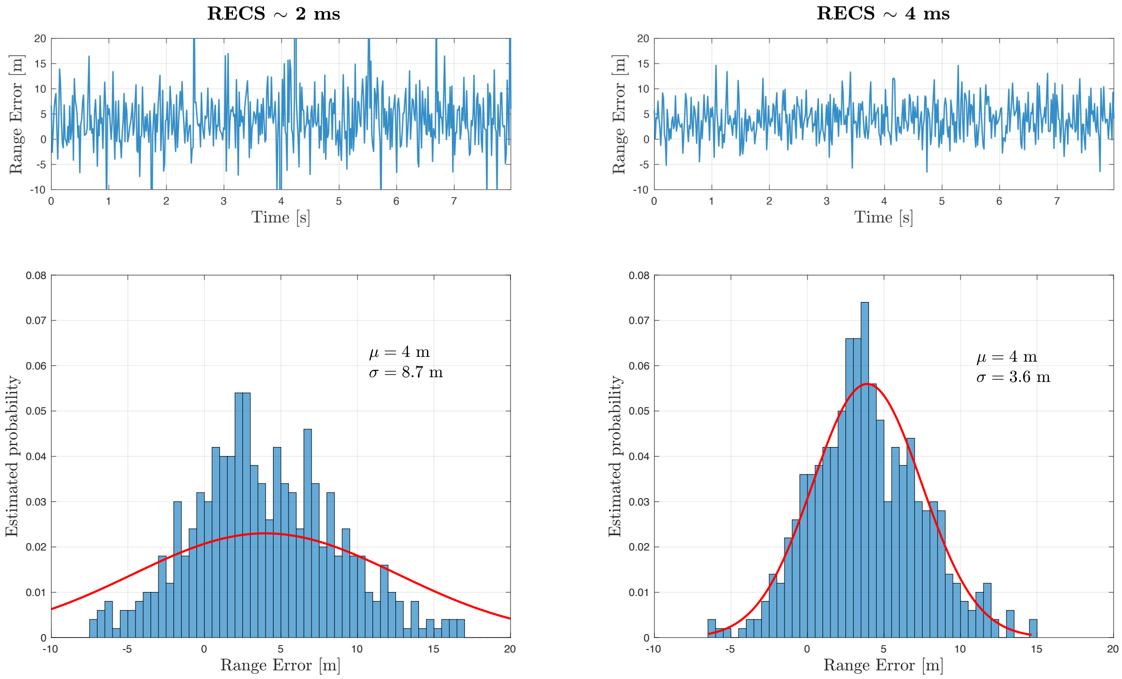


Figure 6.18: RE estimation (E1-B vs E6-C) for SVID 21 in dataset D0 using 2/4-ms RECS.

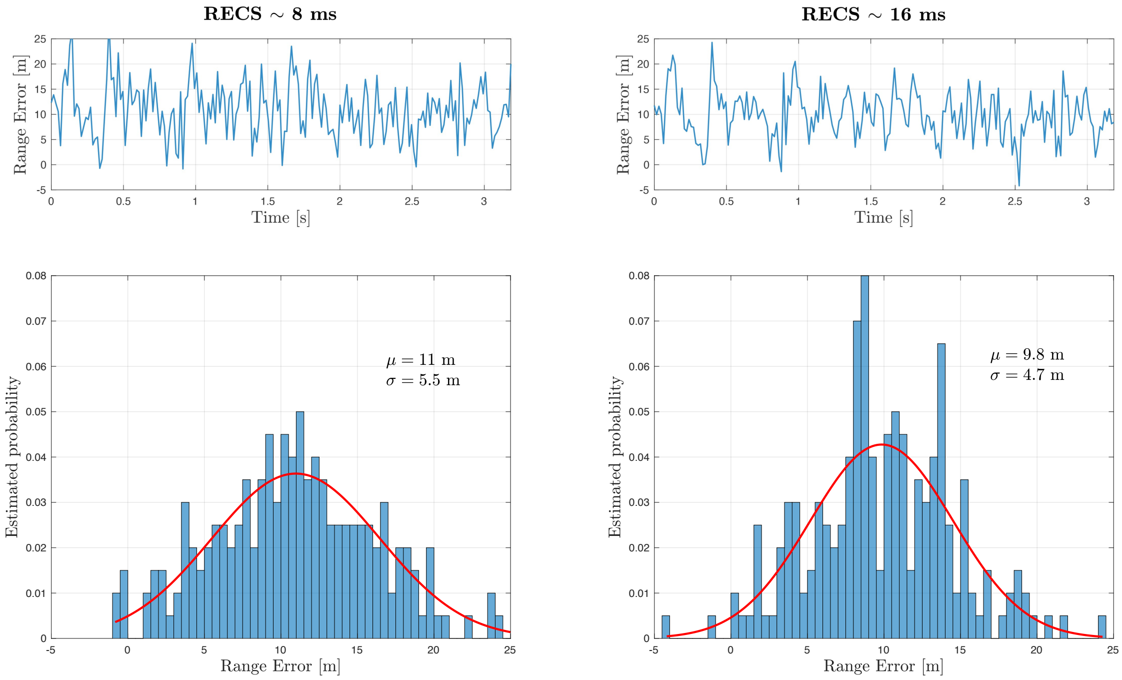


Figure 6.19: RE estimation (E1-B vs E6-C) for SVID 24 in dataset C0 using 8/16-ms RECS.

In Table 6.10 we summarize the results obtained regarding the code phase estimates comparison. The recommended RECS lengths are highlighted in red, according to the obtained standard deviation of the range error, considering a $3\sigma < 15$ meters (corresponding to 1 sample at the sampling rate used).

Estimated C/N$_0$	**RECS Length**	**Estimated std. deviation**
45 dB-Hz	$\sim$ 1 ms	$\sigma = 5.0$ m
45 dB-Hz	$\sim$ 2 ms	$\sigma = 3.8$ m
40 dB-Hz	$\sim$ 2 ms	$\sigma = 8.7$ m
40 dB-Hz	$\sim$ 4 ms	$\sigma = 3.6$ m
35 dB-Hz	$\sim$ 8 ms	$\sigma = 5.5$ m
35 dB-Hz	$\sim$ 16 ms	$\sigma = 4.7$ m

Table 6.10: Summary of results of E1-E6 code phase delay/range error comparison.

6.4.2 Signal Detection (ROCs)

To assess the performance of the acquisition for SAS we compute the ROC curves, which compare the probability of detection, denoted P_{D}, against the probability of false alarm, denoted P_{FA}, for a given C/N_0. This probability of detection depends mainly on the C/N_0 of the received signal and the length of the local replica used which, for SAS, is determined by the number of chips used for the RECS. The signal detection principles in GNSS and the ROC curves are described in Section 2.2.5 and Section 2.2.6, respectively.

To compute the ROC curves, each recorded snapshot is divided in chunks of 16 milliseconds. For each chunk, the MATLAB simulator obtains the acquisition metric by taking the maximum value from the squared CAF. Finally, the simulator plots the corresponding curve for the selected satellite. To emulate the alternative hypothesis (i.e., the absence of signal), the signal read from the snapshot is multiplied by a random binary sequence that does not correspond with the actual PRN sequence of the satellite analyzed.

The results are shown in Figure 6.20, where different configurations have been used, using the previously selected satellites of the real datasets C0 and D0 (see Table 6.9). The curves correspond to the results obtained with SAS nominal procedure, i.e., reducing the acquisition search space to just a few samples (20 in our case), thanks to the estimates obtained from the E1-B signal.

