UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingeniería Aeronaútica y del Espacio

Navigation Systems for Rockets and Projectiles

DOCTORAL THESIS

Submitted for the degree of Doctor by:

Miguel Ángel Gómez López
Master of Science (MsC)

Madrid, 2024

UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingeniería Aeronaútica y del Espacio

Doctoral Degree in Aerospace Engineering

Navigation Systems for Rockets and Projectiles

DOCTORAL THESIS

Submitted for the degree of Doctor by:

Miguel Ángel Gómez López
Master of Science (MsC)

Under the supervision of:
Dr. Eusebio Valero Sánchez(Supervisor)
Dr. Ignacio Gómez Pérez

Madrid, 2024

Title: Navigation Systems for Rockets and Projectiles
Author: Miguel Ángel Gómez López
Doctoral Programme: Aerospace Engineering

Thesis Supervision:

Dr. Eusebio Valero Sánchez, Catedrático de Universidad (Universidad Politécnica de Madrid)(Supervisor)

Dr. Ignacio Gómez Pérez, Titular de Escuela Universitaria (Universidad Politécnica de Madrid)(Co-Supervisor)

External Reviewers:

Thesis Defense Committee:

Thesis Defense Date:

A mi hermano, por su apoyo incondicional,
y a Esther, por su amor y paciencia en cada paso de este (largo) camino.

Acknowledgement

He dudado muchas veces si este momento llegaría. Y, cuando ha llegado, como tantas cosas importantes, lo ha hecho atropelladamente.

Escribir estas líneas me ha obligado a hacer una recapitulación de los últimos 5 años. Curiosamente, esta aventura comenzó por estas fechas, un 1 de octubre de 2019, gracias al apoyo de mis dos directores: Ignacio y Eusebio. Muchas gracias por vuestra ayuda (casi amparo) en los momentos clave. Este trabajo ha sido posible gracias a vosotros dos. Ignacio, ¿quién me iba a decir, en aquella primera clase de Control en 2014, que acabaríamos aquí?

Por otro lado, me gustaría agradecer su apoyo y su disponibilidad a casi cualquier hora y día de la semana a Javier y Carles. Vosotros me habéis enseñado qué es esto del GNSS, y cómo puede llegar a absorberte y no tener final. Echando la vista atrás, fuisteis el conejo blanco que me sacó de mi libro sin ilustraciones ni diálogos, y me hizo caer por la madriguera. Javier, pasaste de ser un conocido de trabajo a un fiel amigo y compañero de aventuras (y mareos). Gran parte de los buenos recuerdos asociados a este trabajo tienen alguna relación contigo. *Comencé una tesis, pero me llevo un amigo.* Carles, gracias por estar ahí cuando ha sido necesario, sacando las castañas del fuego y aportando siempre esa serenidad que a veces nos faltaba.

Gracias a Juan, formalmente Dr. de la Mora, por acogerme durante tres meses en su laboratorio en Yale. Fue un verano único en el que aprendí cómo se hace la ciencia de manera profesional, y una estancia en la que tuve la oportunidad de redescubrir lo que significa poner todo tu empeño y pasión en algo. Y, si fue un verano único, fue sin duda gracias a Luisja (formalmente Dr. Pérez-Lorenzo). De ti nunca dejo de aprender. Aquel tiempo juntos me cambió en más de un aspecto y me ayudó a enfrentar los problemas de la vida de otra manera. Gracias. Ya están en marcha esas camisetas de *Racooning*.

También quiero agradecer formalmente al personal del INTA, en particular al área de misiles, por su apoyo y trabajo durante estos años. Este trabajo ha sido posible gracias a su profesionalidad y saber hacer. Al echar la vista atrás, suelen quedarse los buenos momentos. Jesús, Bauti: gracias por acogerme y tratarme como a uno más desde el principio. Gracias también a los que han llegado más recientemente a mi vida: Carlos, Rodrigo y Alberto (no tan reciente). Sabéis que, gracias a vuestro apoyo, no he abandonado esta aventura en más de una ocasión. Gracias por ser mis amigos, además de compañeros. Sé que no es posible nombrar a todas las personas que me he cruzado en este viaje y que me han ayudado de una u otra manera, pero si os veis reflejados en esta frase, entonces, es que os estoy agradecido.

En el plano personal, 5 años dan para muchos aciertos y errores. Este final de una tortuosa aventura coincide, además, con una época especialmente complicada. Cuando en 2019 comenzaba este trabajo, estábamos más. Primi, Cristina, Miguel: este trabajo también es por vosotros. Y, yaya: el ficus está espléndido.

Es imprescindible ahora nombrar a mi tío Mario y *las tías*: Irene e Isabel. Desde pequeños nos habéis transmitido un amor por el saber, por descubrir y por la experimentación práctica que nos define. Incluso también, por qué no decirlo, por los experimentos que podrían haber puesto en riesgo nuestra integridad física. Soy gracias a vosotros. Y esas frases están escritas en plural porque, llegado este punto, es difícil entender la vida en solitario. Gracias, Alberto, *por ser*. Esta tesis va por ti, y en muchos de los aspectos técnicos y humanos tiene tanto de ti como de mí. Es *telecociencia*. Gracias a mis padres por estar ahí. Siempre estar ahí. Aunque no lo demuestre, sé que estáis ahí.

Esther. Bueno, gr gr. Por fin. ¡Por fin! Esta tesis tiene también tanto de mí como de ti. Gracias por estar ahí todos estos años, por tu infinita paciencia y por tener siempre una palabra amable. Probablemente este no es el final (**de tesis**) que nos esperábamos, *tan sólo es un final*. Pero ahora que esta aventura ha acabado, otras llegarán. Te quiero.

Abstract

This thesis focuses on the analysis of navigation systems for rockets and projectiles, considering this analysis as a crucial step in the development of their guidance capability. The performance of the navigation system imposes limitations on the possible control and guidance strategies in an application characterized by unique challenges related to the extreme speeds and accelerations suffered by the platform. These factors have historically been a limitation for navigation systems usable on board rockets and projectiles, preventing their effective guidance.

The main objective of this research is to explore the behavior of Global Navigation Satellite Systems (GNSS) receivers installed on a rocket or a projectile. A methodology leveraging Software Defined Radio (SDR) is employed to investigate the limitations of GNSS receivers and the required architectures that enable their operation under extreme conditions, both as standalone systems and in conjunction with other sensors. In particular, the capacity of different tracking loops to deal with sudden changes in speed, acceleration, and jerk of the launch is analyzed. First, with an independent or *scalar* channel architecture, and then with a *vector* architecture, which joins and optimizes the information from all the satellite tracking channels in view. The advantages and disadvantages of these methodologies are compared, obtaining, for example, a reduction in the noise of the Doppler shift measurement for each observed satellite in the *vector* architecture.

The study also examines the fusion of the vector tracking channels with other sources of information, such as: (i) the vehicle model and (ii) inertial sensors. Through this sensor fusion, the performance of the GNSS receiver is improved during launch. One of the key objectives is to validate the capability of these systems to operate in real time, both in laboratory simulations and in real launches. To do this, tests are carried out on board a rocket, and data is collected that allows comparison with the results obtained in the laboratory.

The findings demonstrate that the combination of GNSS-SDR receivers with inertial sensors (accelerometers) provides a robust navigation system, capable of operating in real time and maintaining the PVT solution, despite the extreme accelerations and rotations characteristic of projectiles. The *ultra-tight* architecture used, which combines GNSS and inertial data at the tracking level, offers significant improvements in tracking capabilities. In addition, the flexibility of the SDR platform allows signal processing to be tailored to the specific requirements of each mission.

The results of this research highlight the potential to integrate GNSS-SDR technologies and sensor fusion techniques into rocket and projectile navigation systems. Not only can these systems achieve the required levels of accuracy and reliability, but their flexibility makes them particularly useful in environments where GNSS signals may be blocked or degraded.

This work contributes to the advancement of navigation systems for rockets and projectiles, providing both theoretical advances and practical solutions. The integration of GNSS and SDR technologies, together with sensor fusion, represents a promising approach to achieve accurate and reliable navigation in aerospace and defense applications.

Resumen

Esta tesis se centra en el análisis de los sistemas de navegación para cohetes y proyectiles, considerando este análisis como un paso crucial en el desarrollo de sus sistemas de guiado. Las capacidades del sistema de navegación imponen limitaciones a las estrategias de control y guiado posibles, en una aplicación caracterizada por desafíos únicos relacionados con las velocidades y las aceleraciones extremas sufridas por la plataforma. Estos factores han sido históricamente una limitación para los sistemas de navegación utilizables a bordo de cohetes y proyectiles, imposibilitando su guiado.

El objetivo principal de esta investigación es explorar el comportamiento de los receptores de Sistemas Globales de Navegación por Satélite (GNSS) instalados en un cohete o en un proyectil. Se emplea una metodología basada en Radio Definida por Software (SDR) para estudiar las limitaciones de los receptores GNSS y las arquitecturas que les permitan operar en condiciones extremas. En particular, se analiza la capacidad de distintos bucles de seguimiento para enfrentarse a los repentinos cambios en velocidad y aceleración característicos del lanzamiento. Primero, se estudia una arquitectura de canales independientes o *escalar*, y posteriormente se introduce una arquitectura *vectorial*, que optimiza la información de todos los canales de seguimiento de los satélites en vista. Estas metodologías se comparan, obteniendo por ejemplo una reducción de ruido en la medida del desplazamiento Doppler de cada satélite observado en la arquitectura *vectorial*.

El estudio también examina la fusión de los canales de seguimiento vectorial con otras fuentes de información, tales como: (i) el modelo del vehículo y (ii) sensores inerciales. Mediante esta fusión de sensores, se mejoran las prestaciones del receptor GNSS durante el lanzamiento. Uno de los objetivos clave es validar la capacidad de estos sistemas para operar en tiempo real, tanto en simulaciones de laboratorio como en lanzamientos reales. Para ello, se realizan pruebas a bordo de un cohete, y se recopilan datos que permiten comparar con los resultados obtenidos en laboratorio.

Los hallazgos demuestran que la combinación de receptores GNSS-SDR con sensores inerciales (acelerómetros) proporciona un sistema de navegación robusto, capaz de operar en tiempo real y mantener la solución PVT, a pesar de las aceleraciones y rotaciones extremas características de los proyectiles. La arquitectura *ultra-tight* utilizada, que combina los datos GNSS e inerciales a nivel de correlador, ofrece mejoras significativas en las capacidades de seguimiento. Además, la flexibilidad de la plataforma SDR permite adaptar el procesamiento de señales a los requisitos específicos de cada misión.

Las conclusiones de esta investigación subrayan el potencial de integrar tecnologías GNSS-SDR y técnicas de fusión de sensores en sistemas de navegación para cohetes y proyectiles. Estos sistemas, por su flexibilidad, son particularmente útiles en entornos donde las señales GNSS pueden verse bloqueadas o degradadas y pueden llegar a alcanzar los niveles de precisión requeridos.

La integración de tecnologías GNSS con SDR y la fusión de sensores, es un enfoque prometedor para lograr una navegación precisa y fiable en aplicaciones aeroespaciales y de defensa.

Contents

List of Figures

List of Tables

Abbreviations

ADC Analog-to-Digital Conversion.

APWS Advanced Precision Weapon System.

ASIC Application-Specific Integrated Circuit.

BPSK Binary Phase Shift Keying.

C/A Coarse Acquisition Code.

CEDEA Centro De Experimentación De El Arenosillo.

CHIMERA Common High-level Message Encryption and Risk Assessment.

DLL Delay-Locked Loop.

DOF Degree Of Freedom.

DSP Digital Signal Processor.

ECEF Earth-centered Earth-fixed.

EGNOS European Geostationary Navigation Overlay Service.

EKF Extended Kalman Filter.

FLL Frequency-Locked Loop.

FPGA Field-Programmable Gate Array.

GNC Guidance, Navigation & Control.

GNSS Global Navigation Satellite Systems.

GPP General-Purpose Processor.

GPS Global Positioning System.

HAS High Accuracy Service.

ICs Integrated Circuits.

IF Intermediate Frequency.

IMU Inertial Measurement Unit.

INS Inertial Navigation System.

KF Kalman Filter.

LTE Long-Term Evolution.

MEMS Micro-Electro-Mechanical Systems.

MIMO Multiple-Input Multiple-Output.

MLE Maximum Likelihood Estimator.

MPMM Modified Point Mass Model.

OCXO Oven Controlled Crystal Oscillator.

OSNMA Open Service Navigation Message Authentication.

PGK Precision Guidance Kit.

PLL Phase-locked Loop.

PRN Pseudorandom Noise.

PVT Position Velocity and Time.

RF Radio Frequency.

SA Select Availability.

SAL Semi-Active Laser.

SDR Software-defined radio.

SNR Signal to Noise Ratio.

STL Scalar Tracking Loop.

SV satellite Vehicle.

TLE target location error.

TTFF Time To First Fix.

TCXO Temperature-Compensated Crystal Oscillator.

VTL Vector Tracking Loop.

WAAS Wide Area Augmentation System.

Chapter 1

Introduction

1.1 Motivation and scope of the project

The navigation of rockets and projectiles presents a distinct set of challenges due to the extreme conditions encountered during flight. These vehicles operate in high-dynamic environments, characterized by exceptionally high speeds, intense accelerations, compact designs, and often unstable trajectories. Unlike traditional aircraft or missiles, which benefit from established navigation systems, rockets and projectiles require specialized systems to meet the demands of their fast-paced, short-duration flights.

This project's motivation stems from the need to develop navigation systems that are more robust and precise in these challenging conditions. The high-G forces, rapid velocity shifts, and inherent instability of spin-stabilized and fin-stabilized projectiles further complicate navigation. Cost constraints and the brief operational window of projectiles also hinder the application of conventional navigation technologies.

To address these issues, this research explores the integration of Global Navigation Satellite Systems (GNSS) with Software-Defined Radio (SDR) technology. The goal is to develop a system that fuses GNSS data with inputs from inertial sensors, enhancing accuracy and reliability for real-time applications. By advancing these navigation technologies, this research seeks to contribute to the aerospace engineering field by offering practical solutions to improve the precision and reliability of navigation systems in extreme conditions. The results could significantly improve the performance of guided munitions and other high-dynamic aerospace applications, where both precision and resilience are critical.

1.2 Objectives

This thesis focuses on advancing the development and analysis of navigation systems for rockets and projectiles, addressing the challenges posed by high-dynamic environments characterized by extreme velocities, accelerations, jerks and the necessity of pack the system in small places, under high mechanical loads. The primary objective is to explore the effectiveness of Global Navigation Satellite Systems (GNSS) integrated with Software-Defined Radio (SDR) technology in this scenario. By combining GNSS receivers with inertial sensors through sensor fusion techniques, the study aims to improve navigation accuracy and reliability. Performance will be compared with GNSS-only systems and a broad context of the problem and the solutions explored will be given. Furthermore, the thesis introduces a novel architecture for vector-tracking GNSS receivers, specifically tailored for GNSS receivers based on Software Defined Radio architecture. The receiver selected is GNSS-SDR, as it is the only GNSS SDR based receiver that (i) works in real-time and (ii) is an open source project, with all the code available [64]. This is an open-source receiver and the only open-source GNSS receiver known to work in real-time and be tested in different real environments. In addition, the study highlights the potential of SDR technology as a research tool. Different signal processing techniques are applied to the same RF scenario in a repeatable and consistent manner. In this way, the resilience of the navigation system is tested. Moreover, real field tests of the GNSS-SDR receiver onbard a rocket are performed. We aim to validate these technologies in real-time applications, offering solutions for contexts where precision and resilience are paramount, and extreme customization could be an approach to achieve them.

1.3 State of the art

1.3.1 First developments

On the first appearance of tests that try to put sensors in munitions was published on 1993: They tried to instrument 40mm grenades to study the behavior of "sub-munitions" that they had problems characterizing with radar [109]. Their objective at this point was to obtain the behavior of the submunition, in particular its spin velocity. The year 1993 is also the year that the first 24 NAVSTAR GPS satellites are fully operational and begin to provide their service [68].

From 1993 to 1996, a multitude of articles appeared with projectile-sized sensors, with telemetry and ruggedized [35], [138], [159]. Even with the stated objective of characterization only. These developments, from 1996 to 1998, go in parallel with the technology of inertial MEMS sensors [36], sometimes working to improve the MEMS sensors that existed at that time [17]. The first work studying the effect of rotation on GPS antennas appears [8], [14].

In 1998 it was already possible to measure pitch and yaw, in addition to spin (roll). At this time, MEMS technology is more mature and, using automotive grade, the first article appears in which both laboratory tests (drop tower, or air cannon) and aboard a 2.75-inch rocket are carried out [104]. A warhead is manufactured with all the sensors and telemetry, which is capable of supporting up to 95,000g in an air cannon and flying on the 2.75-inch rocket rotating at 35rps(Hz).

Related to these experiments, the first article appeared in 2001 (which until now had been proposed as experimental or measurement tools for the development of ammunition), with results aimed at using these fuzes as navigation systems [18], [19]. The first reports also appear on how to perform INS+GPS navigation on projectiles [15]. One of the first open documents on complete flight controllers appears 3 years later [99].

However, in parallel to these technical publications, several military projects are being developed. Those that stand out for their subsequent success are: COPPERHEAD (1970), widely used in the Gulf War, Operation Desert Storm in 1991, EXCALIBUR, which began as a program in 1992, and was operational in September 2007 (with a scope of 23 km and without SAL guidance; later it would reach 40km with BaseBleed). Moreover, the GMLRS (guided ammunition for the MLRS) in 2000 already showed some technical publications on how to guide rockets. In fact, this guided ammunition [61] expressed for the first time the need to decouple the movement of the control and guidance sections from the rest of the projectile movement).

At this point, as will be addressed later in the thesis, The *kit* concept for controlling rockets and projectiles had already arisen. This concept focuses on trajectory control fuzes for artillery, as an idea adapted from the successful JDAM and Paveway technologies. The aim is to enhance precision by adjusting the flight path of projectiles in real-time, improving accuracy in dynamic conditions, as an upgrade of exiting munition. It is important to note that neither EXCALIBUR nor COPPERHEAD used this concept. EXCALIBUR is not a conventional 155mm artillery projectile since, unlike typical projectiles which rely on rotational stabilization, EXCALIBUR uses additional fins for stability and control. In contrast, COPPERHEAD functions more like a rocket in its operation designed as controllable from the beginning, unlike regular rocketry artillery.

GPS, on the other hand, began in 1978 (after *Transit* program success) and was operational in 1993 as mentioned. Even with the initial Select Availability (SA), which decreased the accuracy achievable by the receiver around 100m [68], aroused enormous interest. The joint use of GPS with other systems (inertial sensors mainly) was researched from the beginning [74], [129], [169]. Quite a lot of work has been done, both in the theoretical aspect of hybridization [5], [46], [70], as well as in the practical [47], [119]. A summary of all this effort came in 2003 with the work of Petovello [124] among others. There are three main ways to merge INS and GNSS [142]. Loosely [142], Tight[143] and Ultra-Tight[141]. Each of these algorithms has various advantages and/or disadvantages, especially with respect to the case of rockets and projectiles.

Spanish rocket artillery

The Army acquired 12 Teruel launchers. From 1987 to 2011, they equipped two batteries of the 62^{nd} Field Rocket Launcher Artillery Regiment (currently 63^{rd} Field Rocket Launcher Artillery Regiment) of Astorga (León). It originally had Teruel-2 rockets (also called simply T-2), with a range of 18.5 km, but they were replaced in 2006 by MC-25 rockets with a new warhead, manufactured by Expal. Developed by this company and the Polytechnic University of Madrid, they had a range of 25 km. These rockets were somewhat longer in length, which required modifications to the launchers. This update was made to extend its useful life into the early 2010s, not without some problems [38].

Since then, the recovery of rocketry capacity in field artillery has been one of the priorities of the Spanish Army. Finally, in 2023, the new SILAM program (*Sistema Lanzador de Alta Movilidad*) was made public. Based on the work of the Israeli company Elbit, it will have guided and conventional rockets [39]. Some experimental work has been done on the *kit* control concept for rockets [93].

Summary

Between 1998 and 2002, advances in MEMS-based inertial navigation systems (INS), GPS technology (initially with large receivers later miniaturized), and algorithms for navigating artillery rockets and projectiles laid the foundation for modern developments in precision munitions. The first commercially viable guided projectiles, such as Excalibur, emerged in 2007 during the Iraq war. All of them are based on GPS receivers. Prior to this, systems like COPPERHEAD (and the ATACMS) were developed, though they were rocket ammunition, and lacked the miniaturization that would later define the field.

Since 2002, efforts have focused on reducing the dependence on GPS and enhancing terminal guidance precision. In 2002, development began on the Advanced Precision Weapon System (APWS) with Semi-Active Laser (SAL) guidance, which became operational in 2012. Similarly, the Excalibur has been updated with a SAL seeker (*Excalibur S*). *Excalibur S* was started in 2013, reached operational status in 2020.

1.3.2 Guidance, Navigation and Control

There are two very different projectiles, those passively stabilized by fins and those passively stabilized by rotation. Among the firsts, we can include rocket-type ammunition mainly, while the rotation-stabilized ones are the conventional artillery projectiles. These seconds are the most challenging from the technical point of view.