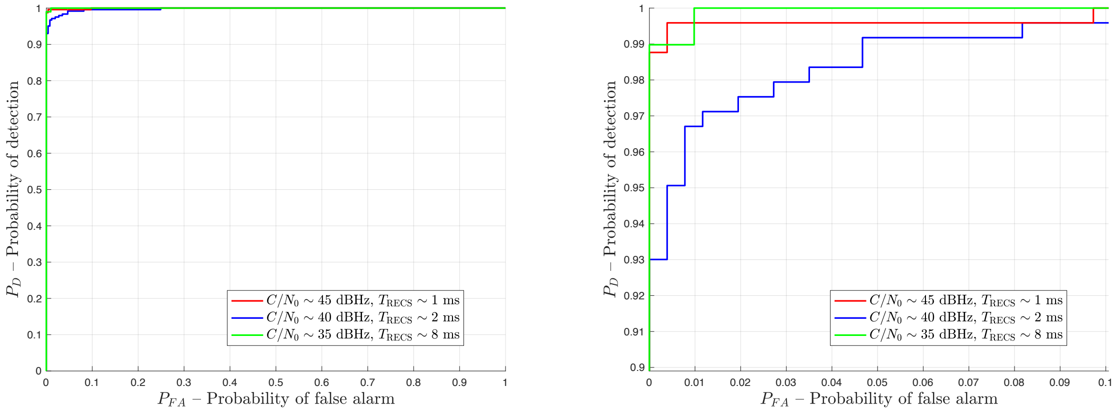


Figure 6.20: ROC curve for a selection of satellites visible in real datasets selected (on the right, zoomed version for $P_D \geq 0.9$).

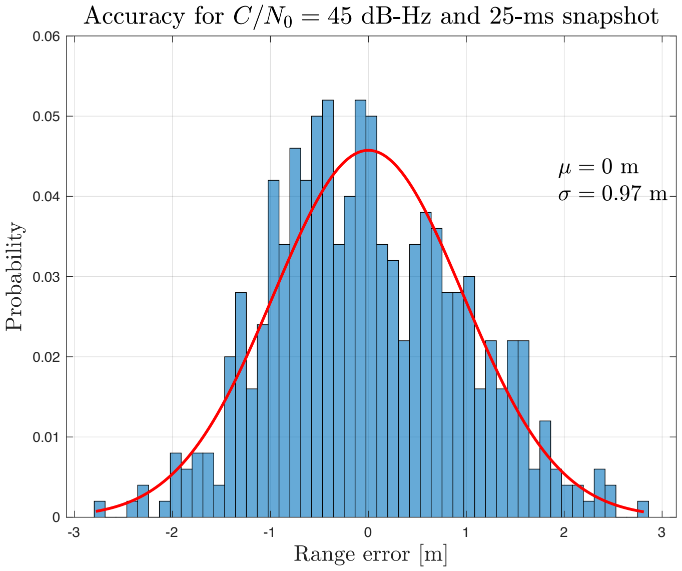
As we can observe, just a 1-ms RECS for $C/N_0 = 45$ dB-Hz, 2-ms RECS for $C/N_0 = 40$ dB-Hz and 8-ms RECS for $C/N_0 = 35$ dBHz will suffice to obtain a $P_D > 0.95$ for a $P_{\text{FA}} \sim 10^{-2}$. For lower probabilities of false alarm, doubling at least the size of the RECS will be required ($\sim$ 2 ms for $C/N_0 = 45$ dB-Hz, $\sim$ 4 ms for $C/N_0 = 40$ dB-Hz and 16 ms for $C/N_0 = 35$ dB-Hz), which matches the recommendations stated in Table 6.10.

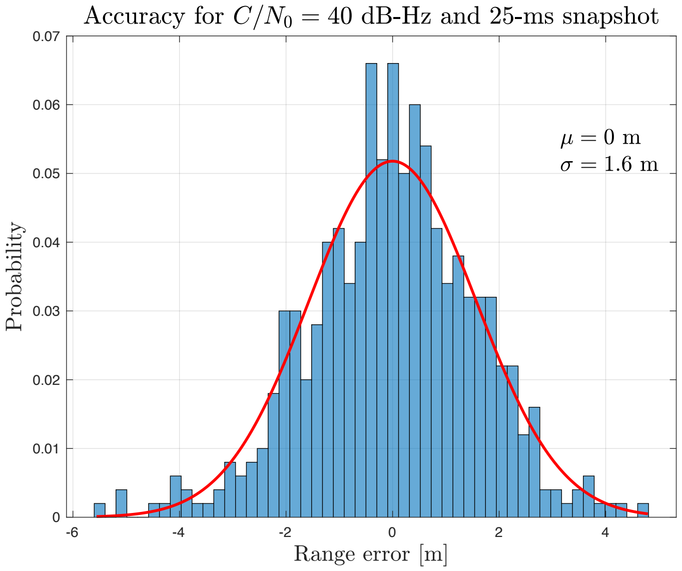
6.4.3 E6-B Snapshot Code Delay Estimation

The subsequent results focus on estimating the ranging accuracy from processing snapshots of E6-B samples. The dataset used (D0) was recorded under clear-sky conditions with the antenna covered by concrete blocks to simulate different C/N_0 levels. As in Section 6.4.1, the snapshots are processed with the MATLAB simulator, but, in this case, they are divided into smaller segments of the required sizes (see Table 5.7).

For each segmented snapshot, the code delay is estimated. As expected, owing to the Doppler frequency effect, the code phase delay estimates exhibit a linear trend over time. This trend line is subtracted from the estimates to obtain an estimation of the ranging accuracy by estimating the standard deviation of the interpolated Gaussian distribution.

As shown in Figure 6.21, the standard deviations estimated for ranging accuracy are consistent with the CRLBs in Table 2. The slight discrepancies may be attributed to factors such as non-ideal filters in the SDRs or the method used to estimate ranging accuracy, which involved removing the trend line from the code delay estimates. This method may not perfectly capture the difference between the estimated and true measurements.





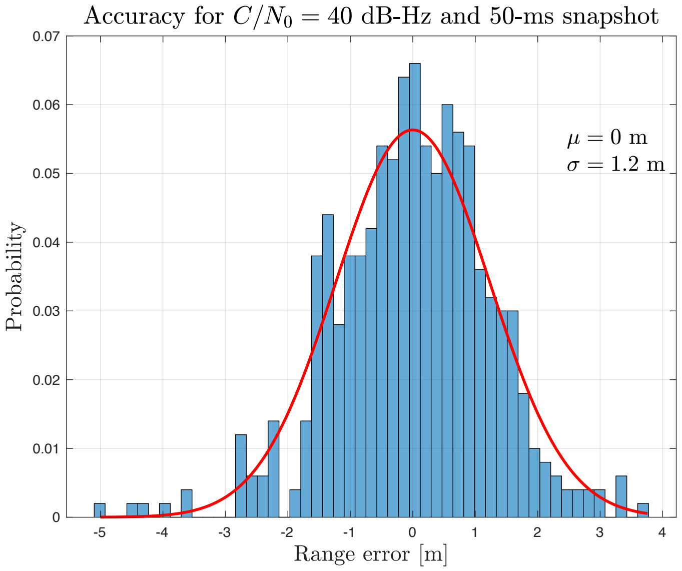


Figure 6.21: Code phase delay estimates histograms, after removing the trend line, using E6-B snapshots of different lengths and C/N_0's.

6.5 Conclusions

In this chapter, we have provided a preliminary evaluation of the signal-level performance of SAS. This has been facilitated by implementing a low-cost experimental platform based on bladeRF SDR boards. This platform comprises two bladeRF micro boards capable of capturing 16-bit samples up to 60 Msps at the E1 and E6 frequency bands, meeting the requirements for SAS. As the alignment of the estimates obtained from both bands is crucial for the nominal operation of SAS, the boards needed to be synchronized. Despite the absence of an official synchronization feature, we have successfully performed synchronous recording of E1 and E6 samples by connecting both boards with specific test points and providing a triggering mechanism inferred from previous bladeRF versions. The entire configuration procedure has been described in detail, enabling easy reproduction by other researchers.

Several snapshots of E1 and E6 samples have been captured with the evaluation platform under different C/N_0 conditions. This has allowed us to compare the code phase delay and Doppler frequency estimates of both bands, corroborating the assumptions about the alignment of such estimates. The real datasets used have also been made available online.

To accomplish this, the MATLAB simulator implemented to evaluate SAS with synthetic data has been extended to work with real signals. However, as the E6-C signal is still broadcast unencrypted, the RECSs have been emulated. To do so, the simulator performs a coherent integration of the 1-ms E6-C primary codes required to match the emulated RECS length, masking the rest of the signal with a random key.

Each of the snapshots recorded with the platform has been divided into smaller chunks. The simulator is then used to estimate both the code phase and Doppler frequency for each chunk. From these estimates, a Gaussian-like distribution is inferred, from which an estimated standard deviation for the range error is derived. The results show that a RECS length of 16 ms seems to be a good compromise for the C/N_0 values analyzed.

Finally, estimations of the accuracies for the code phase delay using a snapshot approach based on the E6-B signal are provided, which corroborate the feasibility of this approach.

Chapter 7

Conclusions and Future Research

This thesis has contributed to the topic of GNSS authentication by analyzing the new Galileo service named SAS, originally referred to as ACAS. This service is part of the strategy of GNSS providers to offer secure and protected satellite navigation against malicious attacks, without relying exclusively on the defenses implemented by the receivers themselves. SAS provides authentication at range level by using spreading code encryption, and aims to establish an additional barrier against spoofing attacks. It is designed to work in conjunction with OSNMA, which already offers authentication at the data level in the navigation message.

This new service is currently being defined and implemented by the Galileo Program through various projects led by the EC. The author and supervisors of this thesis have been involved in these projects from the outset, enabling them to contribute to the definition of key parameters that characterize SAS. The analysis conducted in this thesis focuses on understanding the underlying principles of this service and highlighting the differences compared to conventional GNSS services. The specific requirements of SAS call for novel strategies to ensure its effective implementation for future compatible receivers.

The contribution of this thesis is twofold. First, an analysis of the characteristics and parameters of the service has been performed, leading to several approaches for its implementation. This analysis has been complemented with synthetic-data simulations to determine the impact of the key SAS parameters in the service performance at signal level. Second, an evaluation using real data samples has been conducted to test this new service under real-world scenarios, in order to validate its feasibility. For this purpose, an evaluation platform based on SDR has been developed, which enables the synchronous capture of both Galileo E1 and E6 bands, as required by the default operating mode of SAS.

The conclusions of the thesis are presented next, along with potential future research directions.

7.1 Conclusions

Below, we summarize the contributions and conclusions for each of the topics addressed in this thesis.

Chapter 2 aimed to provide the necessary context for this work, which focuses on GNSS authentication. First, the fundamentals of GNSS were introduced, covering the key concepts of satellite navigation, including the architecture, signals used, and the implementation of GNSS receivers. Special emphasis was placed on the acquisition process and signal detection, which are critical for analyzing Galileo SAS. Second, a brief background on authentication principles and mechanisms in GNSS was provided. Understanding GNSS vulnerabilities is essential for analyzing the protection mechanisms used to mitigate these threats, revealing the challenges a receiver may face. Finally, recent anti-spoofing techniques were also reviewed.

In Chapter 3, the SAS is described in detail, based on both the original specifications and the latest available at the time of writing this thesis. The underlying concept of the service and its key parameters—currently under definition—are reviewed, highlighting the primary aspects to be considered. Additionally, OSNMA is briefly introduced, as SAS relies in the keys used in the former service. Finally, the cryptographic operations involved in SAS, although outside the scope of this thesis, are also outlined.

Chapter 4 aimed to provide guidelines for implementing SAS, with a focus on signal detection and authentication algorithms. A generic approach was presented, where the receiver relies solely on the E6-C signal for acquiring RECSs and subsequent authentication. In this approach, the receiver's time reference is shown to be critical for acquisition performance at the signal level. Simulations demonstrate the significant impact of uncertainties surrounding this reference on performance. Additionally, the influence of key parameters such as RECS Length and Period reveals important trade-offs to consider. Finally, the threats a receiver may face are analyzed, emphasizing the importance of VSS algorithms.

In Chapter 5, the so-called Nominal Operating Mode for SAS has been presented. In this mode, the receiver leverages the E1-B signal, which is already used to obtain the required TESLA keys from OSNMA. This enables the receiver to obtain an accurate time reference independent of its own clock, thereby narrowing the search space in the acquisition process. Simulations corroborate the benefits of this approach and reveal the minimum parameter requirements for the scenarios in which SAS is intended to operate. Additionally, different operating modes are compared based on how the time reference is obtained. The threats faced by a receiver operating in this mode are also examined: the inclusion of the E1-B signal can be utilized to accelerate VSS algorithms. Finally, to further enhance SAS performance for both signal detection and authentication, an alternative using E6-B samples is proposed, demonstrating the potential benefits of employing auxiliary signals.

In Chapter 6, an evaluation of the performance of SAS using real data samples is presented. For this purpose, an experimental platform based on SDR was developed. This platform enables the synchronous capture of E1-B and E6-C samples, which proved crucial for comparing the code phase delay and Doppler frequency estimates of both bands. The test datasets used in the evaluation are also described. The results obtained verified the alignment of these estimates and validated the feasibility of SAS at the signal level. The ROCs curves provided further helped determining the recommended values for the RECS and corroborated the previous results. Finally, the results derived from the E6-B processing confirmed the advantages of snapshot-based approaches for SAS.

7.2 Future Research

Next, the open issues identified for future work are outlined.

- The simulations carried out with synthetic data in Chapters 4 and 5 could be refined by incorporating more realistic scenarios to bridge the gap with the results obtained from real data samples in Chapter 6. This would allow for a more direct comparison between the simulated and real-world results.

- The authentication mechanisms presented in Chapter 5 are based on the difference in ranging measurements and their comparison to a predefined threshold. Quantifying the natural errors that contribute to this range difference, including all inter-frequency biases between E1 and E6, is crucial for accurately modeling these contributions and, consequently, for establishing a reliable mechanism for authenticating the measurements. Additionally, fine-tuning these thresholds could further enhance the precision and effectiveness of the authentication process.

- In addition to the primary algorithms implemented for signal detection and authentication, auxiliary algorithms can be considered to enhance resilience against spoofing attacks. Specifically, the analysis of VSS algorithms could be extended by incorporating the use of additional auxiliary signals, such as E6-B, to further strengthen the detection and authentication process.
- As depicted in Chapter 5, enhancing the resilience of an SAS receiver involves introducing additional checks to detect a broader range of spoofing attacks. In addition to the VSS algorithms described earlier, which focus on searching in the time domain, it would be also feasible to perform a search in the frequency domain. For instance, if the signal is not detected in the E6-C after transitioning from E1-B (which involves testing some samples in the time domain but only within a single Doppler frequency bin), the receiver could initiate a frequency search across the remaining Doppler bins that were initially discarded.

Bibliography

[1] I. Fernandez-Hernandez, J. Winkel, C. O'Driscoll, S. Cancela, R. Terris-Gallego, *et al.*, "Semiassisted Signal Authentication for Galileo: Proof of Concept and Results," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 4393-4404, Aug. 2023.