Navigation

There are various methods to calculate the attitude of a rocket or projectile, with the most widely adopted being based on the Inertial Navigation Systems (INS) and GNSS systems[1]. However, numerous studies focus on determining the attitude solely from magnetic data because of its inherent advantages.

In addition, alternative methods to GPS to determine the position of ammunition remain limited. The only operational alternatives are terminal guidance systems, such as laser-based or radar-based methods (e.g. SPACIDO), alongside emerging technologies under investigation, such as signals of opportunity [166] or sensor arrays [43]. Furthermore, terminal guidance systems are limited to use in either the final phase of engagement or when continuous localization and tracking of the target has been maintained from the beginning. This presents a significant operational constraint, making them unsuitable for indirect fire applications, as will be discussed in Chapter 4. Finally, the level of GPS/INS integration plays a significant role in system performance, with three main integration approaches commonly used: Loosely, Tight, and Ultra-Tight integration. Ultra-Tight integration, which operates at the hardware level, has only recently been implemented by manufacturers.

---

[1] Although the term GNSS us used, it should be noted that nations have not integrated systems such as Galileo or BeiDou for these applications. The use of GLONASS is neither known

Both attitude and position, either absolute or relative to the target, are essential to be able to perform control and guidance tasks. Attitude capabilities would define control algorithms and performances, while position and velocity data enable both guidance and control.

Control Systems

Increasing the capacity of the control system has been the subject of extensive study [40], [56], [60] through an active or passive mechanism. The objective of this increase in performance is to improve maneuverability [54], avoid instabilities [30], correct the trajectory sooner [81] or hit moving targets [55], [57]. Although guided projectiles with internal mass devices were initially explored in the 1980s [76], [112], almost all later works have focused on the use of fins in different ways. However, this concept, internal mass devices, has received renewed attention, with theoretical developments in the early 2000s [58], [130] and more recent studies in 2023 [135].

Guided projectile concepts using aerodynamic control devices can generally be divided into two categories based on the stabilization method: fin-stabilized systems (e.g., the American M712 Copperhead and M982 Excalibur) or spin-stabilized systems like the XM1156 Precision Guidance Kit and the SPACIDO system [153]. Spin-stabilized projectiles, unlike fin-stabilized ones, rely on high-speed rear-body rotation for stability [116], which introduces certain drawbacks, such as complex mechanical requirements for fuzes (due to huge accelerations).

The capabilities of the navigation system will determine the types of controls that can be performed. Or, in other words, it will be the requirements achievable by the navigation system that will lead to choosing the suitable algorithm in the controller, for achieving the desired performances.

Guidance Systems

Guidance systems are typically GNSS-based, such as those found in Excalibur, PGK, Silver Bullet, or TopGun systems. These systems require on-board trajectory propagation to estimate the projectile landing point. Alternatively, radar, SAR, or imaging systems are being studied as complementary [116] or substitute technologies. Some modern projectiles combine GPS with Semi-Active Laser (SAL) guidance, such as the new Excalibur variant, while others rely solely on SAL (as APWS). Systems equipped with terminal guidance benefit from less precision requirements in the early stages of flight, since final corrections are made upon entering terminal guidance *cone*, but suffer from their inability to perform *indirect* fire.

1.4 Organization of the Thesis

The thesis is organized into eight distinct chapters. It begins with a discussion of the dynamics of projectile flight in detail (Chapter 2). The following chapter shifts focus to the onboard navigation systems for projectiles (Chapter 3). In Chapter 4, the focus is on GNSS receivers based on Software Defined Radio (SDR), leading to Chapter 5), which investigates the utilization and limits of GNSS receivers in projectiles. Chapter 6 introduces the concept of sensor fusion navigation and presents one of the findings of the thesis: a Vector Tracking

Loop architecture for GNSS-SDR receivers. The penultimate chapter, Chapter 7, is dedicated to presenting the results derived from the research. The thesis culminates with Chapter 8, which draws conclusions and encapsulates the findings of this work.

Chapter 2

Projectiles flight dynamics

This chapter provides a foundation for the behavior of projectiles in flight. Several of the ideas presented, and the flight physics detailed, are necessary for (i) developing an accurate simulator for the trajectories that are introduced in the GNSS signal simulator, and (ii) understanding the key differences between the dynamics of projectiles and aircraft and missiles, focusing on their stabilization methods and the forces they encounter during flight.

By examining the different models used for projectile flight, this chapter also sets the stage for integrating these dynamics into a navigation system. This integration will be detailed in chapter 6, and will show how the information of the flight model can help to the navigation system in these high-dynamic environments.

2.1 Comparisons with Aircrafts and Missiles

Aircrafts are designed, on the other hand, to transport cargo or people, powered throughout the flight (except gliders). They have control elements. inherent to their design: ailerons, elevators, and rudders, defining their maneuverability. **Missiles** are guided weapons, designed to be controlled and directed towards a specific target, they often have fins or wings for stability and maneuverability, and their cost and size are not as restrictive in their design as in projectiles. Finally, **projectiles** are unguided bodies projected forward by forces, following a predetermined trajectory. They are generally small, designed for single use, and may or may not be stable in their trajectory.

In essence, the main differences lie in their purpose, design considerations, and flight characteristics. Projectiles are simple, unguided, and cost-effective, while aircraft are complex, guided, and designed for repeated use. Missiles, on the other hand, combine aspects of both, being guided like aircraft but single-use like projectiles. Although missiles and projectiles share objective and some design boundaries, the overall cost of the final element differences them heavily.

Table 2.1: A comparison of projectiles, aircraft, and missiles from various perspectives.

	Projectiles	**Aircraft**	**Missiles**
Purpose	Projected or impelled forward by forces. Generally not guided.	Designed to carry cargo/people from one point to another.	Guided, designed to be controlled and directed towards a specific target.
Trajectory	Generally follow a predetermined trajectory.	Powered throughout the flight (except for gliders).	Often have fins or wings to provide stability and maneuverability during flight.
Use	Single use.	Designed for continuous use.	Single use.
Size	Generally designed to be relatively small.	Size are directly related to the lift they are capable of generating and, therefore, the load they are capable of carrying.	Size is not as strong a restriction as in projectiles.
Stability	May, or may not, be stable in its movement along the trajectory.	Stable by design.	Stable by design.
Movement	Characterized by large angles of attitude and large angles with respect to the incident current. Movement with very high angular velocities and very high accelerations.	They have control elements (ailerons, elevator, rudder) to provide maneuverability during flight. Low angular speeds. Small angles of attack, balance, or slip.	The angle of attack and accelerations to which they are subjected are lower than those of projectiles.
Wind Effect	The wind matters, and affects it, but simulations without taking it into account can have reasonable and useful results.	The direction and speed of the wind are key in the analysis of the flight.	Not specified.
Cost	Low cost is a design driving force and a strong restriction.	Not specified.	Cost is not as strong a restriction as in projectiles.

2.2 Projectiles stabilization

There are two main ways to achieve projectile stabilization, namely: spin-stabilized and fin-stabilized. Spin-stabilized projectiles keep their orientation steady by rotating around their own axis, whereas fin-stabilized projectiles gain stability through the strategic placement of fins and by having their mass center positioned ahead of the pressure center. The mathematical models used in their flight simulations are, therefore, distinct. Moreover, for spin-stabilized projectiles, gyroscopic stability is sought, whereas the dynamic stability condition will be used in fin-stabilized projectiles.

2.2.1 Spin Stabilized projectiles

Projectiles that are spin-stabilized revolve around their own longitudinal axis. The rotation of the mass generates gyroscopic forces, which help maintain the alignment of the bullet's length axis, counteracting the destabilizing torque that occurs when the center of pressure is ahead of the center of mass. This mechanism ensures that the projectile's orientation remains steady. If the spin is insufficient, it can lead to tumbling, while excessive spin can prevent the nose of the shot from dipping as it follows its trajectory. It is crucial to examine static stability, also known as gyroscopic stability, to prevent the projectile from tumbling. One of the main benefits of spin stabilization is its simplicity in keeping the projectile directed towards a specific direction. The modified projectile linear theory trajectory report proves useful in estimating the trajectories of high-spin-stabilized projectiles, which allows us not to solve the spin angle along the rest of the degrees of freedom and therefore avoids the use of a really small time step in the integration scheme [91]. The models that will be detailed later and used in the simulations take advantage of this fact, as will be explained later.

However, spin-stabilized projectiles with course correction fuzes actuated by fixed canards have the problem of great coupling in both the normal and lateral directions because of intensive gyroscopic effects. This was first identified by Morrison [109] [110], and has been thoroughly researched [92], [111]. Ollerenshaw started to develop different model predictive controllers in [116] leading to interesting results, and then in [118] develops a general approach. The limits of stability in the flight of an asymmetric projectile with activated canards were then investigated in [30], with a general summary and overview in [154]. Recently, [60] or de Celis [25] have presented the efficiency and suitability of different canard configurations. This research showed a limited maneuverability in different directions due to the limited correction capabilities. A target-aiming method is proposed to make full use of the correction ability of canards in [162].

2.2.2 Fin Stabilized projectiles

Fin-stabilized projectiles rotate at a low speed or do not spin at all during flight. Their design ensures static stability, with the center of mass placed in front of the center of pressure. The dynamics of these projectiles are influenced by two opposing moments: the rolling moment induced by the fins, which increases the spin, and the spin damping moment, which works to decrease the spin. These opposing forces eventually balance out, leading to a steady state of spin. Interestingly, this steady-state spin is often slow. This slow spin rate is advantageous because it averages small asymmetries, either of mass or related to the configuration, over long trajectories. This improves the flight stability and accuracy of the projectile.

2.3 Projectiles' Aerodynamic Forces and Moments

Symbol	Aerodynamic coefficient
C_{D_0}	Drag force linear coefficient
$C_{D^2_\alpha}$	Drag force square coefficient
C_{L_α}	Lift force linear coefficient
$C_{L^3_\alpha}$	Lift force cubic coefficient
C_{N_q}	Pitch damping force coefficient
C_{M_α}	Overturning moment linear coefficient
$C_{M^3_\alpha}$	Overturning moment cubic coefficient
C_{M_q}	Pitch damping moment coefficient
C_{mag-m}	Magnus moment coefficient
C_{mag-f}	Magnus force coefficient
C_{spin}	Spin damping moment coefficient

Table 2.2: Aerodynamic coefficients nomenclature.

2.3.1 Forces

Drag force

The aerodynamic drag force F_d acting on a projectile opposes its forward velocity. Drag is the most early force studied on exterior ballistics. The vector equation used for calculating the drag force is:

$$\vec{F}_d = -\frac{1}{2}\rho A C_d V^2 \vec{i} \tag{2.1}$$

Where:

- ρ is the air density,
- A is the reference area of the projectile,
- C_d is the drag coefficient,
- V is the velocity of the projectile,
- $\vec{i}$ is a unit vector pointing in the direction of the velocity, $\vec{i} = \frac{\vec{V}}{V}$.

The drag coefficient (C_d) is a dimensionless quantity that depends on the shape and surface properties of the projectile. It is typically determined experimentally through testing or simulation. On the other hand, the term $\frac{1}{2}\rho V^2$ is the *dynamic pressure*, and plays a central role in the calculus of all forces and aerodynamic moments. Finally, the reference area A is often taken as $A = \frac{\pi d^2}{4}$ where d is the reference diameter of the projectile.

Lift force

In the context of projectiles, particularly spin-stabilized ones like bullets or artillery shells, the lift and normal forces acting on the projectile tend to pull the projectile in the direction its noise is pointing. More concisely: If the nose of the projectile is *above its trajectory*, then it makes it to *climb*; whereas if the nose of the projectile is *below its trajectory*, then it makes it to *dive* These forces influence the trajectory and stability of the projectile, causing deviations from its intended trajectory, particularly at longer ranges. In most projectile flights, the lift force is minimized compared to the drag force. The lift forces acting on a projectile can be calculated as:

$$\vec{F}_{\text{lift}} = \frac{1}{2}\rho V^2 A C_{L_\alpha} \left[\vec{i} \times (\vec{x} \times \vec{i})\right] \tag{2.2}$$

Where:

- F_{lift} is the lift force,
- C_{L_α} is the lift coefficient.

The lift coefficient C_{L_α} depends on the shape of the projectile and α_t: the total angle of attack. More specifically depends on $\sin(\alpha_t)$. The lift force not only produces the drift of projectiles for long ranges as introduced, but also causes aerodynamic jump and epicycilc swerve. This coefficient C_{L_α} has non-linear behavior and should be fully characterized with simulations, wind tunnel, and flight tests.

Magnus force

The Magnus force is a phenomenon that occurs when a spinning object moves through a fluid medium. It results in a sideways force perpendicular to both the direction of the object's motion and its axis of rotation. This force is responsible for the curved paths of spinning balls in sports such as soccer, baseball, and tennis.

The Magnus force can be expressed mathematically as:

$$\vec{F}_M = \frac{1}{2}\rho V^2 A \left(\frac{pd}{V}\right) C_{\text{mag-f}}(\vec{i} \times \vec{x}) \tag{2.3}$$

Where:

- F_M is the Magnus force,
- C_{mag-f} Magnus force coefficient, which depends on the object's spin and total angle of attack α_t,
- p is the spin velocity,
- d is the reference diameter of the projectile.

The Magnus force is perpendicular to both the velocity vector and the axis of rotation of the spinning object. It is proportional to the product of spin and total angle α_t. It is important to note that it vanishes for $\alpha_t = 0$ or $p = 0$. It can cause the object to curve in the direction of the Magnus force, especially at longer ranges.

Pitch damping force

The pitch damping force acting on spin-stabilized projectiles is generally much smaller than the lift and drag force. In fact, it is quite difficult to measure in tests or a wind tunnel. Together with the Magnus force, it is considered in fact only for completeness, but it is neglected when practical calculations are done. However, the moments produced by these forces will be important for the dynamic stability of the projectiles.

The pitch damping force contains two parts. One is related to the transverse angular velocity q_t and the other is proportional to the rate of change of the total angle of attack $\dot{\alpha}_t$.

Mathematically, the pitch damping force can be expressed as (in vectorial form):

$$\vec{F}_{\text{pitch}} = \frac{1}{2}\rho A d C_{M_q} V \left(\frac{d\vec{x}}{dt}\right) + \frac{1}{2}\rho A d C_{M_{\dot{\alpha}}} V \left[\left(\frac{d\vec{x}}{dt}\right) - \left(\frac{d\vec{i}}{dt}\right)\right] \tag{2.4}$$

In this case, is easier to interpret if we express the pitch damping force in scalar form:

$$F_{\text{pitch}} = \frac{1}{2}\rho V^2 A \left[\left(\frac{q_t d}{V}\right) C_{N_q} - \left(\frac{\dot{\alpha}_t d}{V}\right) C_{N_{\dot{\alpha}}}\right] \tag{2.5}$$

Where:

- q_t is total transverse angular velocity; $q_t = \sqrt{q^2 + r^2}$,
- q is pitching angular velocity and r is the yawing angular velocity,
- $\frac{d\vec{x}}{dt}$ is the rate of change of the unit vector that defines the attitude of the body. Therefore, noted that q and r are contained inside it,
- $\dot{\alpha} = \frac{d\alpha}{dt}$ is rate of change of the total angle,
- C_{N_q} is the pitch damping coefficient due to q_t,
- $C_{N_{\dot{\alpha}}}$ is the pitch damping coefficient due to $\dot{\alpha}$.

2.3.2 Moments

Spin damping moment

The spin damping moment refers to the torque exerted on a spinning object to resist changes in its spin rate or rotational motion. In the context of spin-stabilized projectiles, such as bullets or artillery shells, this moment opposes the spin of the projectile.

Mathematically, the spin damping moment M_{spin} can be expressed as:

$$\vec{M_{\text{spin}}} = \frac{1}{2}\rho V^2 A d \left(\frac{pd}{V}\right) C_{m_\omega} \vec{x} \tag{2.6}$$

Where:

- ρ is the air density,
- V is the velocity of the projectile,
- A is the reference area of the projectile,
- d is the projectile reference diameter,
- p is the spin rate of the projectile,
- C_{m_ω} is the spin damping moment coefficient,
- $\vec{x}$ is a unit vector along the projectile symetry axis.

As it was introduced, M_{spin} is always negative, and therefore the spin damping moment coefficient C_{m_ω} is always negative. It depends on various factors such as the projectile's shape, surface properties, and aerodynamic characteristics. It is determined experimentally through testing or simulation.

On the other hand, the quantity $\frac{pd}{V}$ has a special significance in exterior ballistics; it is the spin per caliber, or in non-dimensional units, the ratio of axial spin to forward velocity. It is relevant to note that in general the spin damping is much smaller than the effect of drag on velocity. This makes the spin per caliber to increase along a trajectory for a spin stabilized projectile.

Overturning (pitching) moment

The overturning moment is the moment associated with the lift force. It is also known as the pitching moment, which refers to the torque exerted on an object that tends to rotate it around its lateral axis (pitch axis), causing it to pitch or rotate nose-up or nose-down. The pitching moment coefficient C_{m_α} depends on various factors, including the angle of attack, the shape and size of the fins, and the aerodynamic properties of the projectile-fins system. It is determined experimentally through testing or simulation.

Mathematically, the overturning moment $\vec{M}_{pitch}$ can be expressed as:

$$\vec{M}_{pitch} = \frac{1}{2}\rho A d C_{m_\alpha} V^2 (\vec{i} \times \vec{x}) \tag{2.7}$$

Or its value (scalar):

$$M_{pitch} = \frac{1}{2}\rho V^2 A d C_{m_\alpha} \sin \alpha_t \tag{2.8}$$

Where:

- C_{m_α} is the Overturning moment coefficient,
- α_t is the total angle of attack.

A non-spinning projectile with positive C_{m_α} is usually unstable, as an increasing value of the lift induced a moment that increases the total angle of attack, again increasing the lift.

Therefore, in order to make a projectile stable, two approaches are usually followed: Add fins to the tail of the projectile, whose lift contributes to a negative C_{m_α} . The other is to give the projectile an axial spin. The amount of spin needed to make a projectile stable will lead to the concept of gyroscopic stability.

Magnus moment

The Magnus moment is the moment associated to the Magnus effect force. The Magnus moment coefficient C_{mag-m} can be positive or negative. This depends on the shape, center of gravity of the projectile. In can also depend on the *yawing* motion itself.

Mathematically, the Magnus moment $\vec{M}_{Magnus}$ can be expressed as:

$$\vec{M}_{Magnus} == \frac{1}{2}\rho V^2 Ad\left(\frac{pd}{V}\right) C_{mag-m}\left[\vec{x}\times(\vec{i}\times\vec{x})\right] \tag{2.9}$$

Where:

- ρ is the air density,
- V is the velocity of the projectile,
- A is the reference area of the projectile,
- r is the radius from the center of mass to the point where the Magnus force is applied,
- C_L is the lift coefficient,
- ω is the angular velocity of the projectile.

In 2.3.1 the Magnus force was introduced as negligible. However, the Magnus moment must be taken into account because large values of C_{mag-m} will dramatically impact the dynamic stability.

Pitch damping moment

The pitch damping moment refers to the moment generated by the pitch damping force. It also contains two parts, one proportional to the total angle velocity q_t and the other proportional to the rate of change of the total angle of attack $\dot{\alpha}_t$.

Mathematically, the Pitch Damping Moment $\vec{M}_{\text{pitch}}$ can be expressed (in vectorial form) as:

$$\vec{M}_{\text{pitch}} = \frac{1}{2}\rho A d^2 C_{M_q} V\left(\vec{x}\times\frac{d\vec{x}}{dt}\right) + \frac{1}{2}\rho A d^2 C_{M_{\dot{\alpha}}} V\left[\left(\vec{x}\times\frac{d\vec{x}}{dt}\right) - \left(\vec{x}\times\frac{d\vec{i}}{dt}\right)\right] \tag{2.10}$$

In this case, is easier to interpret if we express the pitch damping force in scalar form:

$$M_{\text{pitch}} = \frac{1}{2}\rho V^2 Ad\left[\left(\frac{q_t d}{V}\right) C_{M_q} - \left(\frac{\dot{\alpha}_t d}{V}\right) C_{M_{\dot{\alpha}}}\right] \tag{2.11}$$

Where:

- C_{M_q} is the pitch damping moment coefficient due to q_t,
- $C_{M_{\dot{\alpha}}}$ is the pitch damping moment coefficient due to $\dot{\alpha}_t$.

As happened with the Magnus moment, the pitch damping moment is more relevant than the pitch damping force, as it has an important influence on the dynamic stability of the projectile. A positive pitch damping moment increases the total transverse angular velocity, destabilizing the projectile.

Positive sums of the pitch damping moment coefficients ($C_{m_\alpha} + C_{M_q}$) could be observed in the transonic or subsonic speeds of some projectile shapes, making them dynamically unstable during that period.

Rolling moment (for canted fins)

The rolling moment, particularly in the context of canted fins on a projectile, refers to the torque generated by the fins due to their asymmetrical arrangement, which induces a rotation around the projectile's longitudinal axis (roll axis). This moment is significant in controlling the projectile's orientation and stability during flight.

Canted fins are fins that are intentionally angled or offset from the longitudinal axis of the projectile. This configuration is often used to impart spin or induce aerodynamic forces that help stabilize the projectile's flight. When the projectile encounters disturbances, such as crosswinds or uneven airflow, the canted fins produce a rolling moment that counteracts these disturbances and maintains the stability of the projectile.