[2] I. Fernandez-Hernandez, J. Winkel, C. O'Driscoll, G. Caparra, R. Terris-Gallego, *et al.*, "Galileo Signal Authentication Service (SAS)," in *Proceedings of the 37th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2024)*, Baltimore, Maryland, Sep. 2024, pp. 3292 –3307. [Online]. Available: `https://www.ion.org/publications/abstract.cfm?articleID=19707`.

[3] R. Terris-Gallego, I. Fernandez-Hernandez, J. A. López-Salcedo, and G. Seco-Granados, "Guidelines for Galileo Assisted Commercial Authentication Service Implementation," in *Proceedings of the International Conference on Localization and GNSS (ICL GNSS)*, Tampere, Jun. 2022.

[4] R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, and I. Fernandez-Hernandez, "Operating Modes and Performance Evaluation of Galileo Assisted Commercial Authentication Service," in *Institute of Navigation Conference (ION+ GNSS 2022)*, 2022.

[5] R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, and I. Fernandez-Hernandez, "E1-E6 SDR platform based on bladeRF for testing Galileo Assisted Commercial Authentication Service," in *European Navigation Conference, ESA ESTEC*, Noordwijk, The Netherlands: MDPI, 2023.

[6] R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, and I. Fernandez-Hernandez, "Preliminary Evaluation of Galileo ACAS using Existing E1-E6 Open Signals and a Low-Cost SDR Platform," in *Institute of Navigation Conference (ION+ GNSS 2023)*, Denver, Sep. 2023.

[7] R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, and I. Fernandez-Hernandez, "Efficient Detection of Galileo ACAS Sequences using E6-B Aiding," in *European Navigation Conference, ESA ESTEC*, Noordwijk, The Netherlands: MDPI, 2024.

[8] European Commission, *Call for Tenders (DEFIS/2020/OP/0002) – Test Platform on Galileo HAS/CAS/OSNMA*, 2020. [Online]. Available: `https://etendering.ted.europa.eu/cft/cft-display.html?cftId=6271`.

[9] UAB, “PAULA Research Report: Detailed Analysis On Assisted Commercial Authentication Service,” Tech. Rep., 2023.

[10] GMV, “PAULA Final Report,” Tech. Rep., 2023. [Online]. Available: `https://etendering.ted.europa.eu/cft/cft-document.html?docId=155981`.

[11] *Call for Tenders (DEFIS/2023/OP/0011) – Contribution to Radio-navigation Accuracy and Resilience*, 2023. [Online]. Available: `https://etendering.ted.europa.eu/cft/cft-display.html?cftId=14288`.

[12] J. Karl, *Celestial Navigation in the GPS Age.* 2007. [Online]. Available: `www.paracay.com`.

[13] EUSPA, “EO and GNSS Market Report. 2022,” EUSPA, Tech. Rep., 2022. [Online]. Available: `https://www.euspa.europa.eu/sites/default/files/uploads/euspa_market_report_2022.pdf`.

[14] H. E. Worth and M. Warren, *Transit to Tomorrow – Fifty Years of Space Research at The Johns Hopkins University Applied Physics Laboratory.* JHU/APL, 2009, ISBN: 0-978-0-615-33024-2.

[15] V. Lucas-Sabola, G. Seco-Granados, J. A. López-Salcedo, J. A. García-Molina, and M. Crisci, “Cloud GNSS receivers: New advanced applications made possible,” *2016 International Conference on Localization and GNSS (ICL-GNSS)*, pp. 1–6, 2016. [Online]. Available: `https://api.semanticscholar.org/CorpusID:32179678`.

[16] ESA, *Navipedia.* [Online]. Available: `https://gssc.esa.int/navipedia/index.php/Main_Page`.

[17] P. J. Teunissen and O. Montenbruck, *Handbook of Global Navigation Satellite Systems.* Springer, 2017.

[18] J. B.-Y. Tsui, *Fundamentals of Global Positioning System Receivers - A Software Approach*, 2nd ed. 2005, ISBN: 3175723993.

[19] E. Kaplan and C. Hegarty, *Understanding GPS - Principles and Applications*, 2nd ed. 2006.

[20] F. van Diggelen, “A-GPS: Assisted GPS, GNSS, and SBAS,” Tech. Rep.

[21] B. W. Parkinson, P. Enge, P. Axelrad, and J. J. Spilker, *Global Positioning System: Theory and Applications.* Progress in Astronautics and Aeronautics, 1996.

[22] G. Seco-Granados, J. A. López-Salcedo, D. Jiménez-Baños, and G. López-Risueno, “Challenges in indoor global navigation satellite systems: Unveiling its core features in signal processing,” *IEEE Signal Processing Magazine*, vol. 29, no. 2, pp. 108–131, 2012, ISSN: 10535888. DOI: `10.1109/MSP.2011.943410`.

[23] US Government, *GPS Space Segment.* [Online]. Available: `https://www.gps.gov/systems/gps/space`.

[24] K. Borre, I. Fernandez-Hernandez, J. A. López-Salcedo, H. Bhuiyan, and M. Zahidul, *GNSS Software Receivers.* Cambridge University Press, 2023. DOI: `10.1017/9781108934176`.

[25] C. Gentile, N. Alsindi, R. Raulefs, and C. Teolis, *Geolocation techniques: Principles and applications.* Springer New York, 2013, pp. 1–287, ISBN: 9781461418368. DOI: 10.1007/978-1-4614-1836-8.

[26] S. A. Zekavat and R. M. Buehrer, *Handbook of position location: theory, practice, and advances.* 2019. [Online]. Available: https://cataleg.uab.cat/iii/encore/record/C__Rb2083164?lang=cat.

[27] M. Petovello and C. O'Driscoll, "Carrier phase and its measurement for GNSS," *Inside GNSS*, Jul. 2010. [Online]. Available: https://www.insidegnss.com/auto/julaug10-solutions.pdf.

[28] M. Foucras, "Performance Analysis of Modernized GNSS Signal Acquisition," PhD thesis, INP Toulouse, Jun. 2015. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01169567.

[29] D. Gómez Casco, "Non-Coherent Acquisition Techniques for High-Sensitivity GNSS Receivers," PhD thesis, Universitat Autònoma de Barcelona (UAB), Barcelona, Sep. 2018. [Online]. Available: https://ddd.uab.cat/record/204075.

[30] S. Locubiche-Serra, G. Seco-Granados, and J. A. Lopez-Salcedo, "Doubly-adaptive autoregressive Kalman filter for GNSS carrier tracking under scintillation conditions," in *2016 International Conference on Localization and GNSS (ICL-GNSS)*, IEEE, Jun. 2016, pp. 1–6, ISBN: 978-1-5090-1757-7. DOI: 10.1109/ICL-GNSS.2016.7533859.

[31] S. Bancroft, "An Algebraic Solution of the GPS Equations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-21, no. 1, pp. 56–59, Jan. 1985, ISSN: 0018-9251. DOI: 10.1109/TAES.1985.310538. [Online]. Available: https://ieeexplore-ieee-org.are.uab.cat/document/4104017.

[32] D. Borio, C. O'Driscoll, and G. Lachapelle, "Coherent, noncoherent, and differentially coherent combining techniques for acquisition of new composite GNSS signals," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 3, pp. 1227–1240, Jul. 2009, ISSN: 00189251. DOI: 10.1109/TAES.2009.5259196.

[33] D. Borio, "A Statistical Theory for GNSS Signal Acquisition," PhD thesis, 2008.