Mathematically, the rolling moment $\vec{M}_{roll}$ can be expressed as:

$$\vec{M}_{roll} = \frac{1}{2}\rho V^2 S_{fins} L_{fins} C_{L_\delta} \delta_{fins} \tag{2.12}$$

Where:

- C_{L_δ} is the rolling moment coefficient,
- S_{fins} is the total surface area of the fins,
- L_{fins} is the average distance from the center of pressure of the fins to the roll axis,
- δ_{fins} is the angle of attack of the canted fins.

The rolling moment coefficient C_{L_δ} depends on various factors such as the angle of attack of the fins, their shape, and the aerodynamic properties of the projectile-fins system. It is determined experimentally through testing or simulation.

By adjusting the angle of attack of the canted fins, the rolling moment can be controlled to stabilize the projectile's flight and maintain its desired orientation. This control mechanism is essential for achieving accuracy and precision in projectile trajectories, especially over long distances or under adverse environmental conditions.

2.4 Different models for Projectiles flight

2.4.1 Vacuum

The Vacuum Model is a simplified model used for simulating the motion of projectiles. It is often used as a baseline or starting point for more complex models.

In the vacuum model, it is assumed that the projectile is moving in a vacuum, which means there is no air resistance or any other forces acting on the projectile except gravity. This model is based on the principles of classical mechanics.

The trajectory of a projectile in a vacuum is a parabola. This is due to the constant acceleration caused by gravity acting downwards. The only factors determining the trajectory in this model are the initial velocity and the launch angle.

However, it is important to note that while the Vacuum Model is simple and easy to calculate, it is not very accurate for most real-world applications because it ignores air resistance and other forces. More complex models like the 5-DOF, 6-DOF, or Modified Point Mass Model are often used when a higher level of accuracy is required.

2.4.2 Point Mass Model (PMM)

The point mass model simplifies the analysis of projectile motion by treating the projectile as a single point in space without considering its size, shape, or aerodynamic effects. This model is based on the assumption that the projectile's mass is concentrated at a single point, simplifying calculations while providing a basic understanding of projectile motion.

Here is how the point-mass model typically works:

Initial Conditions: The model requires initial conditions such as the projectile's initial velocity, launch angle, and launch position.

Trajectory Calculation: Using the laws of classical mechanics, such as Newton's laws of motion, and basic kinematic equations, the trajectory of the projectile is calculated. These equations incorporate the effects of gravity and the initial velocity and angle of the projectile.

Flight Path: The trajectory of the projectile can be determined, including its range, maximum height, time of flight, and other characteristics. The flight path is typically parabolic in nature, neglecting air resistance and other external factors.

Simplified Analysis: The point mass model simplifies the analysis of projectile motion, making it easier to understand the basic principles governing the motion of projectiles. It is often used in introductory physics courses and as a starting point for more advanced analyses.

Although the point mass model provides a simplified representation of projectile motion, it does not account for factors such as air resistance, aerodynamic effects, spin stabilization, or variations in projectile shape. For more accurate predictions, especially at high speeds or over long distances, more sophisticated models that incorporate these factors may be necessary. However, the point-mass model remains a useful tool for introductory studies and initial estimations of projectile behavior.

2.4.3 Modified Point Mass Model (MPMM)

The Modified Point Mass Model (MPMM) is a commonly used method to simulate the trajectory of projectiles, particularly in the creation of firing tables. It's a compromise between accuracy, complexity, and computational cost.

In the MPMM, all of the mass of the projectile is assumed to be concentrated at its center of gravity. This model requires four types of input data. It is a four-degree-of-liberty model where the first three degrees correspond to the three degrees of translation of the point mass.

The MPMM has been used for various types of projectiles, including base-burn projectiles and fin (aerodynamically) stabilized projectiles. For base-burn projectiles, the model takes into account factors such as aerodynamic drag, the mass flow rate of the base-burn motor, and changes in base pressure. For fin-stabilized projectiles, the model considers phenomena such as forces and moments.

However, it is important to note that while the MPMM is more computationally efficient than the six-degree-of-freedom (6-DOF) model, it may not be as accurate. The choice of model depends on the accuracy required, the complexity of the projectile and flight conditions, and the available computational resources.

2.4.4 Five DoF Model

The five Degrees-of-Freedom (5-DOF) model is a simplification of the sic Degrees-of-Freedom (6-DOF) model, which is often used in the simulation of projectile motion. The 5-DOF model is particularly useful when the roll motion of the projectile is not significant or can be ignored.

In the 5-DOF model, the motion of the projectile is described by five independent variables, typically including three translational motions (forward/backward, left/right, up/down) and two rotational motions (pitch and yaw). Roll motion is not considered in this model.

This model is often used to simulate the flight behavior of artillery projectiles. Provides a balance between computational efficiency and accuracy. Although less accurate than the 6-DOF model, it requires fewer computational resources, making it suitable for scenarios where a high level of precision is not necessary.

However, it is important to note that the choice of model (5-DOF, 6-DOF, or others) depends on the specific requirements of the simulation, including the level of accuracy needed, the characteristics of the projectile, and the available computational resources.

2.4.5 Six DoF Model

The six degree-of-freedom (6-DOF) model is the most comprehensive model used to simulate the motion of projectiles. It is often used when high accuracy is required, such as in the design and analysis of guided projectiles [87].

In the 6-DOF model, the motion of the projectile is described by six independent variables: three translational motions (forward/backward, left/right, up/down) and three rotational motions (pitch, yaw, roll). This model takes into account all the forces that act on the projectile, such as aerodynamic drag, lift, Magnus force and moment, pitch damping force and moment, spin damping moment and rolling moment.

The 6-DOF model is particularly useful for simulating the flight behavior of spin-stabilized projectiles. It can closely capture their trajectories with high accuracy and detail. However, it comes with the drawbacks of long computation time and the need for many aerodynamic coefficients that can be difficult to obtain [139], [147]. The method used in this work is [115].

A comparative study between the 6-DOF model and the Modified Point Mass Trajectory Model found that the 6-DOF model was more precise, but the Modified Point Mass Trajectory Model was a more simplified solution [73].

It is important to note that the choice of model (6-DOF, 5-DOF, or others) depends on the specific requirements of the simulation, including the level of accuracy needed, the characteristics of the projectile and the available computational resources.

2.5 Simulator and Trajectory Propagator used in the Thesis

The selected vehicle is a 140 mm axisymmetric rocket, rotation-stabilized with a spin rate of approximately 60 Hz. The rocket is launched in the supersonic regime [66]. A five-degrees-of-freedom (5-DoF, 2.4.4) rocket was assumed to calculate the rocket trajectories. In this model, the most relevant simplification of physics (with respect to a full six-degree-of-freedom model) arises from assuming perfect axial symmetry around the longitudinal axis of the projectile, both geometrically and in terms of mass distribution, and at least trigonal aerodynamic symmetry [21]. Assuming this axial symmetry, it is possible to avoid the calculation of the forces during the spin turn of the projectile, and the integration step can be much larger. Decomposing the angular momentum vector into components parallel and perpendicular to the longitudinal axis of the projectile, Equations (2.28), (2.29), and (2.30), the axial angular velocity p and the derivative of the axial vector $\dot{\vec{x}}$ are obtained. The vector $\vec{x}$ therefore defines the attitude of the projectile axis during flight [98].

The model runs in a ground-fixed coordinate system, originating at the launch point, where the y axis follows the local vertical upward, the x axis is horizontal with the launch direction, and the z axis forms a right-handed coordinate system with the previous ones. On the other hand, the module of gravity acceleration $\vec{g}$ depends on the latitude of the launch point and the components on the relative position of the projectile with respect to that point. The atmosphere model has been implemented using the international standard atmosphere, which follows the standard defined in [157].

The complete state of the rocket over time is obtained by integrating Newton's law for its center of mass (2.13) and the conservation of angular momentum (2.20). Regarding the first, the evolution of position $\vec{X}$ is obtained from (2.14) with the initial conditions imposed by (2.15). The used notations is as followed $\vec{a}_G^{B_i}$ stands for the body acceleration (G indicates the point of the body being tracked, the center of mass), ρ for the air density, d is the rocket diameter, $\vec{x}$ axial vector , $\vec{g}$ gravity and $\vec{T}$ the rocket thrust:

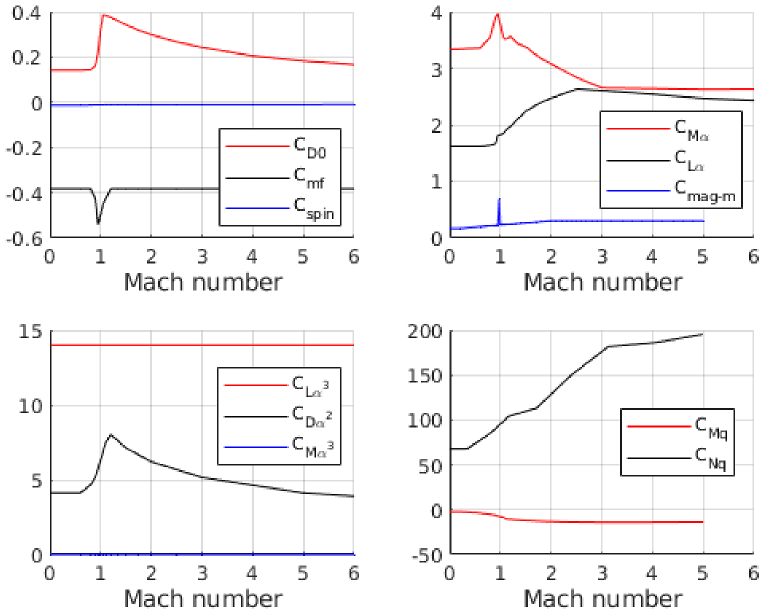


Figure 2.1: Aerodynamic coefficients used by AeroFI SW, an implementation of [115] and [25].

$$\vec{F} = m \cdot \ddot{\vec{X}} \ , \tag{2.13}$$

$$\vec{F} = m\ddot{X} = \vec{DF} + \vec{LF} + \vec{MF} + \vec{PDF} + m\vec{g} \ , \tag{2.14}$$

$$\dot{\vec{X}}_0 = \dot{X}_0 \begin{bmatrix} \cos{(QE)} \\ \sin{(QE)} \\ 0 \end{bmatrix} \ , \tag{2.15}$$

where the forces included are ($\vec{v} = \dot{\vec{X}}$):

$$\vec{DF} = -\frac{\pi \rho d^2}{8} \left(C_{D_0} + C_{D_{\alpha^2}} \alpha^2 \right) v \vec{v} \ , \tag{2.16}$$

$$\vec{LF} = \frac{\pi \rho d^2}{8} \left(C_{L_\alpha} + C_{L_{\alpha^3}} \alpha^2 \right) \left(v^2 \vec{x} - (\vec{v} \cdot \vec{x}) \vec{v} \right) \ , \tag{2.17}$$

$$\vec{MF} = -\frac{\pi \rho d^3 C_{mag-f}}{8 I_x} \left(H \cdot \vec{x} \right) \left(\vec{x} \times \vec{v} \right) \ , \tag{2.18}$$

$$P\vec{D}F = -\frac{\pi \rho d^3 \left(C_{N_q}\right)}{8I_y} v \left(\vec{H} \times \vec{x}\right) \ . \tag{2.19}$$

Regarding the second, the evolution of the axial vector $\vec{x}$ is obtained from Equations (2.21) and (2.22), with initial conditions imposed by (2.23).

$$\dot{\vec{H}} = \sum \vec{M} \ , \tag{2.20}$$

$$\vec{x} = \vec{x_0} + \int_0^t \dot{\vec{x}} dt \ , \tag{2.21}$$

$$\dot{\vec{x}} = \frac{\left(\vec{H} \times \vec{x}\right)}{I_y} \ , \tag{2.22}$$

$$\vec{x_0} = \begin{bmatrix} \cos\left(QE\right) \\ \sin\left(QE\right) \\ 0 \end{bmatrix} \ , \tag{2.23}$$

where QE stands for the launch *quadrant elevation* [98]. The derivative of the axial vector is obtained from the angular momentum $\vec{H}$, taking into account the following effects:

- *Overturning Moment*: Moment due to angle of attack,

$$O\vec{M} = \frac{\pi \rho d^3}{8} \left(C_{M_\alpha} + C_{M_{\alpha 3}} \alpha^2\right) v \left(\vec{v} \times \vec{x}\right) \ , \tag{2.24}$$

- *Pitch Damping Moment*: Perpendicular angular velocity damping moment,

$$P\vec{D}M = \frac{\pi \rho d^4}{8I_y} \left(C_{M_q}\right) v \left[\vec{H} - \left(\vec{H} \cdot \vec{x}\right) \vec{x}\right] \ , \tag{2.25}$$

- *Magnus Moment*: Moment of force due to the Magnus effect,

$$M\vec{M} = -\frac{\pi \rho d^4}{8I_x} C_{mag-m} \left(\vec{H} \cdot \vec{x}\right) \left[\left(\vec{v} \cdot \vec{x}\right) \vec{x} - \vec{v}\right] \ , \tag{2.26}$$

- *Spin Damping Moment*: Axial angular velocity damping moment [55],

$$S\vec{D}M = \frac{\pi \rho d^4}{8I_x} C_{spin} \vec{v} \left(\vec{H} \cdot \vec{x}\right) \vec{x} \ . \tag{2.27}$$

Therefore, angular momentum law evolution,

$$\vec{H} = I_x p \vec{x} + I_y \left(\vec{x} \times \dot{x}\right) = O\vec{M} + P\vec{D}M + M\vec{M} + S\vec{D}M \ , \tag{2.28}$$

$$\left(\vec{H} \times \vec{x}\right) = I_y \dot{x} \ , \tag{2.29}$$

$$\left(\vec{H} \cdot \vec{x}\right) = I_x p \ . \tag{2.30}$$

This will be the model used to simulate the trajectory of the rocket in Section 7.

2.6 Stability and Control

As an initial step in controller development, simulations have been carried out of the effect of applying a constant force at the front end of the projectile and with directions perpendicular to the axis of symmetry (or to the vector $\vec{x}$).

To identify the direction of the applied force, the axes are defined with origin at the point of application of the force, located in a plane perpendicular to the vector $\vec{x}$. The vectors $\vec{d}$ and $\vec{a}$ are defined in Equation 2.31. $\vec{d}$ is a horizontal vector perpendicular to $\vec{x}$, directed to the right, $\vec{a}$ is defined by forming a right trihedron with $\vec{d}$ and $\vec{x}$, pointing towards the top.

$$\begin{aligned} \vec{d} &= \frac{\vec{x} \times \vec{j}}{|\vec{x} \times \vec{j}|} \\ \vec{a} &= \vec{d} \times \vec{x} \end{aligned} \tag{2.31}$$

From the previous axes, an angle of application of the force is defined, ψ_c, with origin in the vector $\vec{a}$ and a positive direction according to the vector $\vec{x}$. Its interpretation is similar to the angle ψ.

Equation 2.32 shows the expression with which the applied force is calculated, which produces the moment indicated in Equation 2.33. In this expression, the value l_c identifies the distance between the center of gravity and the point of application of the force. To graphically show the interpretation of the defined axes and angles, four cases of force application with different values of ψ_c are represented in Figure 2.2.

$$\vec{F_c} = |F_c| \left(\vec{d} \sin \psi_c + \vec{a} \cos \psi_c\right) \tag{2.32}$$

$$\vec{M_{F_c}} = l_c \vec{x} \times \vec{F_c} \tag{2.33}$$

2.6.1 Constant force.

To evaluate the response of the projectile when the force is applied, a case is initially taken with initial elevation $QE = 800$mil, initial velocity of 563.9 m / s and a force $|F_c| = 10$N. Figure 2.3 shows the effect of the applied force on the dispersion of the point of impact,

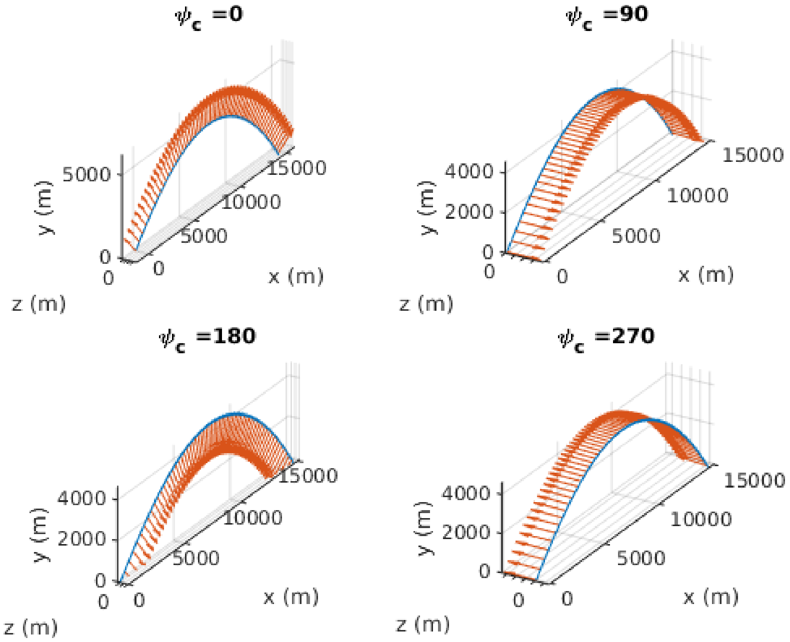


Figure 2.2: Applied forces.

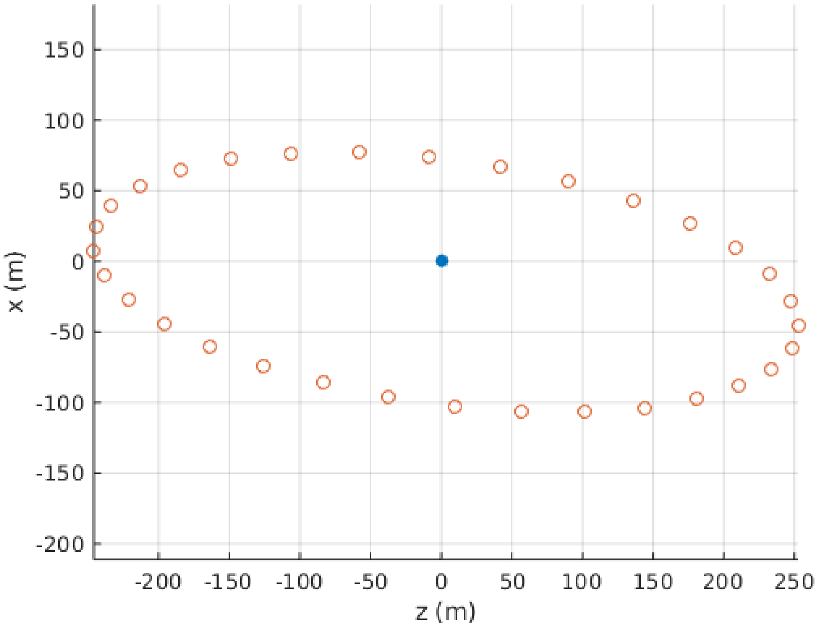


Figure 2.3: Dispersion at the point of impact due to force.

showing that for this elevation of the draft and in this case, where the force is constant, there is greater capacity for lateral correction than longitudinal correction.

To begin identifying the direction in which the projectile moves when applying the force as a function of ψ_c, the Figure 2.4 qualitatively shows the direction and magnitude of the displacement relative to the case of reference. The origin of each of the arrows with respect to the center represents the direction in which the force is applied, and the arrow represents the relative displacement of the point of impact. It can be seen that the relationship is complex and not intuitive.

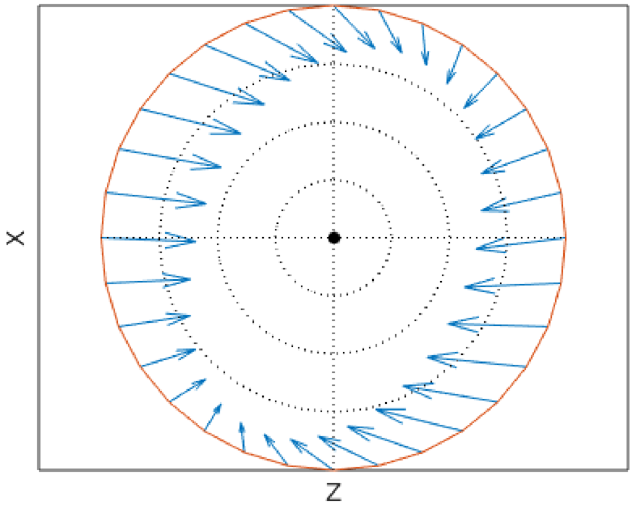


Figure 2.4: Displacement of the final impact point due to constant force F_c applied along the trajectory.

To identify which displacement corresponds to each angle of force application, ψ_c, another angle ψ_t is defined according to Equation 2.34. This angle identifies the direction in which the impact point moves with respect to the reference case without applied force, the origin is the $\vec{x}$ axis and positive clockwise.

$$\psi_t = \text{atan2}(\Delta Z, \Delta X) \tag{2.34}$$

Using this angle, the relative phase, $\psi_t - \psi_c$ and the magnitude of the displacement are now represented as a function of the angle of application of force, ψ_c.

The results shown vary greatly if the elevation of the force is changed. Figure 2.8 shows the comparison by increasing and decreasing the elevation of the force, and a huge difference in the results is observed. In the case with a greater elevation of the force, the control capacity in range is lost, as the effects of the force in the ascending and descending sections of the trajectory are canceled, so a more refined control method will be required later.

In the same idea, Figure 2.11 shows what the response is like for each angle ψ_c, again it is observed that with a lower draft elevation the response is more uniform at any angle ψ_c, with an almost constant phase and magnitude of the response. On the other hand, in the high elevation shot, the response depends greatly on the angle of application of the force.

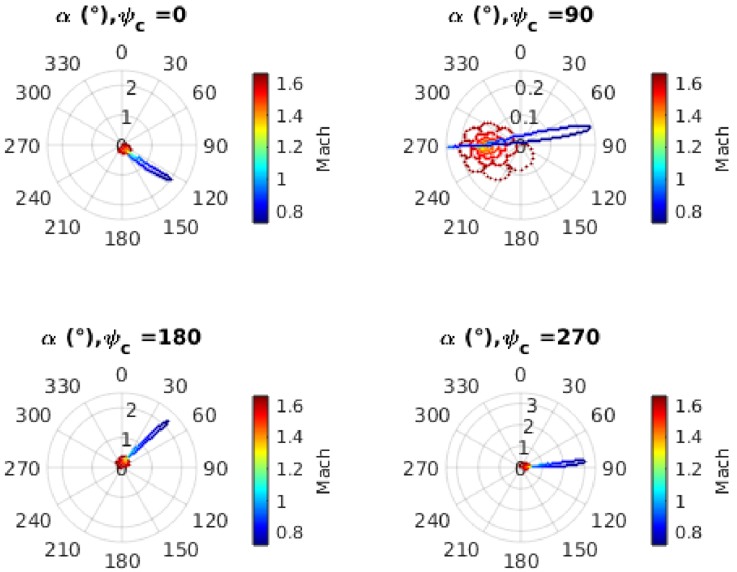


Figure 2.5: Evolution of the angle of attack α as a function of the angle of application of the force ψ_c.

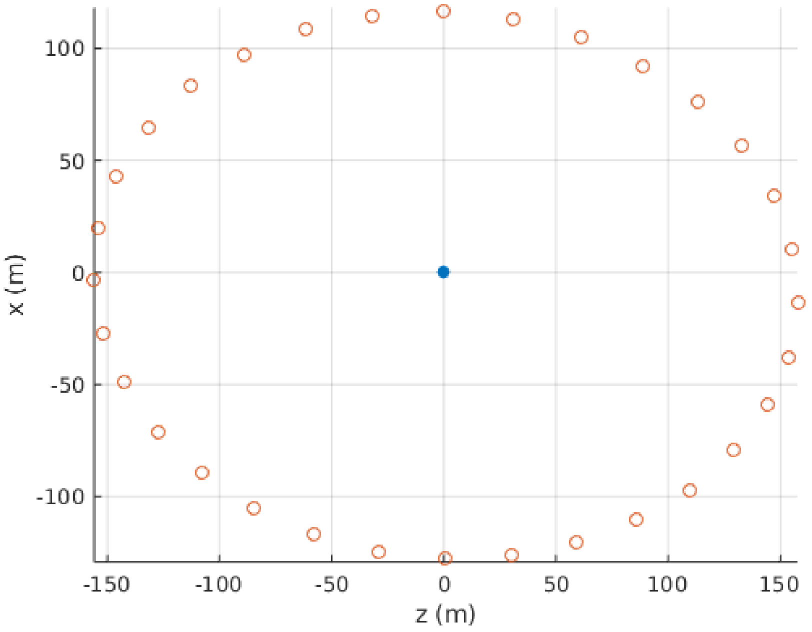


Figure 2.6: QE = 600 mil.

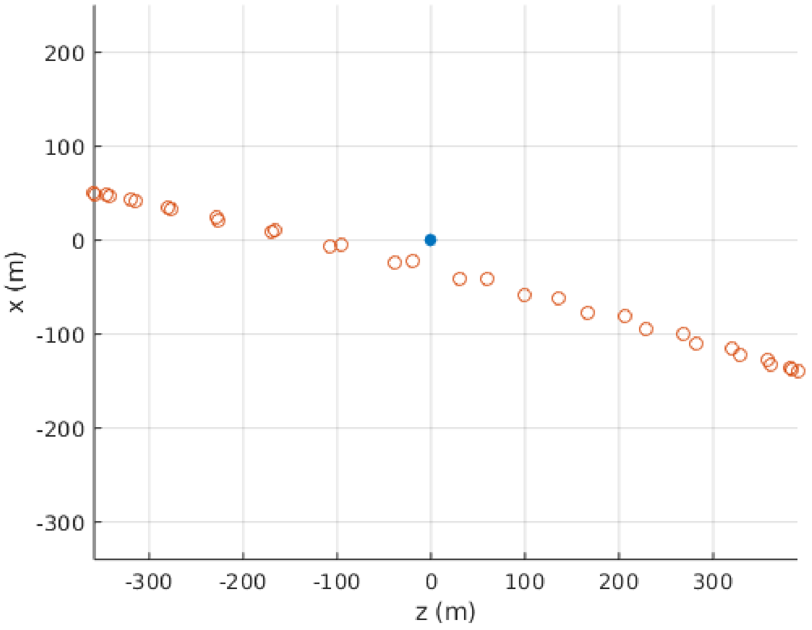


Figure 2.7: QE = 1000 mil.

Figure 2.8: Dispersion due to constant force F_c for 33.75 deg and 56.25 deg elevations.

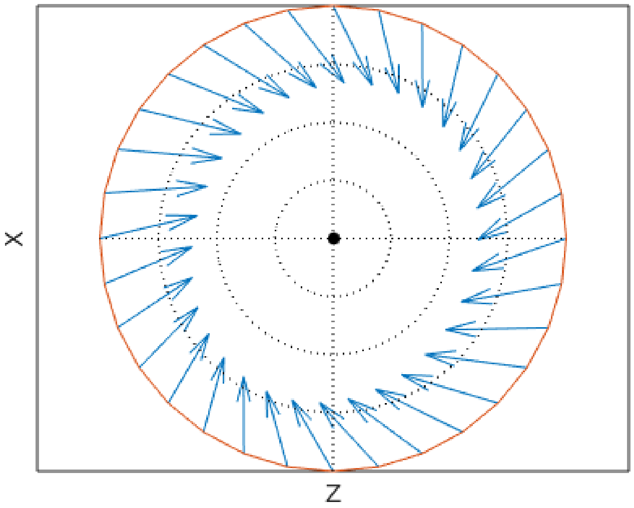


Figure 2.9: QE = 600 mil.

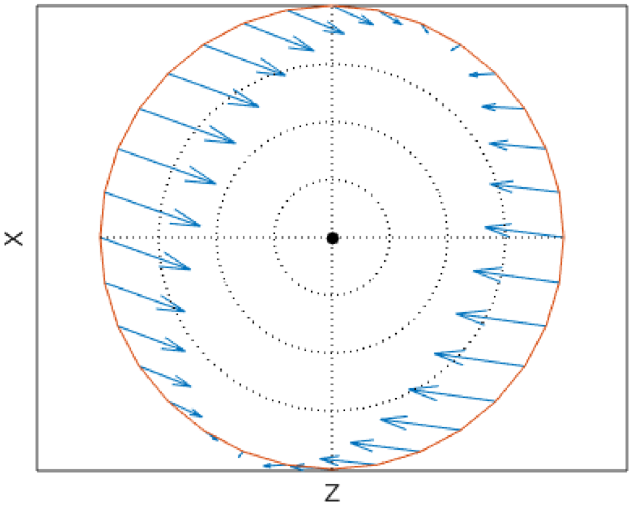


Figure 2.10: QE = 1000 mil.

Figure 2.11: Movement of the final impact point due to constant force F_c along the trajectory. Calculated for 33.75 deg and 56.25 deg elevations.

Chapter 3

GNC aboard projectiles

Once we have introduced what the mechanics of the flight of rockets and projectiles are like, vehicles of interest in this work, it is clear that the flight of projectiles through the atmosphere is influenced by various factors. Wind, temperature, and muzzle velocity errors greatly affect the accuracy of both the predicted trajectory and the point of impact. Downrange errors are the most significant, and the area where projectiles land is often modeled as an ellipse. Unfortunately, errors related to temperature and wind cannot be eliminated by merely improving the gun or projectile. Instead, in-flight corrections will be required to mitigate these effects. A complete Guidance, Navigation & Control system is necessary for this task.

Control and Guidance systems of the projectiles will be vital to be able to carry out this mission, and this chapter introduces the concepts related to the Guidance, Navigation & Control of rockets and projectiles in a general way, which will be used in the rest of the chapters of the thesis.

However, it is important to recall that these tasks can only be performed once the system is aware of their *state*. This task is completed by the navigation system, the system that will be the focus of the rest of the work.

3.1 Trajectories

The trajectory of a projectile can be divided into three main phases:

1. **Supersonic Phase**: This is the first third of the flight where the projectile velocity is supersonic. Acceleration and Jerk will reach their maximum values within this phase.
2. **Summit Phase**: Following the supersonic phase, is the summit phase. The dynamic pressure and spin rate decrease but remain significant.
3. **Terminal Phase**: This is the last third of the flight path. The spin rate influences control authority, and the risk of jamming/spoofing in GNSS signal is the highest.

We will detail the phases as follows:

Supersonic Phase

The projectile remains supersonic during the first third of its flight, after leaving the muzzle of the cannon. Afterwards, the projectile transitions to subsonic speeds. During the supersonic phase, the dynamic pressure is high, making aerodynamic control surfaces highly effective. In this phase, even small control inputs (forces or moments) can significantly influence the impact point. However, since most of the flight occurs at subsonic speeds, it is logical to optimize the control surfaces for subsonic operation. Transonic aerodynamic effects near the end of the supersonic phase can complicate the control surface performance. The supersonic phase is also when the projectile is closest to the gun, facilitating easier data uplink, such as actual muzzle velocity measurements and radar data of position and velocity of the projectile.

In addition, in this phase, the GNSS receiver can start acquiring satellite signals. If a **warm start** is available, the receiver will have all the information (ephemeris, alamanac, satellites on view, and approximate values of Doppler shift and code delay) available except the actual Time Of the Week. Therefore, it only has to go quickly from acquisition to tracking and recover this value from the Navigation Message. If any data was uploaded, before the launch to the receiver, then a **cold start** will occur. This implies that the receiver must: (i) star the acquisition with not *priori* knowledge, (ii) once a satellite or satellites are acquired and tracked, download from the GNSS signal the ephemeris data, alamanac data, ToW, and converge a solution. The Worst-Case time for this scenario is 12.5 minutes, which is unacceptable for this application.

Summit Phase

After the supersonic phase, when the projectile has passed through the transonic airspeed region, the change in airspeed (and hence axial acceleration) is no longer as high. This marks the start of the middle third part of the flight path, which we shall call the summit phase. At the beginning of this phase, the dynamic pressure and spin rate are still high, meaning that the projectile has good efficiency of aerodynamic control surfaces and good gyroscopic stability. These conditions generally hold throughout the summit phase.

Terminal Phase

The terminal phase is the last third of the flight path. During this phase, the spin rate has decreased to a level where it must be taken into account in any calculation of the available control authority. In fact, the available control authority is generally limited during this phase by the gyroscopic stability properties of the projectile [92]. Furthermore, during this phase, the likelihood of encountering jamming, spoofing, is greater, as well as GNSS outages due to other reasons such as the target point being inside a forest or urban cannon.

3.2 Error Sources

The main sources of dispersion in long-range applications are variations in wind, temperature, and the muzzle velocity of the projectile when it leaves the cannon, even variations in projectile weight can also have a significant influence. This dispersion leads to downrange and cross-range errors. Downrange errors almost always dominate over cross-range errors, and the region within which 50 % of the projectiles fall, centered around the nominal impact points, is most often modeled as an ellipse (with its major axis in the downrange direction) [128]. For example, a 1 deg error in the assumed sea-level atmosphere temperature will yield a range error exceeding 50 m at a 25km target range, and a 1m/s tailwind error will yield almost the same down-range error. Under the same conditions, an error of 1m/s in the muzzle velocity will give about a 25 m downrange error. These errors are of the same order as the kill radius for a fragmentation-type grenade (which is about 50m).

It should be noted, however, that temperature and wind errors are distinctly different in character from the other errors since temperature and wind errors can never be removed by improving the performance of the gun or the projectile. The only way to remove the effects of these, often large, errors is to correct them in-flight by guidance. Therefore, there are good reasons to try to improve the precision of conventional indirect fire weapons by introducing guidance and control capabilities to the projectiles.

3.3 Control and Guidance

The guidance and control problem for spinning projectiles, where the control actuators are confined to a tip-mounted assembly, possesses distinctive features that differentiate it from other types of missile control. Firstly, the projectile was not originally designed for control and the placement of control actuators is not ideal in relation to the dynamics [117]. Therefore, the common procedure in control theory in which a controller is introduced in the dynamics chain with the aim of replacing (improving) the open-loop dynamics with the desired closed-loop dynamics will be impossible for an arbitrary closed-loop dynamics behavior. In other words, this approach implies that only small changes in the natural dynamics of the projectile can be performed.In particular, the gyroscopic stability boundaries are often unalterable and the addition of control mechanisms mounted on the tip degrades stability properties.

Therefore, although it is generally possible that a tip-mounted guidance and control assembly can provides trajectory correction, this correction is limited the extent to which the natural dynamics can be modified [58]. The primary expectation is trajectory correction rather than a comprehensive control redesign. This is the primary difference with missiles, drones, or loitering ammunition.

3.3.1 Control architectures

The control actuators for spinning projectiles are confined to a tip-mounted assembly when the concept of a fuze or control **kit** is used. Conventional control mechanisms for projectiles include movable canards at the nose that provide 2D course corrections for spin-stabilized projectiles. The canard assembly must roll independently for spin-stabilized projectiles but not for fin-stabilized ones. For example, in the PGK system the control is made by a fuze that is spin-stabilized with four despun canards. These provide constant torque and lateral force for maneuvering. The braking mechanism adjusts and stabilizes the force orientation in any roll angle, operating in a bank-to-turn fashion. However, simpler mechanisms, such as drag brakes and spin brakes, offer 1D course correction. When used together, they may provide 2D correction. Both mechanisms act along the body x-axis, resulting in irreversible maneuverability.

The simplest way to introduce correction in the crossrange (drift) direction, i.e., to add 2D-correction capabilities, is to employ a spin brake. Spin brakes consist of a set of canards mounted with a small angle against the wind, thereby inducing a moment on the fuze, which can be more or less transferred to the projectile via a mechanism. However, spin brakes can only make small corrections in crossrange. To achieve significant crossrange correction, a free-spinning fuze assembly must be employed. In this concept from, the fuze canard assembly was stabilized relative to the Earth, resulting in a simple control formulation and efficient use of control surfaces. The limiting factor for such designs is the stability limits represented by the maximum total angle of attack before gyroscopic instability occurs, as studied by [92], [112].

3.3.2 Guidance

In guided munitions contexts, three main types of guidance are commonly considered [40]: trajectory shaping, trajectory prediction [32], and trajectory following [60]. However, trajectory shaping, often inefficient in terms of energy, is generally more suitable for meeting terminal conditions, which we will leave unspecified, except for the endpoint. Our focus is on guided projectiles with course-correcting fuzes, as opposed to those with specially designed airframes that possess the capability to meet terminal conditions.

Predictive guidance is inherently an open-loop control strategy. The natural application of predictive guidance involves using it repeatedly along a flight path, providing a natural feedback mechanism. This feedback mechanism is based on the actual state of the projectile (position and velocity) to compute a new trajectory leading to the target, even if previous deviations occurred [55]. This differs from path following guidance, which incorporates a feedback mechanism by design, with the aim of tracking a given path. Path following introduces a natural "integral action" into the guidance due to tracking position as a variable. However, even if the projectile is brought back to the nominal path, it will generally be on a trajectory to the target only if the state coincides with the nominal state, making further errors and corrections likely. Nonlinear approaches of this, using artificial neural networks, can be found in the work of De Celis [23], [24], [26].

There are cases where predictive guidance and path following guidance are equivalent. One example of this equivalence occurs when path following guidance is enough to bring the projectile onto a path leading to the target, with the projectile having a velocity such that the state is along an admissible trajectory. This is used as inspiration for the braking strategies used by some implementations of course-correcting fuzes. If there is excess energy near the end of the flight path, proper braking (velocity brake for downrange correction and spin brake for drift correction) can be applied to redirect the projectile onto a lower-energy path leading to the target. It is important to note that using path following guidance up to the point of braking can be catastrophic if there is a loss of navigation data, as the projectile may have too high a velocity to hit the target. Furthermore, to apply braking, the projectile must revert to predictive guidance for the remainder of the flight [153].

3.4 Navigation

Once the different mechanism for Control and Guidance has been introduced, we relate them with the Navigation system. The first idea is that, if a purely axial control mechanism is implemented, only position and velocity will be needed for guidance. In this case, from the data availability point of view, a GNSS receiver would suffice (removing jamming / spoiling attacks). On the other hand, if canard control is to be used, then the body frame attitudes are definitely needed for defining the orientation of the lateral control force.

This information, attitude, position, and velocity will generally be provided by an integrated GNSS/INS in other vehicles. Such navigation systems are routinely employed in many missiles as an example and are available today as commercial-off-the-shelf components. There are series of deeply integrated MEMS and GNSS receivers, gun-hardened, and anti-jam capable, with INS navigation capabilities when GNSS is not available.

However, this may not be a viable solution given the demands of guided ammunition. Apart from the elevated cost, there are other reasons for avoiding the INS. First, the target location error (TLE) is a critical parameter for the effectiveness of any precision guidance concept. It may not be on par with the very low CEP (< 5 m) specified for the INS unit, making the cost for the latter unjustified. There are also known observability issues regarding the estimation and in-flight calibration of the inertial measurement unit bias and drift errors that may be of concern. The IMU is not needed for feedback purposes in the control system. The fact that the airframe should be dynamically stable from the beginning, together with modest requirements on the speed of response to guidance, makes it possible to do so without an acceleration and angular rate feedback control.

3.4.1 GNSS with other sensors

In a standard integrated GNSS/INS system, IMU signals are used as input to the process model: accelerometer signals drive the navigation equations, and gyroscopes the attitudes. For spin-stabilized or fin-stabilized projectiles, the total angle of attack is usually small. Analytical approximations exist for the steady-state value of the relative orientation of the body frame x-axis with respect to the velocity vector, referred to as the *yaw of repose* [98], chapter 2. Used as corrections to the flight path angles, the body system pitch and yaw angles can be directly obtained from the velocity vector. The roll angle is generally estimated with other sensors (for example, magnetometers, dedicated gyroscopes, or thermopiles [131]).

This reduction in the number of navigation states (by removing pitch and yaw) is also used in the impact point prediction by the guidance system. The small angle of attack makes a *Modified Point Mass Model* [98] of Chapter 2 sufficiently accurate for trajectory prediction (3-DOF dynamics plus expression for the yaw of repose).

3.4.2 Without INS

A further simplification of this concept is presented in [120]. Only the GNSS receiver gives the position and velocity. In this case, the roll rate and roll angle are measured using an accelerometer and a magnetometer. This is an uncoupled architecture. The problem of estimating the angle of rotation of a spinning projectile is commonly also referred to as *up-finding*. Much effort has been put into methods to perform this task. To save costs, space, and power consumption, it would be much preferable if the up-finding capability could be included in the GNSS receiver [152]. All navigation functions would then be performed by one sensor only. A comprehensive explanation of the physical effects can be found in [97]. More detailed works are [27] and [102]. Work in other directions has been done, with theoretical promising results, like [16], [132], [136], [138], [155], but for the moment they lack of final working implementations.

Both the PGK and the VAPP concepts use standalone GNSS receivers that include the determination of the roll angle [55]. The ECF relies on a GPS C/A receiver only, not including the roll calculus. The SPACIDO projectile has no navigation system at all; instead, the trajectory corrections are based on measurements from a muzzle velocity radar. However, this approach has important drawbacks. GNSS receivers are very vulnerable to jamming, and this becomes one of the most important questions to deal with for any GNSS-only system [88], [140]. Success really hinges on the effectiveness of included anti-jamming capabilities.

Chapter 4

GNSS-SDR receivers

A GNSS SDR, or Global Navigation Satellite System Software-Defined Receiver, is a software-based solution designed to convert received GNSS signal samples into Position Velocity and Time (PVT) estimates. This technology operates on a general-purpose computer and relies on an Analog-to-Digital Conversion (ADC) front-end to sample the incoming RF signal. However, beyond the ADC, all functionality is realized through software. This concept gives rise to three distinct categories of software receivers.

One category comprises real-time receivers, which are typically implemented as monolithic or modular software packages. These packages are usually written in C or C++. The emphasis here is on optimizing these receivers for run-time efficiency and stability [64]. Another category involves teaching and research tools. In this case, the software is written in high-level programming languages such as Python or Matlab. The focus shifts to code readability and flexibility, making these tools valuable for educational purposes and research projects [13], [126]. The third category, known as snapshot receivers, is designed to process very short batches of signal samples efficiently. These receivers find applications in scenarios where GNSS receiver Size, Weight, Power and Cost (SWaP-C) is essential [63].