[34] J. P. Egan, *Signal Detection Theory and ROC analysis.* Academic Press, 1975.

[35] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Researchers," 2004.

[36] D. W. Allan, "Statistics of Atomic Frequency Standards," in *Proceedings of the IEEE*, vol. 54, 1966.

[37] S. R. Stein, "Frequency and Time-Their Measurement and Characterization," in *Precision Frequency Control*, E. Gerber and A. Ballato, Eds., vol. 2, New York: Academic Press (Time and Frequency Division, National Bureau of Standards), 1985, ch. 12, pp. 191–416. [Online]. Available: https://tf.nist.gov/general/pdf/666.pdf.

[38] T. Bruggemann, D. Greer, and R. Walker, "Chip Scale Atomic Clocks: Benefits to Airborne GNSS Navigation Performance," Proceedings of the International Global Navigation Satellite Systems Society IGNSS Symposium 2006, 2006. [Online]. Available: `http://eprints.qut.edu.au`.

[39] R. G. Brown and P. Y. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 4th. John Wiley & Sons, 1992.

[40] J. A. Volpe, "Vulnerability Assessment of the U.S. Transportation Infrastructure that Relies on the Global Positioning System," U. S. Department of Transportation, Tech. Rep., Aug. 2001. DOI: `10.1017/S0373463303002273`.

[41] O Pozzobon, L Canzian, M Danieletto, and A. D. Chiara, "Anti-spoofing and open GNSS signal authentication with signal authentication sequences," in *5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, 2010, pp. 1–6. DOI: `10.1109/NAVITEC.2010.5708065`.

[42] P. F. MacDoran, M. B. Mathews, F. A. Ziel, K. L. Gold, S. M. Anderson, *et al.*, *Method and apparatus for authenticating the location of remote users of networked computing systems*, May 1998.

[43] L. Scott, "Anti-spoofing & authenticated signal architectures for civil navigation systems," in *Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS)*, 2003, pp. 1543–1552.

[44] C. Wullems, O. Pozzobon, and K. Kubik, "Signal authentication and integrity schemes for next generation global navigation satellite systems," in *European Navigation Conference (ENC-GNSS 2005)*, 2005.

[45] M. L. Psiaki and T. E. Humphreys, "GNSS Spoofing and Detection," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1258–1270, 2016.

[46] I Fernandez-Hernandez, G Vecchione, and F Díaz-Pulido, "Galileo authentication: A programme and policy perspective," in *69th International Astronautical Congress*, 2018.

[47] I. Fernandez-Hernandez, V. Rijmen, G. Seco-Granados, J. Simon, I. Rodríguez, *et al.*, "A navigation message authentication proposal for the Galileo open service," *NAVIGATION, Journal of the Institute of Navigation*, vol. 63, no. 1, pp. 85–102, 2016.

[48] European Union Agency for the Space Programme (EUSPA), *OSNMA Public Observation Test Phase*, 2023. [Online]. Available: `https://www.gsc-europa.eu/support-to-developers/osnma-public-observation-test-phase`.

[49] P. Gutierrez, *Galileo Authentication and High-Accuracy Service: Coming on Fast*, Jul. 2021. [Online]. Available: `https://insidegnss.com/galileo-authentication-and-high-accuracy-service-coming-on-fast/`.

[50] J. Qiao, Z. Lu, B. Lin, J. Song, Z. Xiao, *et al.*, "A survey of GNSS interference monitoring technologies," *Frontiers in Physics*, vol. 11, Mar. 2023, ISSN: 2296-424X. DOI: `10.3389/fphy.2023.1133316`.

[51] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned Aircraft Capture and Control via GPS Spoofing," *Journal of Field Robotics*, Jul. 2014. DOI: `10.1002/rob.21513`.

[52] J.-S.H.C.o.S.v.F.G.S. D. Bhatti, Detection, and T. E. Humphreys, "Hostile Control of Ships via False GPS Signals: Demonstration and Detection," *Navigation, Journal of the Institute of Navigation*, vol. 64, no. 1, pp. 51–66, Mar. 2017, ISSN: 00281522. DOI: `10.1002/navi.183`.

[53] Omar ElKaraksy, *Understanding GNSS Signal Vulnerabilities: Exploring Jamming vs. Spoofing*, Apr. 2024. [Online]. Available: `https://www.rokubun.cat/gnss-jamming-vs-spoofing-associated-risks-and-solutions/`.

[54] Septentrio, "GNSS Interference Brochure," Tech. Rep., 2020. [Online]. Available: `https://septentrio.sharepoint.com/Marketing4Sales/Shared%20Documents/Forms/AllItems.aspx?id=%2FMarketing4Sales%2FShared%20Documents%2FBrochures%5FLeaflets%2FInterference%2FSeptentrio%5FGNSS%5FInterference%5FA5%5FLR%2Epdf&parent=%2FMarketing4Sales%2FShared%20Documents%2FBrochures%5FLeaflets%2FInterference&p=true&ga=1`.

[55] M Wildemeersch, E Cano Pons, A Rabbachin, and J Fortuny Guasch, "Impact Study of Unintentional Interference on GNSS Receivers," EC Joint Research Centre Security Technology Assessment Unit, Tech. Rep., 2010. DOI: `10.2788/57794`. [Online]. Available: `http://www.jrc.ec.europa.eu`.

[56] Spirent, "GNSS Jamming: How to test the risks to safety-critical and liability-critical systems (White Paper)," Tech. Rep., Nov. 2022.

[57] DOD/NATO, *U.S. Military definition of Meaconing.* [Online]. Available: `https://web.archive.org/web/20100529174246/http://www.dtic.mil/doctrine/jel/doddict/data/m/03301.html`.

[58] Septentrio, "GNSS Spoofing Brochure," Tech. Rep., 2024. [Online]. Available: `https://septentrio.sharepoint.com/Marketing4Sales/Shared%20Documents/Forms/AllItems.aspx?id=%2FMarketing4Sales%2FShared%20Documents%2FBrochures%5FLeaflets%2FSpoofing%2FSeptentrio%5FSpoofing%5Fbrochure%5FA5%5FLR%2Epdf&parent=%2FMarketing4Sales%2FShared%20Documents%2FBrochures%5FLeaflets%2FSpoofing&p=true&ga=1`.

[59] T. E. Humphreys, M. L. Psiaki, B. W. O'hanlon, and P. M. Kintner, "Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer," in *ION GNSS 2008*, Savanna, GA, Sep. 2008. [Online]. Available: `http://philosecurity.org/2008/09/07/gps-spoofing`.

[60] T. E. Humphreys, "Detection Strategy for Cryptographic GNSS Anti-Spoofing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1073–1090, 2013, ISSN: 00189251. DOI: `10.1109/TAES.2013.6494400`.

[61] O. Pozzobon, C. Wullems, and M. Detratti, “Security Considerations in the design of tamper resistant GNSS receivers,” in *5th ESA Workshop on Satellite Navigation Technologies (NAVITEC 2011)*, ESTEC, Noordwijk, The Netherlands, Apr. 2011, ISBN: 9781424487394.

[62] Z. Liu, S. Lo, J. Blanch, Y.-H. Chen, T. Walter, *et al.*, “GNSS Spoofing and Jamming in Eastern Europe,” *Inside GNSS*, Mar. 2024. [Online]. Available: `https://insidegnss.com/gnss-spoofing-and-jamming-in-eastern-europe/`.

[63] Intertanko, “Jamming and Spoofing of Global Navigation Satellite Systems (GNSS),” Tech. Rep., 2019. [Online]. Available: `https://www.maritimeglobalsecurity.org/media/1043/2019-jamming-spoofing-of-gnss.pdf`.