4.1 Software Defined Radio

The origin of software-defined radio (SDR) dates back to 1995 when J. Mitola introduced the idea to the community [106]. Prior to this, efforts were underway to improve radio flexibility and interoperability. Two years later, in 1997, after completing his Ph.D. thesis [2], Akos started as an Assistant Professor in the Systemteknik Department of Luleå University of Technology in Sweden [3]. He introduced a Matlab-based GPS SDR receiver to his students as a class project and challenged them to make it run as fast as possible. It was at this moment when a GPS SDR receiver achieved real-time operation. They were able to process 60 seconds of IF data in 55 seconds. Meanwhile, Kai Borre, was also developing Matlab code for GPS receivers, focusing on navigation and related functions. The combined efforts of Akos, Borre, and others culminated in the first book on GNSS SDR receivers [13], and the development of the SoftGPS Matlab receiver. This was the first open version of a GNSS SDR featuring a novel front-end design for sampled data collection and storage.

In 2009, the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) began working on a multi-band, multi-system software receiver. This software receiver was published as open source under a GNU General Public License (GPL) in 2010 and has since evolved continuously. They based their efforts on the work of Akis and Borre, using their book, and the work done by RTKlib [151] for the PVT engine. GNSS-SDR is to this day the only open-source GNSS SDR receiver capable of processing signals in real time [64], and has been tested onboard different vehicles. In fact, this approach has become a standard. Primarily written in C++, it leverages object-oriented programming to provide significant flexibility in radio design while efficiently implementing complex operations [125], particularly through the use of SIMD (Single Instruction, Multiple Data) instructions. It is worth noting that without these instructions, it is unlikely that acquisition and tracking tasks could be performed in real-time on a conventional PC.

SDR offers advantages such as reduced development efforts and enhanced flexibility, but they also place higher demands on hardware, including power consumption and processing performance, compared to traditional application-specific integrated circuits (ASICs). However, to be used in real time, in an efficient manner (a recent benchmark can be found in [69]), this receivers still needs a hardware acceleration that if not done by ASICs, should be done by FPGA [50], [83]. The tasks that are typically accelerated by hardware are the acquisition and tracking of incoming SV signals (Figure 4.1).

The lack of reconfigurability in current commercial-off-the-shelf receivers and the emergence of new radionavigation signals and systems have made software receivers an attractive approach for designing new architectures and signal processing algorithms. This approach allows for the exploration of new signal structures and frequency bands. In contrast, the advent of new GNSS systems, the modernization of existing ones, and the deployment of augmentation systems represent a driving force for the continued development and adoption of software-defined receivers [52]. The introduction of new GNSS systems such as Galileo and COMPASS, along with the modernization of existing systems such as GPS (with L1C, L2C and L5 signals) or GLONASS (with L3OC signals, and the movement from FDMA to CDMA), has significantly expanded the number of available signals, frequency bands, and constellations. This happened at a velocity never seen up to date in the GNSS community and has also spawned a competition among the constellations and the services they are able to provide to the user (either open, commercial, or military).

These new GNSS services, such as the High Accuracy Service (HAS, Galileo), Open Service Navigation Message Authentication (OSNMA, Galileo) and Common High-Level Message Encryption and Risk Assessment (CHIMERA, GPS), have emerged as part of these developments. HAS, for instance, focuses on providing centimeter-level positioning accuracy, opening up new possibilities for precise navigation in areas where high precision is necessary. OSNMA addresses the issue of spoofing, by authentication of navigation signals, ensuring the integrity and authenticity of the GNSS message. CHIMERA adds another layer of security through high-level message encryption and risk assessment. These new services are the response of the GNSS system developers to the security and trustworthiness question, at a time when spoofing attacks are becoming more and more common [160].

These advancements in GNSS services, signal structures, and enhancement systems have not only expanded the possibilities for precise positioning and navigation, but also introduced new challenges and opportunities for software-defined receivers. The increased complexity and diversity of signals and services require flexible, adaptable solutions, software-defined receivers are uniquely positioned to address these reality. As a result, the demand for software-defined receivers continues to grow, with ongoing research and development efforts aimed at harnessing the full potential of these technological advances to further improve global navigation and positioning capabilities [20], [78], [113]. The performance of the high-rate computations needed while maintaining the desired flexibility inherent in a software-based approach ends in a trade-off between computational efficiency and flexibility. Traditional *hardware-based* GNSS receivers rely on specific hardware, such as ASICs, for the most computationally intense processing, sacrificing flexibility in the process, but being more powerful and cheaper than their SDR counterparts.

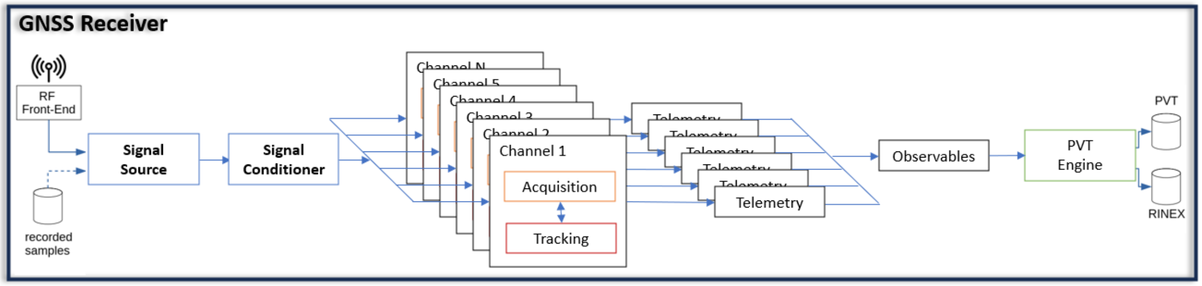


Figure 4.1: General high-level structure of a GNSS Receiver.

4.1.1 Front Ends in GNSS-SDR

As technology advanced, the development of chips for mobile communications also accelerated. These new chips offered several key enhancements: instantaneous bandwidth up to tens of MHz, more bits on the ADC resolution, MIMO capabilities, and even transmit capabilities. This occurred in parallel with the increase in popularity of software-defined radios and the support of crowdfunding initiatives. In 2001 the GNU radio project started, and it was developed in a time in which a wide array of front-end peripherals were emerging.

These new ICs were much more powerful than those available and had prices that placed them within reach of the community, converting the world of amateur radio, which although numerous was a niche, into something attainable for a wide range of people: from universities and small research centers to *makers*. This gave great speed to new developments. A software-defined radio designed for GNSS applications benefits from its position within the RF spectrum at that point, where it is flanked on both sides by signals of interest to the civilian population. In particular, applications such as digital video broadcasting, terrestrial broadcast (DVB-T), and digital video broadcasting, second generation satellite (DVB-S2) receivers have driven the availability of a diverse range of cost-effective RF integrated circuits (IC) for GNSS receivers [53]. These RF ICs are tunable to GNSS frequencies, typically spanning from 900 MHz to 2.1 GHz. They, along with dedicated GPS ICs, formed the foundation for the early GNSS SDR front ends. In fact, the hardware developed for [13] was based on classic GNSS-customized chips, and it was quickly replaced by these new IC.

A selection of GNSS-compatible SDR peripherals, including both new and established offerings, is presented in Table 4.1, showcasing the diversity and capabilities of these hardware enablers. These peripheral RF hardware forms the essential bridge that allows GNSS SDRs to tap into the wealth of information contained within the radio frequency spectrum, enabling precise and adaptable navigation and positioning solutions.

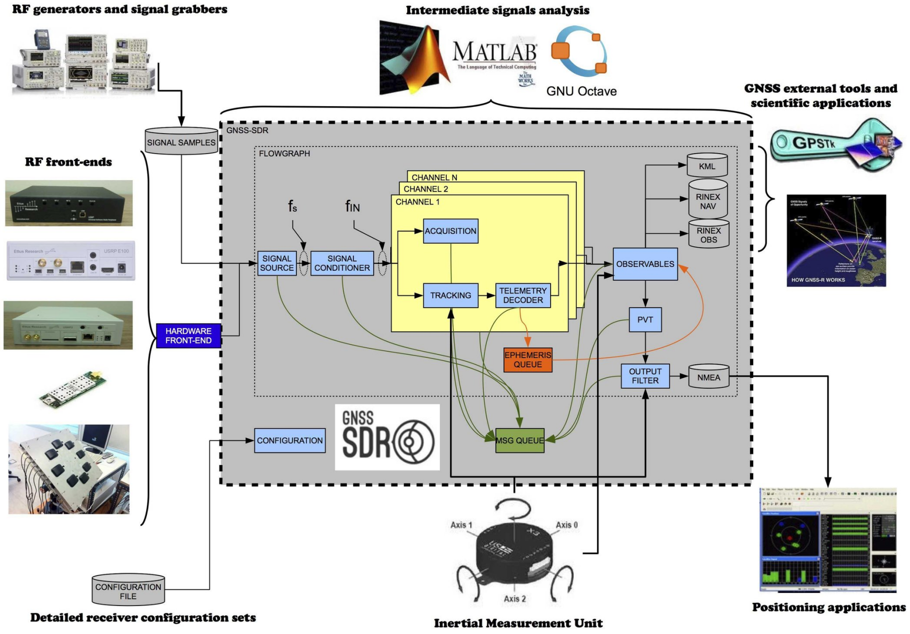


Figure 4.2: General structure of GNSS SDR [64].

4.2 Applications enabled by GNSS-SDR concept

4.2.1 Server-GNSS

In certain GNSS applications, a distinct separation between the GNSS front-end (which collects the RF samples) and the subsequent signal processing and navigation solution can be advantageous. These signal processing and navigation operations can be executed on a computer, located at a different location, and even at a different time (post-processing). This server and the GNSS front end may be connected via a data link, typically with a high B_W. On the server side, the GNSS SDR receiver is responsible for retrieving GNSS signals from one or more front ends and subsequently generating the precise position solution. This configuration presents several advantages. Firstly, the server potentially has access to assistance or differential correction data, PPP data such as precise orbits or clock corrections, and therefore enables the provision of a more accurate position solution in both realtime and post-processing. It can even perform high-sensitivity acquisition techniques, being able to provide a PVT solution in harsh environments.

As the data do not need to be processed in real time and can be stored, a GNSS server receiver is a device that periodically records short segments of the GNSS signal. These snapshots of signals are stored or transmitted to a server, which is responsible for the acquisition, tracking, computation, and visualization of the position solution. In particular, high-sensitivity signal processing algorithms are executed on the server, rather than on the mobile device itself. The difficulties arised in the transmision of this data. Some weapons tracking systems, for example, utilize an S-band link to transmit telemetry. In this case, it can also be used to send the captured signal from the vehicle to the ground server. On the server side, the GPS receiver can achieve a short Time To First Fix (TTFF) and exhibits the capability to track signals with high dynamics.

This captured signal can be a key tool in optimizing the performance parameters of the receiver, as it will have all the characteristics of the signal on board, and even if they are recorded with the oscillator that the receiver will later use, they also inherently have the behavior of it. With this signal, receiver parameters such as sensitivity, precision, and availability can be optimized or oscillator behavior can be analyzed. In fact, in the results section (Chapter 7) this is, in part, the approach followed. The exception is that the available material means do not allow the signal on board the rocket to be recorded at the moment, and only the simulated one.

Another example could be an asset tracking system, such as for containers or vehicles. Small segments of GNSS samples are periodically recorded and temporarily stored. Subsequently, these data are transferred to a PC or Web service running server-side radio software. This approach in [10] uses a hardware, *SnapperGPS* that can run for more than a year with minimal power consumption and is capable of recording 12 ms signal chunks at 4 MHz of BW and 1 bit resolution. With a memory of 512Mbits, one can store about 11000 snapshots in 21 $ Harware. Applications to wildlife tracking or LEO satellites [63] are being investigated in this regard. Moreover, depending on the duration of the sample snapshot, this approach can yield high positioning accuracy. In a related mass-market application, it is conceivable to use this technology to geolocate images captured by a digital camera. The GNSS data samples are then stored alongside the respective images.

4.3 Technology readiness

SDR technology, when applied to GNSS receivers, provides a high degree of flexibility, with a variety of benefits and challenges. Especially now with the new services being deployed, new signals and various capabilities under test of these GNSS constellations, the explored concept of *recording* samples can be used to be able to use these services later. Signal recording is especially effective in navigation warfare environments. It can allow later analisis of the types of spoofing and jamming suffered, or even the use of services such as OSNMA or CHIMERA for position validation or forensic studies of what has happened. Moreover, the highly adaptable architecture allows easy updates during its useful life, which could be particularly valuable in remote and harsh environments. In addition, this technology is a powerful tool for prototyping innovative signal processing algorithms and monitoring signal quality. SDR receivers require little or no hardware reconfiguration, reducing the overall effort of development, maintenance, and upgrades. SDRs also enable advanced simulations, generating high-fidelity, customizable, real-time GNSS signals that surpass the limitations of traditional signal generators. This comes with a drawback to security: It increases the availability of spoofers and their quality.

It can be said that nowdays the larger use of GNSS SDR related technology is actually in this field.

However, SDRs are not without limitations. In terms of performance, they are still behind. A recent study [45] comparing the SDR GNSS receivers found that the SDR offered a 3D-MRSE (3D Mean Radial Spherical Error) of 3.4 m in the single-point-positioning mode, which is lower than existing low-cost receivers as [156]. Other recent studies [149] evaluated GNSS-SDR for the Indian Regional Navigation Satellite System (IRNSS) signals using a low-cost RTL-SDR. They achieved 9.79m 3D-MRSE, but only 5 satellites were available. The cost of these systems is also larger, since they cannot take advantage of the classical ASIC receivers of scale economy in large quantities fabrication. Despite these challenges, the GNSS scientific community has made significant strides in advancing SDR technology, with numerous open-source projects aimed at addressing the current limitations in GNSS receiver testing. Regarding the future of SDR in GNSS Receivers, despite current limitations, there is still room for improvement in SDR positioning accuracy. The advent of open-source software and the growing demand for GNSS receiver customization have led to a surge in interest in SDR technology. This trend is likely to continue, with SDR technology playing a crucial role in the future of GNSS receivers [69], [113].

Front-end / Cost	**Tuning range**	**Bits**	**RF BW / Chipset**	**Ch**	**Clock Stability**	**Bus**	**BiasT**
USRP-B210 / 2200$	70MHz-6GHz	12	56MHz / AD9361	2	±75ppb (opt)	USB3.0	No
USRP-X310 / 9800$	DC-6GHz	12	160MHz / Daughter-Boards	2	±25ppb (opt)	PCIe (x4) /ETH	No
USRP-X410 / 30000$	1MHz-7.2GHz	12	400MHz / RF-SoC ZU28DR	4	±2.5ppm	PCIe (x8) /USB-C	No
BladeRF-Micro2.0 / 420$	47MHz-6GHz	12	28MHz / LMS6002D	2	±1ppm	USB3.0	Yes
LimeSDR-Mini2.0 / 300$	10MHz-3.8GHz	12	30.72MHz / LMS7002M	1	±2.5ppm	USB2.0	No
PocketSDR / 50$	GNSS L1 / L5	2	20MHz / MAX2771	2 or 4	±0.5ppm	USB2.0	Yes
SDRPlay RSPdx / 220$	1kHz-10GHz	14	10MHz / RSPdx	1 Rx	±10ppm	USB2.0	Yes
RTLSDR-R820 / 40$	24MHz-1.7GHz	8	2.8MHz / R820T2	1 Rx	±0.5ppm	USB2.0	Yes
ADALM-Pluto / 200$	325MHz-3.8GHz	12	20MHz (unofficial firmware) / AD9363	1	±25ppm	USB2.0	No
Hack-RF-One / 300$	1MHz-6GHz	8	20MHz / MAX2837-MAX5864	1	±0.5ppm (opt)	USB2.0	Yes
FreeSRP Alpha / 400$	70MHz-6GHz	12	56MHz / AD9364	1	±10ppm	USB3.0	No
GNSS Firehose / 800$	0.7-2.2GHz	8	50MHz / MAX2112	3 (L1, L2 L5)	±1ppm	GbE	Yes

Table 4.1: Comparison of SDR Front Ends usable for GNSS applications.

Chapter 5

GNSS on projectiles

As introduced in Chapter 4, the use of GNSS receivers on board this type of vehicle (rockets and guided projectiles) is the key for them to achieve their objective. Various projects have been carried out on this idea, on sounding rockets in [96], [6] and [107]; or on artillery rockets in [95]. At the moment there is no technologically mature alternative to estimate on-board the absolute position and speed of the ammunition with sufficient precision to guide and control it, although some other approaches are under investigation, such as the use of ground-based signals of opportunity [166].

However, there are several challenges for the use of onboard GNSS receivers in these vehicles:

- Legal/CoCom limitations.
- Mechanical resistance of the components and behavior even when they survive.
- Acquisition tuning.
- Tracking tuning.

The first will not be mentioned again throughout the thesis, as it is understood as a legal limitation, outside the scope of the thesis (for more details see [29]). Regarding the mechanical resistance of the components and their behavior in the dynamics of the ammunition, data will be collected in part to be able to explain the limitations or certain behaviors, especially in the results section in which data from real tests are collected. Finally, it is the objective of this chapter to detail how the acquisition and tracking processes of a GNSS signal must necessarily be modified in a receiver that is embarked on rocket- or projectile-type vehicles.

5.1 Acquisition tuning

The acquisition process involves searching for satellites within a 2D search space. This area of study is well documented, with [144] and [85] providing comprehensive overviews. Introducing the satellite acquisition process is important to understand the challenges faced by a GNSS receiver on board a projectile. In the GNSS satellite vehicle (SV) acquisition process, two unknown parameters must be determined, resulting in a 2D search space. These unknowns are the code delay and the frequency offset of the carrier signal received.

On the one hand, the code delay is influenced by the distance between the satellite and the receiver at any given time, as well as, to a lesser extent, by the Doppler shift, particularly if the shift is substantial [1].

The frequency offset, on the other hand, is affected by two factors: the relative speed between the satellite and the receiver, and the discrepancy between the nominal and actual frequency of the local oscillator [79].

As the uncertainty in each dimension increases, the search space expands, leading to longer search times. The range dimension for the C/A code is limited to 1.023 chips in total, while the range dimension of the Galileo E1 code is limited to 4.096 chips in total. Typically, the code phase is searched in increments of 1/2 chip, referred to as *code bin*.

In contrast, the Doppler dimension is only bounded by the radial velocity between the satellite and the receiver. As mentioned above, it depends on the dynamics of the receiver and the behavior of the receiver oscillator. The size of the *Doppler bin* in the search is approximately 2/(3T) Hz, where T represents the search integration time [85]. For instance, with a T of 1 ms (the minimum integration time for a C/A code), the Doppler search bin is approximately 667 Hz, whereas for a T of 20 ms (the maximum acquisition coherent integration time possible for GPS L1), the Doppler bin is about 34 Hz. The lower the expected signal-to-noise ratio (expressed as a lower C/N_0) of the signal received from the satellite vehicle (SV), the longer the integration time required, which in turn increases the search time.

5.1.1 The search pattern

As explained in [85], the search pattern typically involves scanning in the direction (code) of the range from early to late, at a constant Doppler frequency. This approach helps mitigate the effects of multipath interference, as the direct signal typically arrives before the multipath signals. In the Doppler bin direction, the search pattern usually begins at the mean value of the Doppler uncertainty (or zero Doppler if the actual line-of-sight velocity estimate is unknown). The search then progresses symmetrically, one Doppler bin at a time, on either side of this value until the entire range is covered.

It is also noted in [85] that the autocorrelation and cross-correlation sidelobes of the C/A code can lead to false signal detections if these sidelobes are sufficiently strong. These sidelobes tend to become more pronounced as the integration time is reduced. However, this will lead to a particular problem in high-dynamics receivers, as the integration time should be kept as low as possible [161]. This, with the Doppler range to be explored in the receiver acquisition process on board a projectile, will be explored as follows.

---

[1] The influence over code delay by the Doppler shift, in the L1 band, is divided by a factor of 1540 relative to the carrier Doppler. This effect is typically significant only for LEO satellites, which travel at speeds of up to 7 km/s, resulting in carrier Doppler shifts of up to 50 kHz [11]

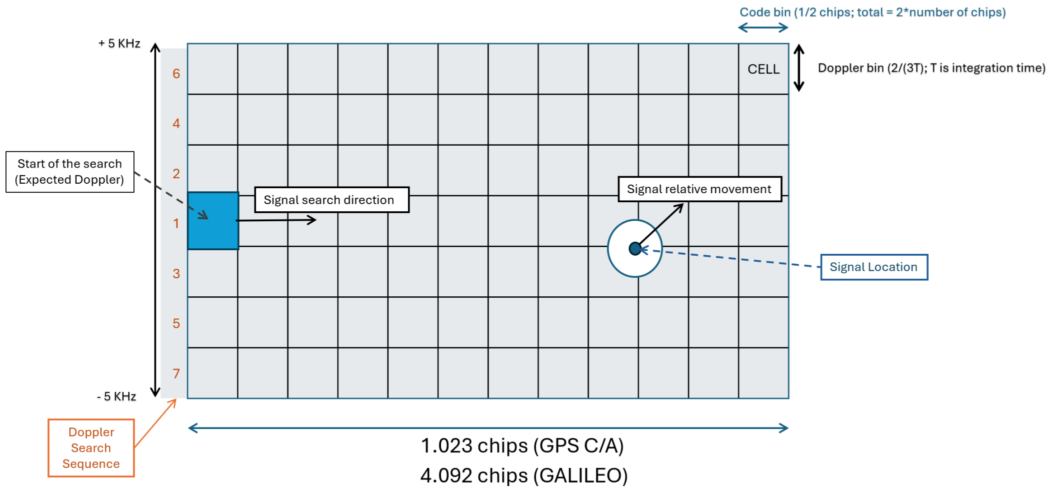


Figure 5.1: Two-dimensional search pattern for GNSS signal acquisition, adapted from [85].

5.1.2 Effects in Rocket flight

The relative importance of these effects in a typical scenario is as follows:

- TCXO frequency drift of ±1 ppm, resulting in an additional 1.575 kHz search range.
- Satellite motion, in the worst-case scenario (after calculations), contributes ±4.2 kHz.
- The receiver motion, for example, in a car moving at 27.8 m/s, adds approximately 150 Hz.

This leads to a total search range of approximately $\pm 5\ kHz$ for a receiver that is under *everyday* motion.

However, in the case of a rocket, additional factors need to be considered:

- Receiver motion: For a projectile traveling at 500 m/s at the muzzle, this adds approximately 2.63 kHz.
- Angular receiver motion: For a projectile rotating at 100 revolutions per second (100 rev/s) with an angular velocity of about 628.32 rad/s, and considering a diameter of 155 mm, this corresponds to a linear speed of 48.7 m/s, adding around 250 Hz.
- The effect of acceleration on the TCXO (temperature-compensated crystal oscillator), both linear and angular, if it is not precisely aligned with the axis of rotation of the projectile:
  - For linear acceleration, a rocket might experience around 100g during the 2-second propulsion phase, followed by a deceleration of around 20g when propulsion ends.
  - For angular acceleration, which is more significant, a rotation speed of 100 rev/s at approximately 3 cm from the axis would result in around 1200g.
- With a typical sensitivity of 1.5 ppb/g, this would add approximately 2.85 kHz to the search range [101] or 0.38 kHz if a Low g-Sensitivity clock is used (as [100] with 0.2

ppb/g).

This results in a total Doppler search range of around ±10 kHz.

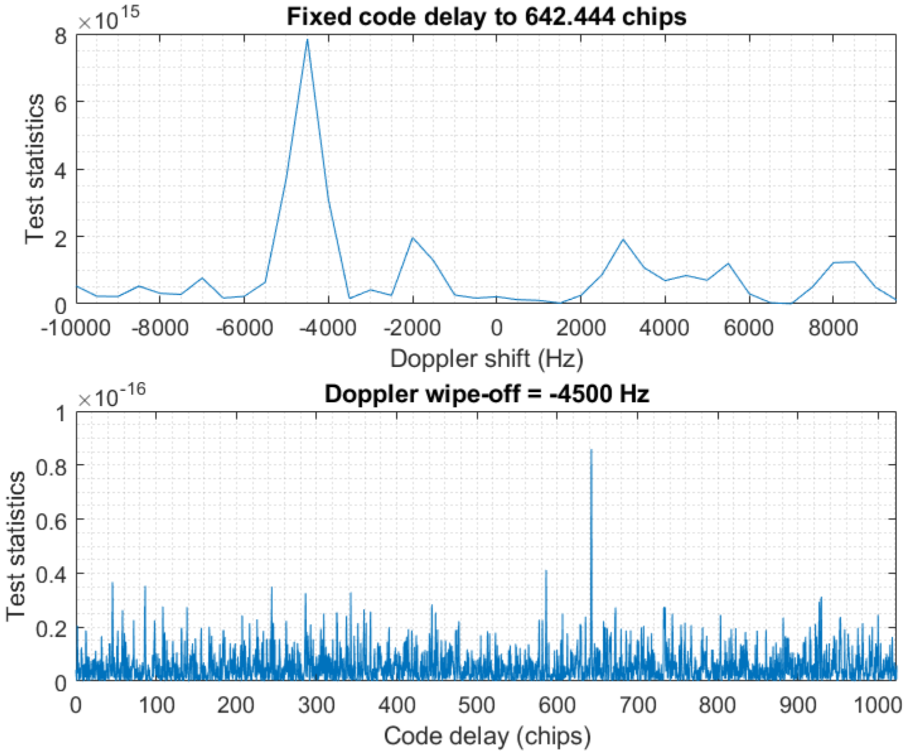


Figure 5.2: Doppler shift and code delay for a positive acquisition of a GPS L1 C/A signal.

The effect on the code is approximately 6.5 Hz, which, for an integration time of 10 ms, is much less than 0.3 chips (0.5 chips would result in a loss of about -6 dB).

To mitigate these effects, it is essential to keep acquisition integration time as short as possible, even if this comes at the cost of receiver sensitivity. Typically, this would be 1 ms for GPS and 4 ms for Galileo. It is also important to note that due to the nature of the trajectories of these munitions (usually reaching altitudes of 8-12 km), there should be no significant issues with maintaining satellite visibility. Therefore, there is no need for a large sensitivity.

5.2 Tracking Architecture

The typical trade-off in loop tracking design is bandwidth versus dynamic performance: the effects of noise increase with increasing loop bandwidth, whereas dynamic tracking errors increase with decreasing loop bandwidth [90]. Some interesting graphics of this trade-off, as a function of bandwidth and dynamic stress, for TCXO and OCXO can be found in 12.3.7 of [105].

Moreover, in [85] an analysis of how to relate the values in Table 5.1 and maximum dynamics is described. From this analysis, Figures 5.5(a) and 5.5(b) are derived.

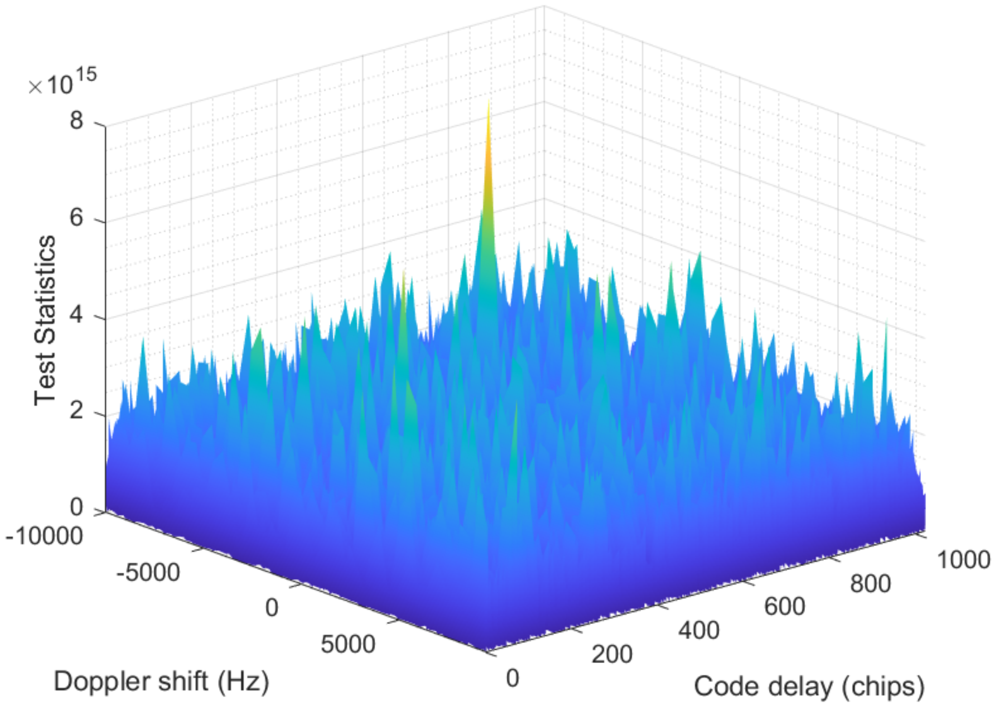


Figure 5.3: Example of a GPS L1 C/A signal positive acquisition in a projectile scenario.

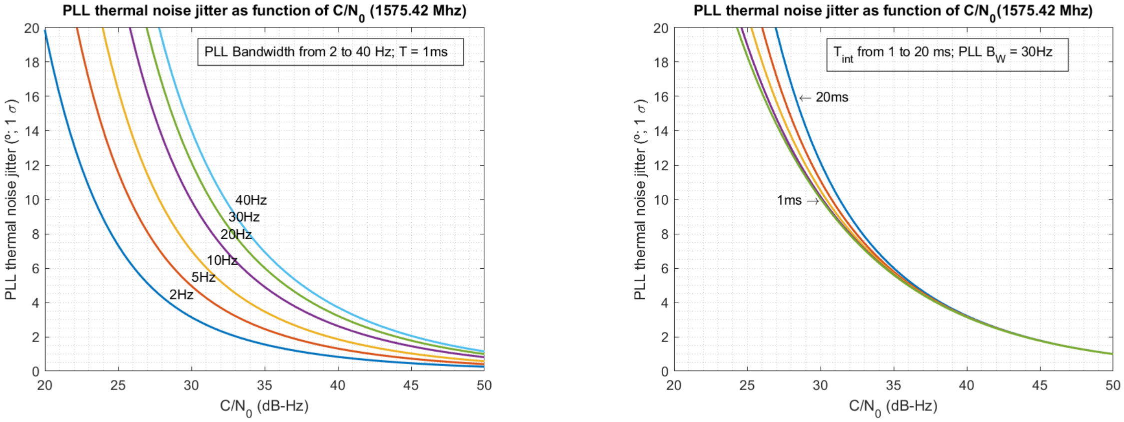
As can be seen, the theoretical stationary state error of the loop depends on the loop bandwidth and order, whereas the order fix the sensitivity to velocity, acceleration, or jerk (Table 5.1).

PLL-DLL: For carrier signal-to-noise ratios commonly encountered by GNSS receivers on aircraft and missiles, loop bandwidths need to be so narrow that reliable carrier phase tracking can only be achieved for acceleration not exceeding 5 g and jerk (the derivative of acceleration) not exceeding 5 g/s.

Nowadays, the loop structure known as FLL-assisted-PLL is very often adopted for GNSS receivers. Rather than using a single loop, it consists of a PLL and an FLL in a coupled mode to reduce locking times and avoid false locks. The advantages of adding FLL to track spread spectrum signals in dynamic environments have already been studied in [31] or [133], [134].

In the case of rockets, the critical moments are the engine turn-on and turn-off that produce large changes in acceleration.

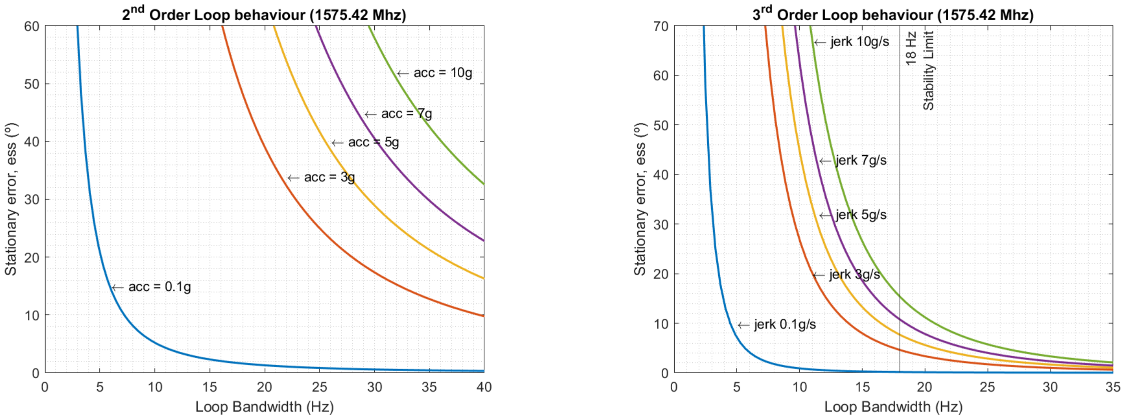
With FLL: The receiver can operate under much higher dynamics: A breadboard receiver has been built that demonstrated reliable tracking with simulated trajectories corresponding to 150 g acceleration and 150 g/s jerk.



(a) PLL thermal noise, for several PLL Bandwidths, and integration time of 1ms.

(b) PLL thermal noise, for several integration times and a fixed bandwidth of 30 Hz.

Figure 5.4: PLL thermal noise vs C/N_0.



(a) 2^{nd} Order Loop dynamics.

(b) 3^{rd} Order Loop dynamics.

Figure 5.5: PLL stationary error (ess) vs C/N_0.

5.2.1 Tracking loops tuning

The acquisition process only provides an initial estimate of the Doppler frequency (Δf) and the code delay (τ). Signal tracking is used to generate precise replicas of the incoming code and carrier. The local code and carrier can be generated correctly only if the signal parameters are correctly tracked. The tracking loops are endorsed with this task: refine the Δf and τ, follow their values during the trajectory, under receiver dynamics, and demodulate the navigation message from the satellite vehicle (SV). If this task is successfully performed, then the correlation of the local copy PRN code with the signals is capable of pulling those signals out of the noise (*de-spreading* the signal).

The tracking module needs to generate two replicas: one for the carrier and one for the code, in order to accurately track and demodulate the signal from a single satellite. The objective is to keep the phase of its replica code in maximum correlation with the desired SV code.

Table 5.1: Loop Filter Characteristics, adapted from [85].

Loop Order	**Filter relation between B_w and ω_0**	**Steady State Error (ess)**	**Characteristics**
First	$B_w = 0.25\omega_0$	$\frac{(dR/dt)}{\omega_0}$	Sensitive to velocity stress. Unconditionally stable at all bandwidths
Second	$B_w = 0.53\omega_0$	$\frac{(d^2R/dt^2)}{\omega_0^2}$	Sensitive to acceleration stress. Unconditionally stable at all bandwidths
Third	$B_w = 0.7845\omega_0$	$\frac{(d^3R/dt^3)}{\omega_0^3}$	Sensitive to jerk stress. Stable only at bandwidths $< 18Hz$ [163]

dR/dt = maximum LOS dynamics (deg $/s$).

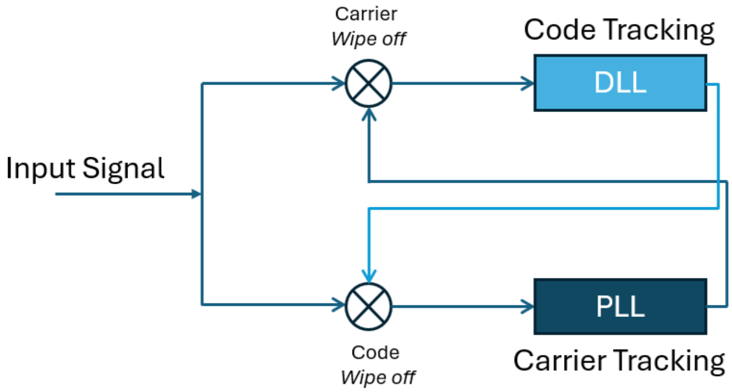


Figure 5.6: High level schematic of the interaction between DLL and PLL tracking loops in GNSS signal processing.

The receivers use two separate loops for the local carrier and the code. The two loops are coupled in the sense that the DLL (code) requires precise carrier wipe-off to operate correctly, and conversely, the PLL (carrier) also requires precise code wipe-off to operate correctly (Figure 5.6).

If the receiver cannot adjust its replica carrier signal to match the frequency of the target SV carrier, then the signal correlation process in the range dimension is significantly weakened due to the frequency response roll-off characteristic of the GPS receiver. This has the consequence that the receiver cannot use the signal from the SV. If the signal was successfully acquired because the SV code and frequency were successfully replicated during the search process, the receiver will subsequently lose track of the SV frequency and subsequently will also lose the code track. This is especially relevant in high-dynamic scenarios, as receiver Doppler shift with respect to SV is relevant, as introduced previously (through LOS relative dynamics, or even receiver's oscillator). This Doppler shift can cause the receiver to lose lock and, as explained, make that channel unable to follow the desired SV signal.

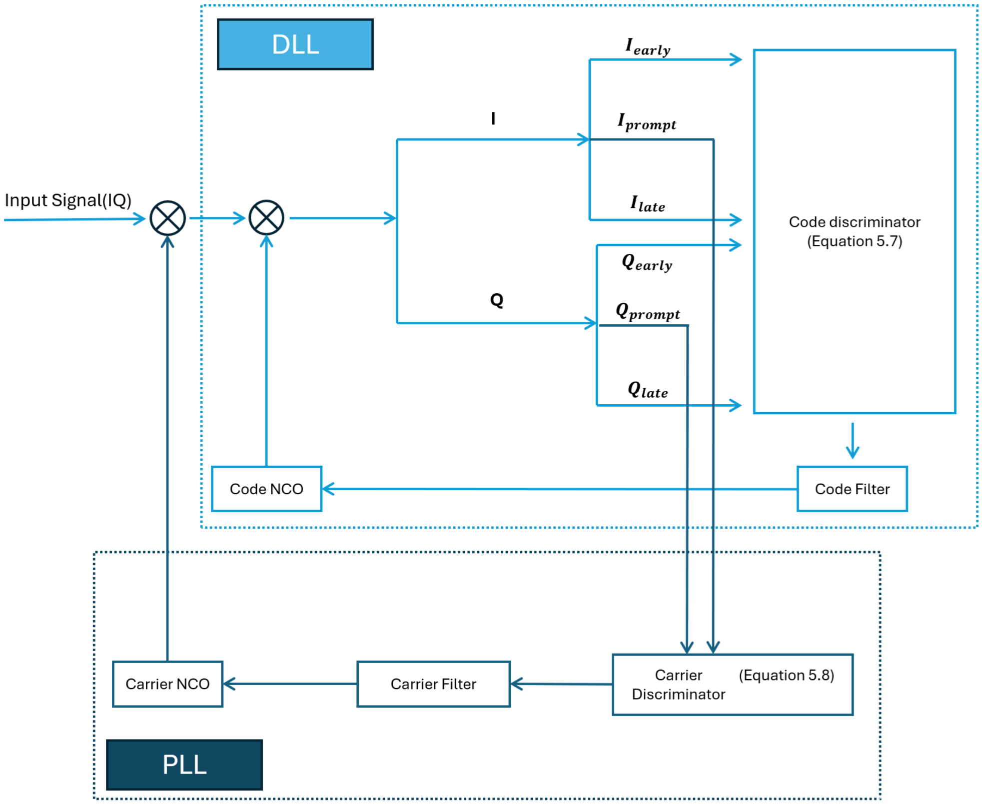


Figure 5.7: Detailed schematic of the operation of DLL and PLL tracking loops in GNSS signal processing.

Generally, tracking employs three correlators: one is placed at the prompt or on-time correlation point for carrier tracking, while the other two are symmetrically placed early and late relative to the prompt phase for code tracking (Figure 5.7). In Galileo five correlators are used, instead of three. This is to avoid false locks on the side lobes of the BOC [7]. Modern receivers may even use massive multiple correlators with the aim of speeding up the search process or for robust code tracking [1].

5.3 High Dynamics architectures for GNSS tracking

The main signal processing challenge for a GNSS receiver in a scenario of very demanding dynamics is the receiver's ability to follow the rapid variations in the synchronization parameters of the received signals. Existing GNSS receivers are usually based on two-step tracking loop structures known as DLL/PLL, where one tracking loop is dedicated to estimate the time delay evolution of the received signal, while another loop is dedicated to follow the evolution of the carrier phase. These structures become unstable at high speeds [22], [77]. For example, a commercial receiver is not only limited by design to a maximum speed of 1900 km/h and an altitude of 18 km, but it may also lose the ability to calculate the PVT solution above accelerations as low as $4g$ [156].

5.3.1 Tracking Loops based on PLL-DLL

The PLL senses the phase difference between the incoming and local signals. The phase rate (frequency) of the local replica is increased or decreased in order to recover the initial phase difference. The phase error is driven to zero only if both phase and frequency are recovered. A detailed diagram is shown in Figure 5.7. The components of this diagram are summarized as follows:

Code discriminators

The designer has many choices for the code discriminator function $\Delta_c(\cdot)$, see [82], [83], [85] for a discussion of possible options and associated trade-offs. Considering the correlator output i as $y_i[k] = y_{i,I}[k] + jy_{i,Q}[k]$, generic code discriminator functions can be expressed, depending on the availability of $\hat{\phi}_0[k]$,

$$\Delta_c^{(\text{non-coherent})}[k] = \sum_{i=-\frac{N_c-1}{2}}^{\frac{N_c-1}{2}} \omega_i \left(y_{i,I}^2[k] + y_{i,Q}^2[k] \right) \ , \tag{5.1}$$

in case of data-modulated channels or before the steady-state of $\hat{\phi}[k]$ estimation,

$$\Delta_c^{(\text{coherent})}[k] = \sum_{i=-\frac{N_c-1}{2}}^{\frac{N_c-1}{2}} \omega_i y_{i,I}[k] \ , \tag{5.2}$$

in case of the availability of a reliable estimation of $\hat{\phi}[k]$.

The number of correlators N_c, their spacing ϵ_i and the weights in Equations (5.1) or (5.2) are design parameters. Design criteria can be output bias and variance, sensitivity, operating range, unambiguity, and computational cost [5], [80].