[64] Novatel-Hexagon, “Spoofed or Jammed? Busting the myths of GNSS interference and mitigation (White Paper),” Tech. Rep., Jan. 2024. [Online]. Available: `https://novatel.com/tech-talk/papers/gnss-jamming-and-spoofing`.

[65] GPS Patron, *GNSS Spoofing Scenarios with SDRs*, Sep. 2021. [Online]. Available: `https://gpspatron.com/gnss-spoofing-scenarios-with-sdrs/`.

[66] T. Nguyen-Tan, L. T. Hoang, A. K. Nguyen, T. D. Minh, B. Bui-Thanh, *et al.*, “GPS Signal Reception and Spoofing Based on Software-Defined Radio Devices,” in *Proceedings - 2022 RIVF International Conference on Computing and Communication Technologies, RIVF 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 513–517, ISBN: 9781665461665. DOI: `10.1109/RIVF55975.2022.10013839`.

[67] R. Ferreira, J. Gaspar, N. Souto, and P. Sebastião, “Effective GPS Jamming Techniques for UAVs using low-cost SDR platforms,” in *The 6th Global Wireless Summit (GWS-2018)*, Nov. 2018, ISBN: 9781538642887.

[68] M. Yuan, X. Tang, and G. Ou, “Authenticating GNSS civilian signals: a survey,” *Springer Satellite Navigation*, vol. 4, no. 1, Dec. 2023, ISSN: 26621363. DOI: `10.1186/s43020-023-00094-6`.

[69] X. Chen, R. Luo, T. Liu, H. Yuan, and H. Wu, “Satellite Navigation Signal Authentication in GNSS: A Survey on Technology Evolution, Status, and Perspective for BDS,” *Remote Sensing*, vol. 15, no. 5, Mar. 2023, ISSN: 20724292. DOI: `10.3390/rs15051462`.

[70] D. M. Akos and M. Pini, “Effect of sampling frequency on GNSS receiver performance,” *Navigation, Journal of the Institute of Navigation*, vol. 53, no. 2, pp. 85–95, 2006, ISSN: 00281522. DOI: `10.1002/j.2161-4296.2006.tb00375.x`.

[71] J. Blanch and T. Walter, “RAIM with Optimal Integrity and Continuity Allocations Under Multiple Failures,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 3, Jul. 2010.

[72] J. Li, X. Zhu, M. Ouyang, W. Li, Z. Chen, *et al.*, “Research on Multi-Peak Detection of Small Delay Spoofing Signal,” *IEEE Access*, vol. 8, pp. 151 777–151 787, 2020, ISSN: 2169-3536. DOI: `10.1109/ACCESS.2020.3016971`.

[73] J. M. Parro, J. A. Lopez-Salcedo, R. Ioannides, and M. Crisci, “Signal-Level Integrity Monitoring Metric for Robust GNSS receivers,” in *31st AIAA International Communications Satellite Systems Conference*, Reston, Virginia: American Institute of Aeronautics and Astronautics, Oct. 2013, ISBN: 978-1-62410-244-8. DOI: 10.2514/6.2013-5613.

[74] W. Wang, N. Li, R. Wu, and P. Closas, “Detection of Induced GNSS Spoofing Using S-Curve-Bias,” *Sensors*, vol. 19, no. 4, p. 922, Feb. 2019, ISSN: 1424-8220. DOI: 10.3390/s19040922.

[75] M. L. Psiaki, B. W. O’hanlon, S. P. Powell, J. A. Bhatti, K. D. Wesson, *et al.*, “GNSS Spoofing Detection using Two-Antenna Differential Carrier Phase,” in *27th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2014)*, Tampa, Florida, Sep. 2014, pp. 2776–2800.

[76] D. Borio and C. Gioia, “A sum-of-squares approach to GNSS spoofing detection,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1756–1768, Aug. 2016, ISSN: 00189251. DOI: 10.1109/TAES.2016.150148.

[77] W. De Wilde, J.-M. Sleewaegen, B. Bougard, G. Cuypers, S. A. Popugaev, *et al.*, “Authentication by polarization: a powerful anti-spoofing method,” in *31st International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2018)*, Miami, Florida, Sep. 2018.

[78] P. F. Swaszek, “GNSS Spoof Detection Using Shipboard IMU Measurements,” in *the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014),*, Tampa, Florida, Sep. 2014, pp. 745–758. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=12372.

[79] C. Tanil, S. Khanafseh, M. Joerger, and B. Pervan, “An INS Monitor to Detect GNSS Spoofers Capable of Tracking Vehicle Position,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 1, pp. 131–143, Feb. 2018, ISSN: 0018-9251. DOI: 10.1109/TAES.2017.2739924.

[80] S. Khanafseh, N. Roshan, S. Langel, F.-C. Chan, M. Joerger, *et al.*, “GPS Spoofing Detection using RAIM with INS Coupling,” in *IEEE/ION Position, Location and Navigation Symposium (PLANS 2014)*, Monterey, CA, USA, May 2014, pp. 1232–1239. DOI: 10.1109/PLANS.2014.6851498. [Online]. Available: https://ieeexplore.ieee.org/document/6851498.

[81] M. L. Psiaki, S. P. Powell, and B. W. O’hanlon, “GNSS Spoofing Detection Using High-Frequency Antenna Motion and Carrier-Phase Data,” in *26th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2013)*, Nashville, TN, USA, Sep. 2013, pp. 2949–2991.

[82] L. He, H. Li, and M. Lu, “Dual-antenna GNSS spoofing detection method based on Doppler frequency difference of arrival,” *GPS Solutions*, vol. 23, no. 3, p. 78, Jul. 2019, ISSN: 1080-5370. DOI: 10.1007/s10291-019-0868-5.

[83] J. Li, X. Zhu, M. Ouyang, D. Shen, Z. Chen, *et al.*, "GNSS spoofing detection technology based on Doppler frequency shift difference correlation," *Measurement Science and Technology*, vol. 33, no. 9, p. 95 109, Jun. 2022. DOI: 10.1088/1361-6501/ac672a. [Online]. Available: https://dx.doi.org/10.1088/1361-6501/ac672a.

[84] L. Meng, L. Yang, W. Yang, and L. Zhang, "A Survey of GNSS Spoofing and Anti-Spoofing Technology," *Remote Sensing*, vol. 14, no. 19, Oct. 2022, ISSN: 20724292. DOI: 10.3390/rs14194826.

[85] A. Broumandan, R. Siddakatte, and G. Lachapelle, "An approach to detect GNSS spoofing," *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, no. 8, pp. 64–75, Aug. 2017, ISSN: 0885-8985. DOI: 10.1109/MAES.2017.160190.

[86] C. Günther, "A Survey of Spoofing and Counter-Measures," *Navigation*, vol. 61, no. 3, pp. 159–177, Sep. 2014, ISSN: 00281522. DOI: 10.1002/navi.65. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=102624.

[87] M. Psiaki and T. Humphreys, "Civilian GNSS Spoofing, Detection, and Recovery," in *Position, Navigation, and Timing Technologies in the 21st Century*, Wiley, Dec. 2021, pp. 655–680. DOI: 10.1002/9781119458449.ch25.

[88] A. J. Kerns, K. D. Wesson, and T. E. Humphreys, "A Blueprint for Civil GPS Navigation Message Authentication," in *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, IEEE, May 2014, pp. 262–269, ISBN: 978-1-4799-3320-4. DOI: 10.1109/PLANS.2014.6851385.

[89] A. Perrig, D Song, R Canetti, I. J. D. Tygar, and B Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA) – Multicast Source Authentication Transform Introduction [RFC 4082]," Network Working Group, The Internet Society, Tech. Rep., Jun. 2005.