The implementation presented in this work uses the noncoherent Early minus Late envelope normalized discriminator, which is a particularization of Equation (5.1). For BPSK signals,

$$\Delta_c[k] = \frac{y_{intercept} - \text{slope} \cdot \epsilon}{\text{slope}} \cdot \frac{|E[k]| - |L[k]|}{|E[k]| + |L[k]|} \ , \tag{5.3}$$

where:

- $y_{intercept}$ is the interception point of the correlation function on the y-axis,
- slope is the slope of the correlation function,

- ϵ is the Early-to-Prompt (or Prompt-to-Late) spacing, normalized by the chip period,
- $|E[k]| = \sqrt{E_I[k]^2 + E_Q[k]^2}$ is the magnitude of the Early correlator output,
- $|L[k]| = \sqrt{L_I[k]^2 + L_Q[k]^2}$ is the magnitude of the Late correlator output.

Phase discriminators

For the phase discriminator function $\Delta_p(\cdot)$, several options can be applied depending on the characteristics of the GNSS signal in which the channel is operating. In channels with the presence of navigation data bits or a secondary spreading code, the phase discriminator must be insensitive to $\pm\pi$ rad phase jumps. This is the case of the Costas loop two-quadrant discriminator [85],

$$\Delta_p^{(Costas)}[k] = \text{atan}\left(\frac{P_Q[k]}{P_I[k]}\right) \ . \tag{5.4}$$

However, in data-less channels, or when secondary code/bit synchronization is already achieved, the receiver can switch to other types of discriminator with improved accuracy, such as the four-quadrant arc tangent discriminator, which is known to be the Maximum Likelihood Estimator (MLE) [33], [86]:

$$\Delta_p^{(atan2)}[k] = \text{atan2}\left(P_Q[k], P_I[k]\right) \ . \tag{5.5}$$

Grouping the code and carrier phase discriminators in a vector, we can define,

$$\mathbf{z} = \left[\Delta_c(\cdot), \Delta_p(\cdot)\right]^T \ , \tag{5.6}$$

where Δ_c is the code discriminator output, defined as in Equation 5.3 for BPSK signals, and Δ_p the phase discriminator output. Those measurements can be related to the states by accounting for frequency rate effects during the integration interval (taking the third term of the Taylor expansion),

$$\Delta_{c,k}(\cdot) = \Delta\tau_k - \frac{\beta T}{2}\Delta f_{d,k} + \frac{\beta T^2}{6}\Delta\alpha_k + w_{c,k} \ , \tag{5.7}$$

$$\Delta_{p,k}(\cdot) = \Delta\phi_k - \pi T \Delta f_{d,k} + \frac{\pi T^2}{3}\Delta\alpha_k + w_{p,k} \ . \tag{5.8}$$

Tunning of a PLL-DLL for High dynamics

From Figure 5.4(b), an interesting conclusion can be drawn: If C/N_0 is sufficiently high $(> 35, \text{dB-Hz})$, then the jitter produced in the PLL is practically independent of the integration time and will only depend on the chosen bandwidth. However, as will be practically demonstrated, this is not entirely true (see Figure 5.8(b) and Figure 5.13(b), where only the PLL is used, in a static situation, with the same bandwidths, but one uses $T = 1, \text{ms}$ and the other $T = 20, \mathrm{ms}$). Nevertheless, in the current analysis, we will not use integration times greater than $1, \mathrm{ms}$ for two reasons:

- In the case of rocket and projectile flights, as introduced, the sky visibility is good, and the C/N_0 is high (typically $> 40, \text{dB-Hz}$).
- When long integration times are used, effects not considered in this analysis, such as frequency variations of the local oscillator due to thermal noise, vibrations, or acceleration, negatively impact the tracking loop's ability and can cause it to lose synchronization.

On the other hand, the use of different bandwidths with the same C/N_0 causes significant differences in jitter (Figure 5.4(a)). Here, the difference is not small, even at $C/N_0 > 40, \text{dB-Hz}$, although for an increase in B_W from $2, \mathrm{Hz}$ to $40, \mathrm{Hz}$, σ increases from 1° to 4°. In other words, at the signal levels available in the study scenario, increasing B_W results in performance loss, but it is not prohibitive (using $T_{\text{int}} = 1, \mathrm{ms}$).

Another aspect to consider is the behavior of loops under different dynamics depending on the chosen filter order (Figures 5.5(a) and 5.5(b)). In this analysis, two aspects must be considered: first, a loop of 2$^{\text{nd}}$ order will only be capable of tracking constant acceleration dynamics (zero jerk). This tracking will have a steady-state error (ess) that will be greater the higher the acceleration (Figure 5.5(a)). With a B_W of $30, \mathrm{Hz}$ at an acceleration of $5g$, we will have an ess of about 29°. Conversely, if we use a loop of 3$^{\text{rd}}$ order, then the ess will be zero for any acceleration, but attention must be paid to the jerk. It is important to note that its stability is no longer unconditional [163]. Stability is not guaranteed above 18Hz . For a B_W of 18Hz and a jerk of $10g/s$, the ess will be 15°.

Behavior of the DLL-PLL in static and launch conditions

The figure 5.8(a) and the following ones are organized as follows: at the top left is the constellation, where I is plotted against Q. To its right are the information bits encoded in the PRN, with the information in I shown in blue and the information in Q shown in orange. Below, the values of the magnitudes of the correlations that will be introduced into the code discriminators are represented. Finally, in the bottom left are the values of the PLL discriminator (Equation 5.4) and the DLL discriminator (Equation 5.3), with and without filtering applied.

Static: After the initial acquisition, the loop takes 1.5 seconds to stabilize (see Figure 5.8(a)). During this time, the signal energy oscillates between the I and Q channels, the constellation rotates because of the unresolved phase, and the Doppler measurement has a very high error (see Figure 5.8(b)). After this period, the tracking channel stabilizes and has a smaller error. This error is further reduced when, after 12 seconds (bit synchronization), narrower B_W (bandwidths) are used.

Launch: In the case of a launch, the initial level of jerk causes the tracking channel to lose lock (see Figure 5.9(a)). The Doppler value recovered from the SV (Space Vehicle) signal during the launch is not validly estimated (see Figure 5.9(b)), and the loop loses the ability to track the signal for about 15 seconds, until the jerk decreases (the rocket motor has shut down) and the acceleration varies smoothly, although it remains high. From then on, the behavior is similar to what occurs after acquisition in static mode, with a transient period followed by steady state operation and improved performance using narrow B_W.

When the PLL-DLL tracking structure works as expected, the phase error is driven to zero and the signal power is only present in the I component (Figures 5.8(a) and 5.9(a)). This will be compared with the FLL case in Section 5.3.2.

5.3.2 Tracking Loops based on PLL-DLL-FLL

The classical solution for dealing with high-dynamics GNSS signal tracking is based on the expansion of the usual dynamics model (considering carrier phase and Doppler shift), from which the DLL/PLL architecture is derived, by another one that includes the rate of change of the Doppler shift [28], [163]. Physically, this corresponds to the estimated velocity, acceleration and jerk of the receiver (Figure 5.10).

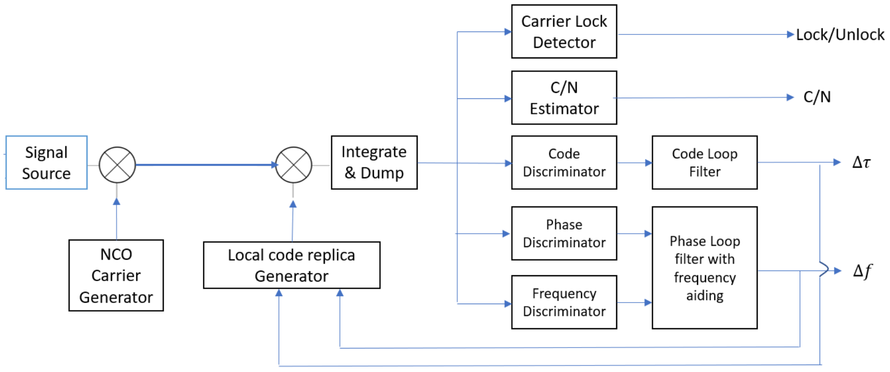


Figure 5.10: Classical tracking loop architecture for high dynamic scenarios.

This solution, combined with adaptive management of the frequency tracking loop (FLL) and its assistance to the phase tracking loop (PLL), has demonstrated its ability to keep track of signals received with high dynamics [12], [145], although important issues can be found in some scenarios [108].

The FLL differs from the PLL in frequency discriminator and in the presence of an additional integrator.

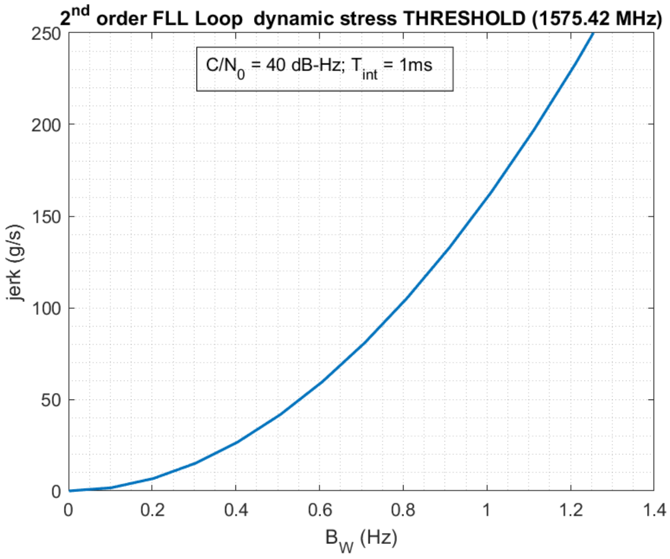


Figure 5.11: Jerk stress thresholds for second-order FLL at T_{int} of 1ms and C/N_0 of 40 $dB-Hz$.

Figure 5.12(a) illustrates the FLL thermal noise tracking jitter and tracking thresholds, assuming a second-order loop with 10 g/s jerk dynamics with different noise bandwidths and an integration time of 1 ms. Figure 5.11 illustrates the jerk stress thresholds for a second-order FLL as a function of the noise bandwidth. Comparing the thresholds in Figure 5.12(b) for the second-order FLL with those in Figure 5.5(b) for a third-order PLL, notice that the FLL has much better dynamic stress performance than the PLL at the same noise bandwidths. For example, at 10 Hz and 35 dB-Hz, the FLL can tolerate up to 240 g/s while the PLL can only tolerate up to 4 g/s. This comparison reinforces the earlier statements that a robust GPS receiver design will use an FLL as a backup to the PLL during initial loop closure and during high dynamic stress with loss of phase lock but will revert to pure PLL for the steady-state low to moderate dynamics in order to produce the highest accuracy carrier Doppler phase measurements.

If the initial frequency estimate (from the acquisition block) is not sufficiently precise, the PLL can fail to lock the signal phase and frequency. The FLL can usually track higher dynamics and can pull the signal in frequency-lock conditions more easily than a PLL. The receiver can start tracking operations using an FLL and progressively move to a PLL with different bandwidths and integration times.

When using the FLL, a residual phase is present and the signal power is split between the I and Q components.

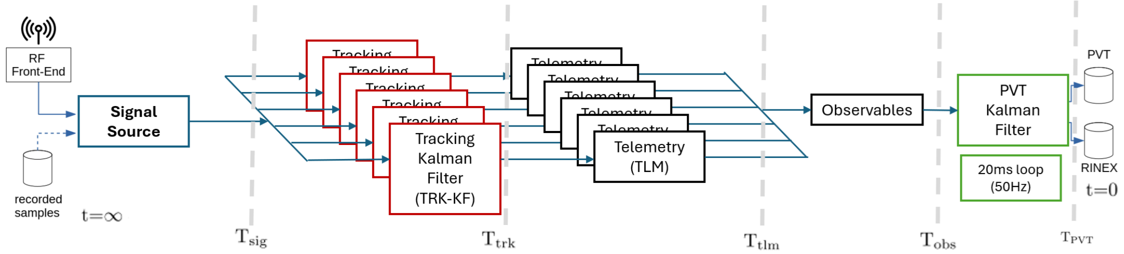


Figure 5.15: Kalman-Filter tracking implementation, high level block diagram.

5.3.3 Kalman Filter tracking

An evolution of the DLL/PLL channel tracking loops leads to the inclusion of a Kalman Filter (KF) from the satellite channel in the loops [164], called *TRK KF* (Figure 5.15). Feedback information coming from the PVT block may or may not be present, as detailed in the next section. An interesting and comprehensive comparison with the traditional PLL-DLL architecture can be found in [158].

The Kalman-filter states on those *TRK KF* blocks can be described as follows:

$$\mathbf{x}_k = [\Delta\tau_k, \Delta\phi_k, \Delta f_{d,k}, \Delta\alpha_k]^T \ , \tag{5.9}$$

where $\Delta\tau_k = \tau_k - \hat{\tau}_k$, in chips units, and $\Delta\phi_k = \phi_k - \hat{\phi}_k$, in radians, are the code delay and carrier phase errors, respectively, $\Delta f_{d,k}$, in Hz, is the residual Doppler frequency shift, and $\Delta\alpha_k$, in Hz/s, is the Doppler frequency rate correction. Their evolution can be modelled as follows,

$$\Delta\tau_k = \Delta\tau_{k-1} + \beta T \Delta f_{d,k-1} + \frac{\beta T^2}{2}\Delta\alpha_{k-1} + v_{\tau,k} \ , \tag{5.10}$$

$$\Delta\phi_k = \Delta\phi_{k-1} + 2\pi T \Delta f_{d,k} + \pi T^2 \Delta\alpha_k + v_{\phi,k} \ , \tag{5.11}$$

$$\Delta f_{d,k} = \Delta f_{d,k-1} + T\Delta\alpha_{k-1} + v_{f_d,k} \ , \tag{5.12}$$

$$\Delta\alpha_k = \Delta\alpha_{k-1} + v_{\alpha,k-1} \ , \tag{5.13}$$

where $\beta = \frac{r_c}{f_L^{(\mathcal{B})}}$ represents the factor used to convert cycles in units of code chips, T is the integration time and v are the noises.

and hence the whole system can be expressed,

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_k \ , \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \ , \tag{5.14}$$

where x_k is the state at the current time, x_{k-1} is the state at the previous time, F is the state transition matrix, or *system* matrix, and v_k is noise.

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \beta T & \frac{\beta T^2}{2} \\ 0 & 1 & 2\pi T & \pi T^2 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} , \tag{5.15}$$

The noise covariance matrix of the dynamics model $\mathbf{Q}_k$ can be written,

$$\mathbf{Q}_k = \begin{bmatrix} \sigma^2_{\Delta\tau,k} & 0 & 0 & 0 \\ 0 & \sigma^2_{\Delta\phi,k} & 0 & 0 \\ 0 & 0 & \sigma^2_{\Delta f,k} & 0 \\ 0 & 0 & 0 & \sigma^2_{\Delta\alpha,k} \end{bmatrix} . \tag{5.16}$$

where the diagonal is filled with the covariances of the noises v.

Considering the code and carrier phase discriminators (Equations 5.7 and 5.8), we can write the measurement matrix H,

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & -\frac{\beta T}{2} & \frac{\beta T^2}{6} \\ 0 & 1 & -\pi T & \frac{\pi T^2}{3} \end{bmatrix} , \tag{5.17}$$

and the filter measurements vector z_k,

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R_k}) , \tag{5.18}$$

$$\mathbf{R}_k = \begin{bmatrix} \hat{\sigma}^2_{\Delta_{c,k}} & 0 \\ 0 & \hat{\sigma}^2_{\Delta_{p_k}} \end{bmatrix} , \tag{5.19}$$

where w_k represents the noise in the measurements, i.e the discriminators. The variance of this discriminators output can be approximated by (based on the recent work in [150]),

$$\hat{\sigma}^2_{\Delta_{c,k}} \approx \frac{\epsilon}{4(C/N_0)_k T_c M} \left(1 + \frac{1}{(C/N_0)_k T_c \left(1 - \frac{\epsilon}{2}\right)} \right) , \tag{5.20}$$

$$\hat{\sigma}^2_{\Delta_{p,k}} \approx \frac{1 + 2(C/N_0)_k T_c}{4M\left((C/N_0)_k T_c\right)^2} \left(1 + \frac{1 + 2(C/N_0)_k T_c}{M\left((C/N_0)_k T_c\right)^2} \right) . \tag{5.21}$$

Based on those definitions, a Kalman-Filter can be defined. Starting from the initialization,

$$\mathbf{x}_{0|0} = \begin{bmatrix} \hat{\tau}_{acq} & 0 & \hat{f}_{d,acq} & 0 \end{bmatrix}^T , \tag{5.22}$$

$$\mathbf{P}_{0|0} = \begin{bmatrix} \sigma^2_{\tau,0} & 0 & 0 & 0 \\ 0 & \sigma^2_{\phi,0} & 0 & 0 \\ 0 & 0 & \sigma^2_{f,0} & 0 \\ 0 & 0 & 0 & \sigma^2_{\alpha,0} \end{bmatrix} , \tag{5.23}$$

$$\mathbf{Q}_0 = \begin{bmatrix} \sigma^2_{\Delta\tau,0} & 0 & 0 & 0 \\ 0 & \sigma^2_{\Delta\phi,0} & 0 & 0 \\ 0 & 0 & \sigma^2_{\Delta f,0} & 0 \\ 0 & 0 & 0 & \sigma^2_{\Delta\alpha,0} \end{bmatrix} , \tag{5.24}$$

$$\mathbf{R}_0 = \begin{bmatrix} \hat{\sigma}^2_{\Delta_{c,0}} & 0 \\ 0 & \hat{\sigma}^2_{\Delta_{p_0}} \end{bmatrix} , \tag{5.25}$$

the time update (prediction) step,

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1} , \tag{5.26}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q}_k . \tag{5.27}$$

and the measurement update (estimation),

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T \left(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}_k\right)^{-1} , \tag{5.28}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}\right) , \tag{5.29}$$

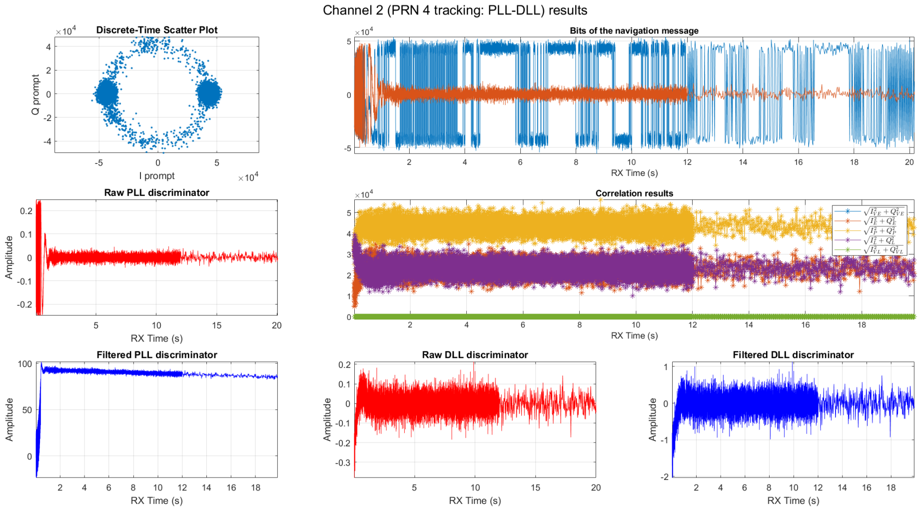
$$\mathbf{P}_{k|k} = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}\right)\mathbf{P}_{k|k-1} . \tag{5.30}$$

Since GNSS receivers used to apply different integration times before and after bit synchronization, this can be incorporated into the filter by doing the,

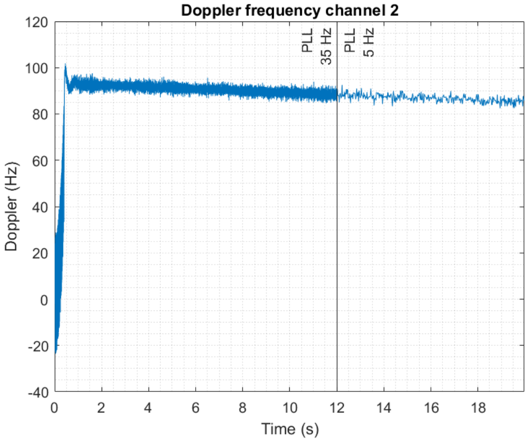
$$\mathbf{P}_{k+M|k} = \mathbf{F}^M\mathbf{P}_{k|k}\left(\mathbf{F}^M\right)^T + \sum_{m=0}^{M-1}\mathbf{F}^m\mathbf{Q}_k\left(\mathbf{F}^m\right)^T , \tag{5.31}$$

$$\mathbf{F}^M = \begin{bmatrix} 1 & 0 & \beta T_i & \frac{\beta T_i^2}{2} \\ 0 & 1 & 2\pi T_i & \pi T_i^2 \\ 0 & 0 & 1 & T_i \\ 0 & 0 & 0 & 1 \end{bmatrix}^M = \begin{bmatrix} 1 & 0 & \beta M T & \frac{\beta M^2 T^2}{2} \\ 0 & 1 & 2\pi T_i M & \pi M^2 T_i^2 \\ 0 & 0 & 1 & T_i M \\ 0 & 0 & 0 & 1 \end{bmatrix} . \tag{5.32}$$

assuming an integration time of T_i before bit synchronization, and MT_i after the preamble of the navigation message has been detected. The values used are $M = 1$ and $T = 1ms$, which leads to a coherent integration time of 1ms. This time is the minimum for GPS C/A signal [85]. If Galileo is to be used this time increases up to, then this time increases up to 4ms [41]. On the other hand, different approaches to the C/N_0 estimation can be performed. In this work the classical one is used, although a faster approach can be found in [137].