[90] G. Seco-Granados, D. Gómez-Casco, J. A. López-Salcedo, and I. Fernandez-Hernandez, "Detection of replay attacks to GNSS based on partial correlations and authentication data unpredictability," *Gps Solutions*, vol. 25, no. 2, pp. 1–15, 2021.

[91] K. D. Wesson, M. P. Rothlisberger, and T. E. Humphreys, "A Proposed Navigation Message Authentication Implementation for Civil GPS Anti-Spoofing," in *24th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2011)*, Portland, Oregon, USA, Sep. 2011, pp. 3129–3140. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=9870.

[92] M. Yuan, Z. Lv, H. Chen, J. Li, and G. Ou, "An Implementation of Navigation Message Authentication with Reserved Bits for Civil BDS Anti-Spoofing," in *China Satellite Navigation Conference (CSNC) 2017*, 2017, pp. 69–80. DOI: 10.1007/978-981-10-4591-2{_}6.

[93] O. Pozzobon, "Keeping the spoofs out – Signal Authentication Services for future GNSS," *Inside GNSS*, Jun. 2011. [Online]. Available: https://www.insidegnss.com/auto/mayjune11-Pozzobon.pdf.

[94] ESA Navipedia, *GPS Signal Plan.* [Online]. Available: https://gssc.esa.int/navipedia/index.php/GPS_Signal_Plan.

[95] L. Scott, "Proving Location Using GPS Location Signatures: Why it is Needed and a Way to Do It," in *Proceedings of the 26th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2013)*, Nashville, Tennessee, Sep. 2013, pp. 2880–2892. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=11180.

[96] J. M. Anderson, K. L. Carroll, N. P. DeVilbiss, J. T. Gillis, J. C. Hinks, *et al.*, "Chips-Message Robust Authentication (Chimera) for GPS Civilian Signals," in *Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017)*, Portland, Oregon, Sep. 2017, pp. 2388–2416. DOI: 10.33012/2017.15206. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=15206.

[97] B. Motella, M. Nicola, and S. Damy, "Enhanced GNSS Authentication Based on the Joint CHIMERA/OSNMA Scheme," *IEEE Access*, vol. 9, pp. 121 570–121 582, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3107871.

[98] L. Scott, *GPS & Galileo Civil Signal Authentication*, Dec. 2021. [Online]. Available: https://www.gps.gov/governance/advisory/meetings/2021-12/scott.pdf.

[99] J. Winkel, I. Fernandez-Hernandez, and C. O'Driscoll, "Implementation Considerations for ACAS and Simulation Results," *ArxiV Preprint*, Jul. 2023. [Online]. Available: http://arxiv.org/abs/2307.12398.

[100] D. S. Maier and T. Pany, "Assessment of a Low Power Offset BPSK Component for Spreading Code Authentication," *Journal of Positioning, Navigation, and Timing*, Apr. 2020. DOI: 10.11003/JPNT.2020.9.2.43. [Online]. Available: https://www.unibw.de/lrt9/lrt-9.2/preprints/2020-maier-assessment-of-a-low-power-offset-bpsk-component-for-spreading-code-authentication-1.pdf.

[101] I. Fernandez-Hernandez, S. Damy, M. Susi, I. Martini, J. O. Winkel, *et al.*, "Galileo Authentication and High Accuracy: Getting to the Truth," *Inside GNSS*, Feb. 2023. [Online]. Available: https://insidegnss.com/galileo-authentication-and-high-accuracy-getting-to-the-truth/.

[102] European Commission, *Call for Tenders (DEFIS/2020/OP/0002) – Test Platform on Galileo HAS/CAS/OSNMA – Tender Specifications*, 2020. [Online]. Available: https://etendering.ted.europa.eu/cft/cft-display.html?cftId=6271.

[103] European Commission, *Call for Tenders (DEFIS/2020/OP/0002) – Test Platform on Galileo HAS/CAS/OSNMA - Tender Specifications - Annex 7*, 2020. [Online]. Available: https://etendering.ted.europa.eu/cft/cft-display.html?cftId=6271.

[104] I. Fernandez-Hernandez, T. Walter, A. Neish, and C O'Driscoll, "Independent time synchronization for resilient gnss receivers," in *Proceedings of the 2020 International Technical Meeting of The Institute of Navigation*, 2020, pp. 964–978.

[105] European Commission, *European GNSS (Galileo) Open Service – Signal-in-Space Interface Document v2.0*, Jan. 2021. [Online]. Available: `https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf`.

[106] J. W. Betz, "The Offset Carrier Modulation for GPS Modernization," *Proceedings of The Institute of Navigation's National Technical Meeting*, Jan. 1999.

[107] J. W. Betz, "Binary offset carrier modulations for radionavigation," *Navigation, Journal of the Institute of Navigation*, vol. 48, no. 4, pp. 227–246, 2001, ISSN: 00281522. DOI: `10.1002/j.2161-4296.2001.tb00247.x`.

[108] D. Gómez-Casco, J. A. Garcia-Molina, A. Gusi-Amigó, M. Crisci, J. A. López-Salcedo, *et al.*, "Mitigation of false locks in the acquisition of high-order BOC signals in HS-GNSS receivers," in *2016 International Conference on Localization and GNSS (ICL-GNSS)*, 2016, pp. 1–6. DOI: `10.1109/ICL-GNSS.2016.7533691`. [Online]. Available: `https://ieeexplore.ieee.org/document/7533691`.

[109] E Göhler, I Krol, M Bodenbach, and J Winkel, "A Galileo E6-B/C Receiver: Signals, Prototype, Tests and Performance," in *Proceedings of the 29th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2016)*, Institute of Navigation (ION), Sep. 2016, pp. 486–496. [Online]. Available: `https://www.ion.org/publications/abstract.cfm?articleID=14828`.

[110] European Commission – DG DEFIS, "Galileo Assisted Commercial Authentication Service (ACAS) – Specification Proposal v1.2," Tech. Rep., 2023.

[111] H. V. De Castro, G Maarel, and E Safipour, "The possibility and added-value of authentication in future Galileo open signal," in *23rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2010)*, 2010, pp. 1112–1123. [Online]. Available: `https://www.ion.org/publications/abstract.cfm?articleID=9227`.

[112] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA Broadcast Authentication Protocol," *CryptoBites*, pp. 2–13, 2002.

[113] EUSPA, "Galileo Open Service Navigation Message Authentication (OSNMA) - Info Note," Tech. Rep., 2021. [Online]. Available: `https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OSNMA_Info_Note.pdf`.

[114] European Union, "Galileo Open Service Navigation Message Authentication (OSNMA) Signal-In-Space Interface Control Document (SIS ICD), Issue 1.1," Tech. Rep., Oct. 2023. DOI: `10.2878/594840`. [Online]. Available: `https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OSNMA_SIS_ICD_v1.1.pdf`.

[115] I. Fernández-Hernández and G. Seco-Granados, "Galileo NMA signal unpredictability and anti-replay protection," in *2016 International Conference on Localization and GNSS (ICL-GNSS)*, 2016, pp. 1–5. DOI: 10.1109/ICL-GNSS.2016.7533686. [Online]. Available: https://ieeexplore.ieee.org/document/7533686.

[116] I. Fernández-Hernández, T. Ashur, and V. Rijmen, "Analysis and Recommendations for MAC and Key Lengths in Delayed Disclosure GNSS Authentication Protocols," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 3, pp. 1827–1839, Jun. 2021, ISSN: 15579603. DOI: 10.1109/TAES.2021.3053129.

[117] S. Wallner, J. A. G. Molina, G. L. Risueno, J. Hahn, J. J. Floch, *et al.*, "Novel Concepts on GNSS Signal Design serving Emerging GNSS User Categories: Quasi-Pilot Signal," in *2020 European Navigation Conference (ENC)*, Dresden, Germany: IEEE, Nov. 2020, pp. 1–22, ISBN: 978-3-944976-27-3. DOI: 10.23919/ENC48637.2020.9317352. [Online]. Available: https://ieeexplore.ieee.org/document/9317352.

[118] J. Garcia-Molina, S. Wallner, F. Melman, C. V. Alocen, G. D. Broi, *et al.*, "Galileo Quasi-Pilot Signals: Assessment and Design Options for Acquisition and Time Dissemination," in *Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2023)*, Denver, Colorado, Oct. 2023, pp. 1363–1387. DOI: 10.33012/2023.19315. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=19315.

[119] Richard Johnson, "MATLAB Style Guidelines 2.0," MATLAB Central File Exchange, Tech. Rep., 2014. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/46056-matlab-style-guidelines-2-0.

[120] European Union, *Galileo E6-B/C Codes Technical Note*, Jan. 2019. [Online]. Available: https://www.gsc-europa.eu/sites/default/files/sites/all/files/E6BC_SIS_Technical_Note.pdf.

[121] G. López-Risueño and G. Seco-Granados, "Measurement and processing of indoor GPS signals using a one-shot software receiver," *Proc. ESA NAVITEC*, pp. 1–9, 2004.

[122] M. Pini, M. Fantino, A. Cavaleri, S. Ugazio, and L. Lo Presti, "Signal Quality Monitoring Applied to Spoofing Detection," in *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011*, Portland, OR, Sep. 2011. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=9738.

[123] T. Walter, J. Blanch, L. De Groot, L. N. Novatel, and M. Joerger, "Ionospheric Rates of Change," in *Proceedings of the 31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018)*, 2018.

[124] F. Ardizzon, G. Caparra, I. Fernandez-Hernandez, and C. O'Driscoll, "A Blueprint for Multi-Frequency and Multi-Constellation PVT Assurance," in *NAVITEC*, Apr. 2022.

[125] F. Ardizzon, L. Crosara, S. Tomasin, and N. Laurenti, "On Mixing Authenticated and Non-Authenticated Signals Against GNSS Spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 4480–4493, 2024, ISSN: 15566021. DOI: 10.1109/TIFS.2024.3381473. [Online]. Available: https://ieeexplore-ieee-org.are.uab.cat/document/10478578.

[126] F. Ardizzon, L. Crosara, N. Laurenti, S. Tomasin, and N. Montini, "Authenticated Timing Protocol Based on Galileo ACAS," *Sensors*, 2022. DOI: 10.3390/s22166298. [Online]. Available: https://www.mdpi.com/1424-8220/22/16/6298.

[127] J. Winkel, I. Fernandez-Hernandez, and C. O'Driscoll, "Combining Galileo's Assisted Commercial Authentication Service (ACAS) with Vestigial Signal Search for Good Protection," in *Proceedings of the 2024 International Technical Meeting of The Institute of Navigation*, Long Beach, California, Jan. 2024, pp. 1225–1234. DOI: 10.33012/2024.19514. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=19514.

[128] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory.* Prentice-Hall, Inc., 1993.

[129] N. B. Delgado, "Signal Processing Techniques in Modern Multi-Constellation GNSS Receivers," PhD thesis, Lisbon, Portugal, Jun. 2011. DOI: 10.13140/RG.2.1.2739.4165. [Online]. Available: https://www.researchgate.net/publication/303824577.

[130] I. Fernandez Hernandez, "Snapshot and Authentication Techniques for Satellite Navigation," PhD thesis, Faculty of Engineering and Science, Aalborg University, Aalborg, Denmark, May 2015.

[131] I. Fernández-Hernández and K. Borre, *Snapshot positioning without initial information*, Oct. 2016. DOI: 10.1007/s10291-016-0530-4. [Online]. Available: https://link.springer.com/article/10.1007/s10291-016-0530-4.

[132] *LimeMicro Website – LimeSDR.* [Online]. Available: https://limemicro.com/products/boards/limesdr/.

[133] *Analog Devices Website – ADRV9364-Z7020.* [Online]. Available: https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/ADRV9364-Z7020.html.

[134] A. Csete and S. Christiansen, "Evaluation of SDR Boards and Toolchains – Final Report," Tech. Rep., 2020. [Online]. Available: https://gitlab.com/librespacefoundation/sdrmakerspace/sdreval/-/raw/master/Report/pdf/Evaluation_of_SDR_Boards-1.0.pdf.

[135] *SDR Makerspace Website.* [Online]. Available: https://sdrmaker.space.

[136] D. Akos, J. Arribas, F. Dovis, P. Di Torino, I. Fernandez-Hernandez, *et al.*, "GNSS Software Defined Radio: History, Current Developments, and Standardization Efforts," Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+), 2021.

[137] C. O'Driscoll and G. Caparra, "Nautilus: An embedded navigation authentication testbed," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2021*, Institute of Navigation, 2021, pp. 3698–3710, ISBN: 9780936406299. DOI: 10.33012/2021.17976.

[138] *Nuand Website – 2023-02 Release*. [Online]. Available: https://www.nuand.com/2023-02-release-122-88mhz-bandwidth.

[139] *Nuand Website – bladeRF 2.0 Micro*. [Online]. Available: https://www.nuand.com/bladerf-2-0-micro/.

[140] *Drotek DA910 multi-band GNSS antenna*. [Online]. Available: https://store-drotek.com/910-da-910-multiband-gnss-antenna.html.

[141] *Nuand Forum – 10 MHz Reference*. [Online]. Available: https://www.nuand.com/frequently-asked-questions/#How_do_I_use_a_10_MHz_reference.

[142] *Wenzel Citrine OCXO 501-24219*. [Online]. Available: https://www.quanticwenzel.com/wp-content/parts/501-24219.pdf.

[143] *Soapy SDR Wiki*. [Online]. Available: https://github.com/pothosware/SoapySDR/wiki/.

[144] *Nuand Wiki – bladeRF Installation*. [Online]. Available: https://github.com/Nuand/bladeRF/wiki/.

[145] *MacPorts Installation*. [Online]. Available: https://www.macports.org/install.php.

[146] *Nuand Website – FPGA images*. [Online]. Available: https://www.nuand.com/fpga_images/.

[147] *Nuand Website – FPGA Autoloading*. [Online]. Available: https://github.com/Nuand/bladeRF/wiki/FPGA-Autoloading.

[148] *Nuand Website – FX3 Images*. [Online]. Available: https://www.nuand.com/fx3_images/.

[149] *Nuand Forum – Synchronized TRX*. [Online]. Available: https://github.com/Nuand/bladeRF/tree/master/host/examples/bladeRF-cli/sync_trx.

[150] *GNSS-Radar*. [Online]. Available: http://taroz.net/GNSS-Radar.html.

[151] European GNSS Service Center, "Orbital and Technical Parameters," *https://www.gsc-europa.eu/system-service-status/orbital-and-technical-parameters*, 2023. [Online]. Available: https://www.gsc-europa.eu/system-service-status/orbital-and-technical-parameters.

[152] *SPCOMNAV Datasets*. [Online]. Available: https://spcomnav.uab.es/resources/acas_datasets.

[153] G. López-Risueño and G. Seco-Granados, "CN0 Estimation and Near-Far Mitigation for GNSS Indoor Receivers," in *2005 IEEE 61st Vehicular Technology Conference, Vol.4*, May 2005, pp. 2624–2628.