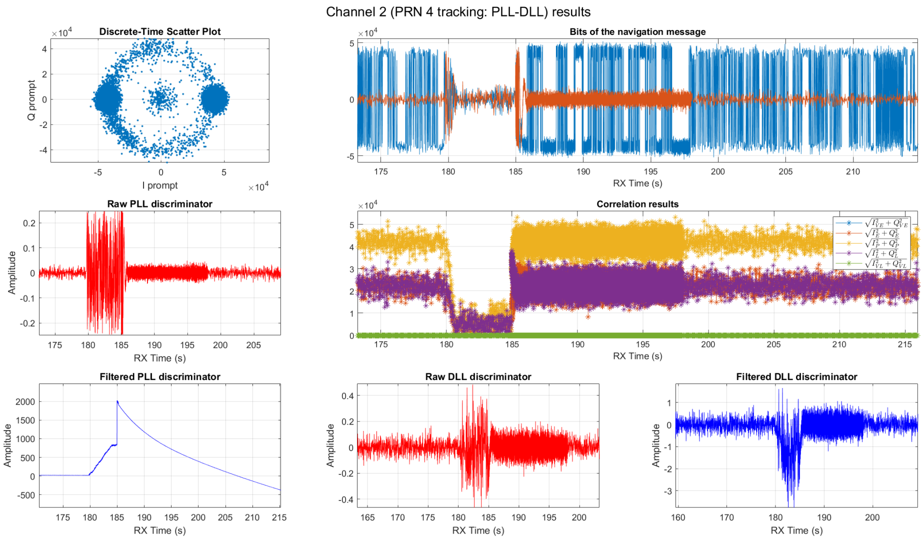


(a) PLL-DLL tracking.

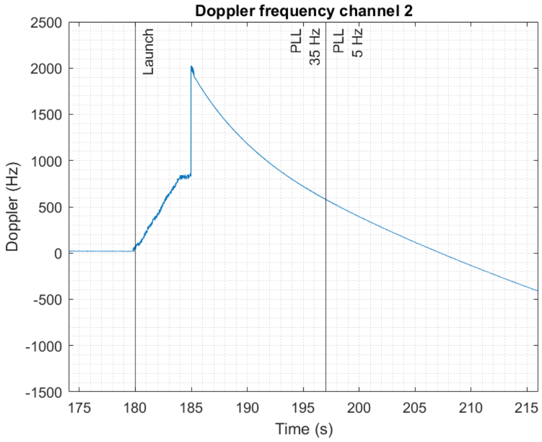


(b) Doppler shift of SV 4 recovered with PLL-DLL tracking.

Figure 5.8: Detailed behavior of PLL-DLL tracking after an acquisition in static position.

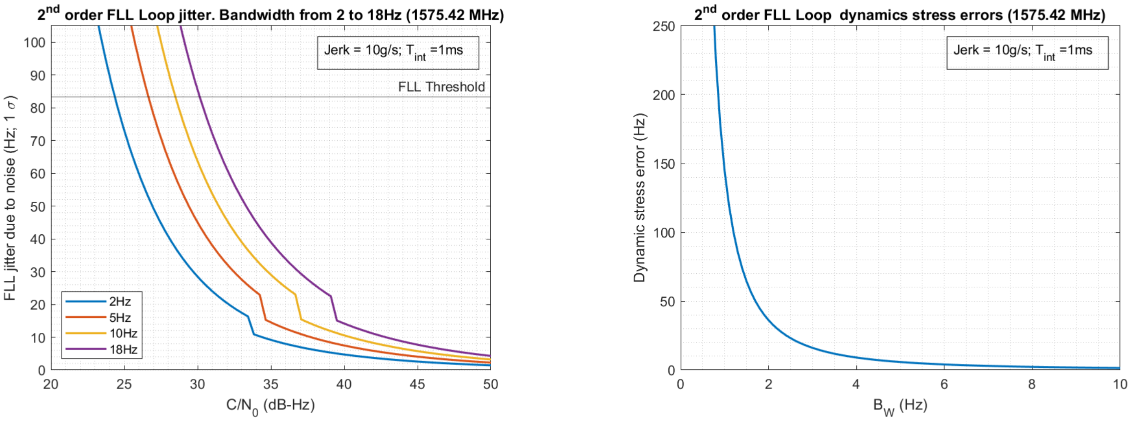


(a) PLL-DLL tracking, after launch event.



(b) Doppler shift of SV 4 recovered with PLL-DLL tracking, after launch event.

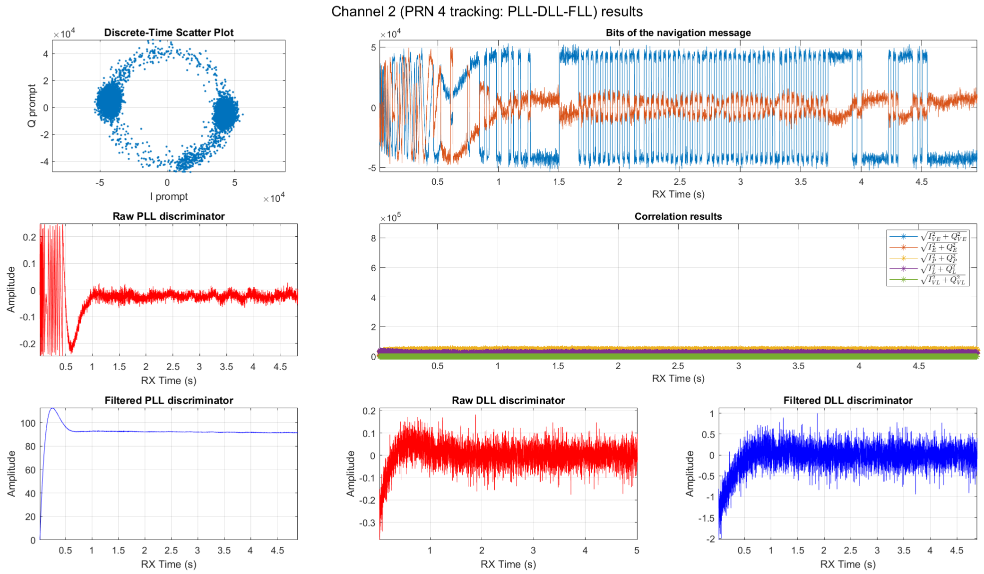
Figure 5.9: Detailed behavior of PLL-DLL tracking with re-acquisition after launch event.



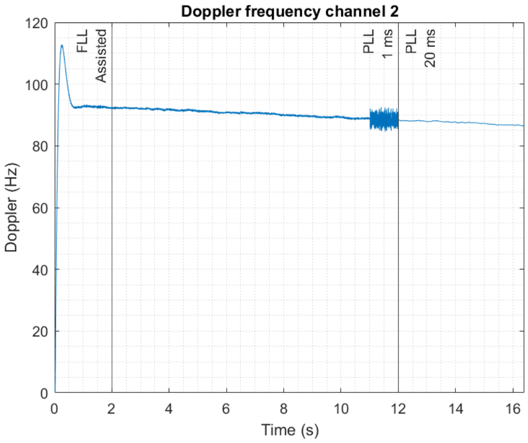
(a) PLL thermal noise, for several PLL Bandwidths at Jerk of 10g/s and integration time of 1ms.

(b) Parametric plots of FLL dynamics stress errors vs. FLL B_W.

Figure 5.12: FLL dynamics behavior.

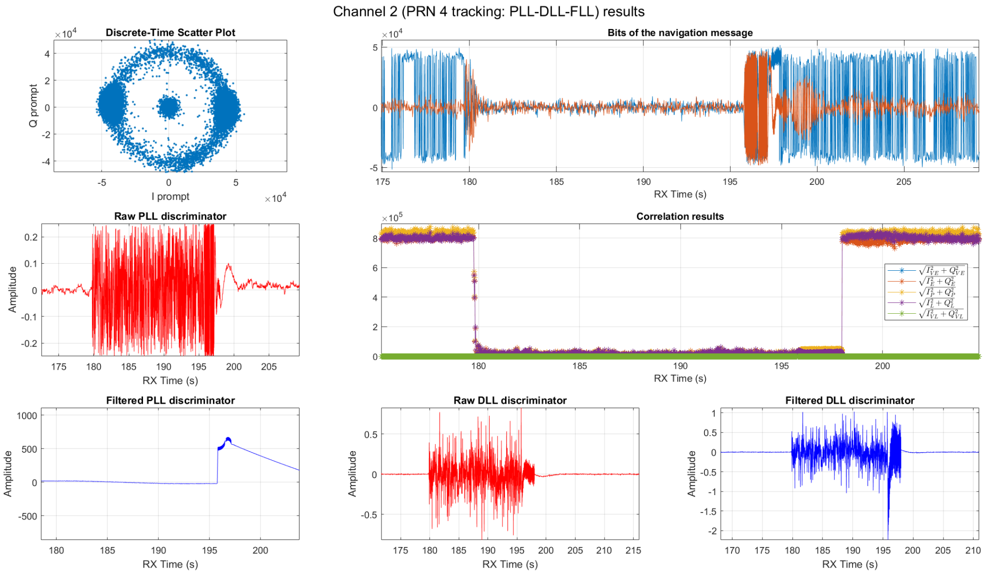


(a) FLL assisted PLL-DLL tracking.

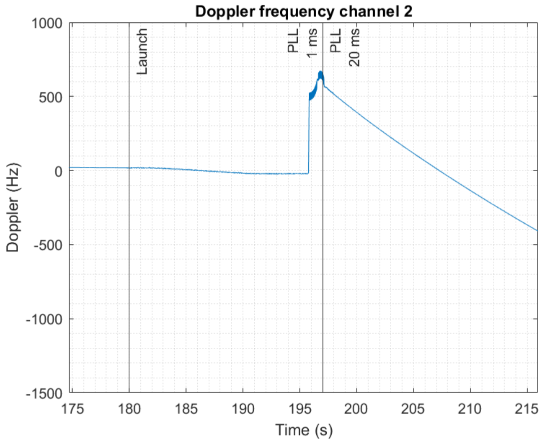


(b) Doppler shift of SV 4 recovered with a FLL assisted PLL-DLL tracking.

Figure 5.13: Detailed behavior of FLL assisted PLL-DLL tracking after acquisition in static position.

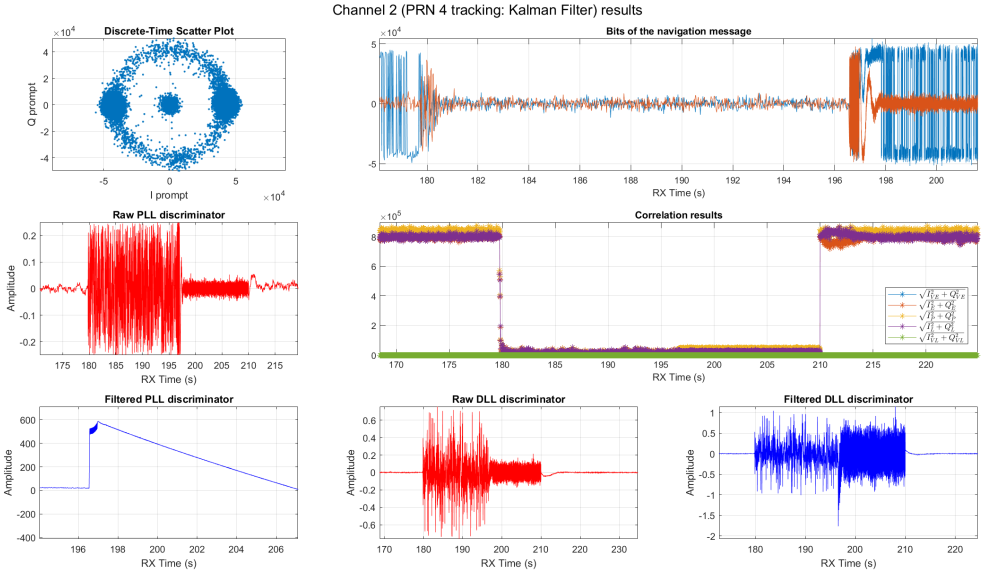


(a) FLL assisted PLL-DLL tracking, after launch event.

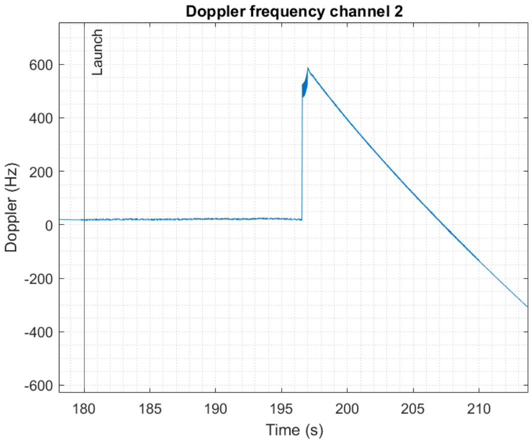


(b) Doppler shift of SV 4 recovered with a FLL assisted PLL-DLL tracking, after launch event.

Figure 5.14: Detailed behaviour of FLL assisted PLL-DLL tracking with re-acquisition after launch event.



(a) Kalman Filter tracking, after launch event.



(b) Doppler shift of SV 4 in Kalman Filter tracking, after launch event.

Figure 5.16: Detailed behavior of Kalman Filter tracking after re-acquisition after launch event.

Chapter 6

Sensor fusion navigation

So far, the work of the previous two chapters has focused on studying what SDR-based GNSS receivers are and how GNSS receivers can be used onboard rockets and projectiles: their peculiarities and limitations. It has also been justified, after analyzing previous works and the state of the art, that it is the main navigation system used by nearly all mature guided munition developments. However, even after Chapter 5 the limitations of GNSS receivers have also become clear.

The use of auxiliary sensors improves, on the one hand, the availability of navigation at different stages of the flight, for whatever reason. As will be explained throughout this chapter, fusing information from GNSS receivers with other sensors will allow us to deal with both jamming or spoofing attack events, as well as adverse conditions, such as the accelerations and jerks suffered during the launch.

The objective of this chapter is to introduce the characteristics that these sensors have, why they are necessary, and the different integration architectures. In addition, one of the contributions of the thesis will be detailed: the Vector Tracking algorithm compatible with a software-defined GNSS receiver that is based on GNU radio architecture.

6.1 Sensors

Generally, GNSS receivers are combined with inertial sensors. This is because they are complementary systems. Inertial type sensors (INS or IMU) made up of accelerometers and gyroscopes allow attitude and position to be obtained through integration, a large number of times per second. However, GNSS receivers provide position, speed and heading are absolute, although with lower refresh rates. The availability of inertial sensor measurements is maximum, but the quality decreases over time because it is an integration. After a certain time, these measures will result from the accumulation of errors and will not be useful. In contrast, although susceptible to signal loss, the data provided by the GNSS receiver will not degrade over time and can be used to *reset* or *align* the inertial unit when available [47]. As a result, a system made up of a GNSS receiver and an IMU/INS will provide the attitude, position, and velocity of the vehicle in which it is installed [114].

Due to space and cost requirements onboard rockets and projectiles, the quality of these units is limited. Given current technology, units are limited up to *tactical* grade. Moreover, as introduced in Chapter 2, the accelerations suffered by the navigation units are so great that mechanical resistance is another aspect to take into account. The combination of these three characteristics: low size, low cost, and high mechanical resistance means that MEMS-type inertial sensors are always used. In fact, as introduced, the development of sensors capable of supporting these dynamics and the development of the navigation systems themselves was in parallel at the beginning. Detailed studies on strap-down MEMS IMU (accelerometers + gyros) have been carried out widely in [17], [18] and in flights in [36]. However, not as many tests are available regarding GNSS-only data, and only few related GNSS/IMU hybridation data [104]. This is in part due to the challenge of getting the data from the test device (rocket) as shown in [35], [49], and [37].

6.2 GNSS/INS Architectures

Sensor fusion architectures in navigation systems are categorized by the degree of integration between sensors, ranging from independent systems to deeply integrated approaches. As the level of integration increases, so do the complexity and interaction between sensors, leading to better performance of navigation systems. Each strategy offers different advantages and trade-offs based on the level at which the sensor data is fused [71].

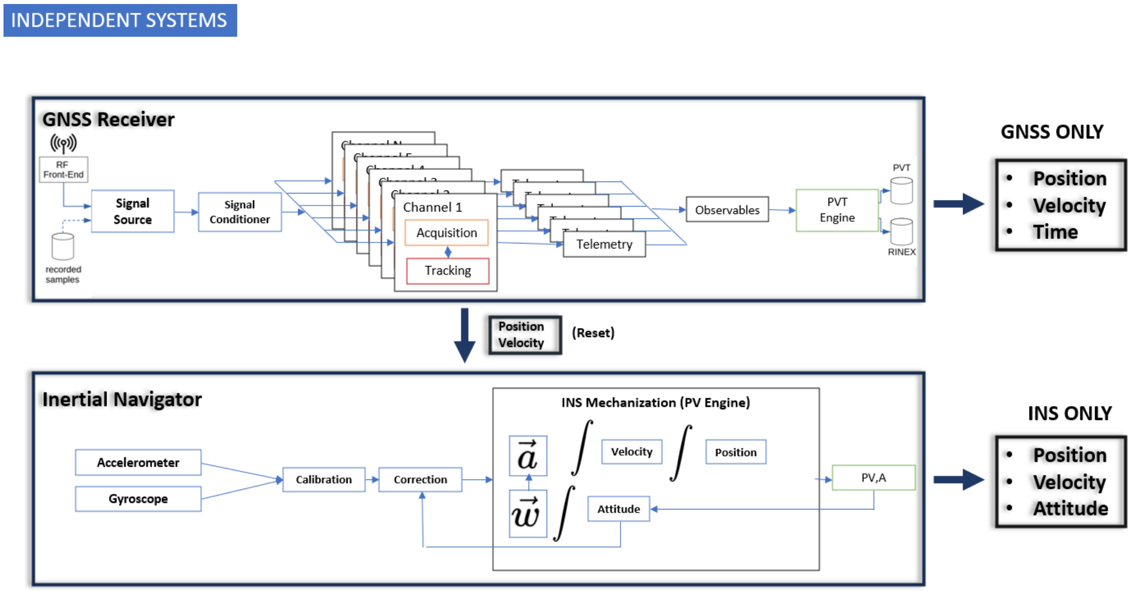


Figure 6.1: Independent GNSS and IMU/INS navigation architectures.

- Independent: Each provides their own solution (Figure 6.1). GNSS can be used to reset the INS error. From now on, they are no longer totally independent [142].
- Loosely Coupled: In a loosely coupled sensor fusion architecture (Figure 6.2), different sensor measurements are processed independently and their outputs are combined at a higher level. For example, the outputs of an inertial navigation system (INS) and a

Global Navigation Satellite System (GNSS) receiver might be combined in a higher-level navigation algorithm. Sensors operate somewhat independently, and their fusion occurs in a higher processing layer [44], [47], [119], [143].

- Tightly Coupled: The tightly coupled architectures (Figure 6.3) involve a more integrated approach, in which the outputs of different sensors are combined at a lower level in the navigation algorithm. For instance, in a tightly coupled GPS-INS integration, the GPS measurements directly influence the INS state estimation, and vice versa. This leads to a more integrated and coherent fusion of sensor data [72] [168] [48] [143] [47].
- Ultra-Tight Integration: Deep integration (Figure 6.4) goes a step further by not only fusing sensor measurements but also integrating them at a deeper level within the navigation system. This may involve combining information at the raw sensor data level, taking advantage of the inherent relationships between different sensor outputs. Deep integration often requires a more complex and sophisticated algorithm to effectively leverage the synergies between sensors [9], [47], [62], [94], [124], [141].

6.2.1 Trade off

Loosely Coupled

A loosely coupled structure schema can be found in Figure 6.2. These architectures offer some advantages. One of the most significant features of these is their modularity, which allows for the straightforward integration of various sensors and subsystems. This modular nature improves the flexibility of the system, as sensors can be added or replaced with minimal disruption, avoiding the need for major changes to the overall structure. Furthermore, because the sensors operate somewhat independently, the overall complexity of the system is reduced. This leads to a simpler and more manageable design compared to tightly coupled or deeply integrated approaches [142].

However, this approach also has disadvantages. One notable drawback is the limited optimization potential, which may result in suboptimal performance, particularly in challenging environments where GNSS information is not fully available, such as GNSS jamming or spoofing scenarios or high-dynamics vehicles. This architecture fuses the information at the PVT level, meaning that if the complete GNSS solution is not available or degraded, the full performance of the system will rely on the IMU/INS solution, directly related to its grade. Although errors in IMU measurements are corrected or *reset* by GNSS, the quality of the GNSS solution is only matched with high-end IMUs/INS (tactical and above for a regular GNSS receiver). If positioning performance is required, high-rate GNSS corrections are needed, or IMUs/INS above tactical grade must be used [46]. A good comparison between loosely and tight integration is the thesis of Petovello [124].

Tightly Coupled

Tightly coupled architectures can provide higher performance by directly integrating sensor measurements at a lower level, leading to even precise results in some scenarios. Additionally, tightly coupled systems perform well under degraded conditions, such as when a GNSS

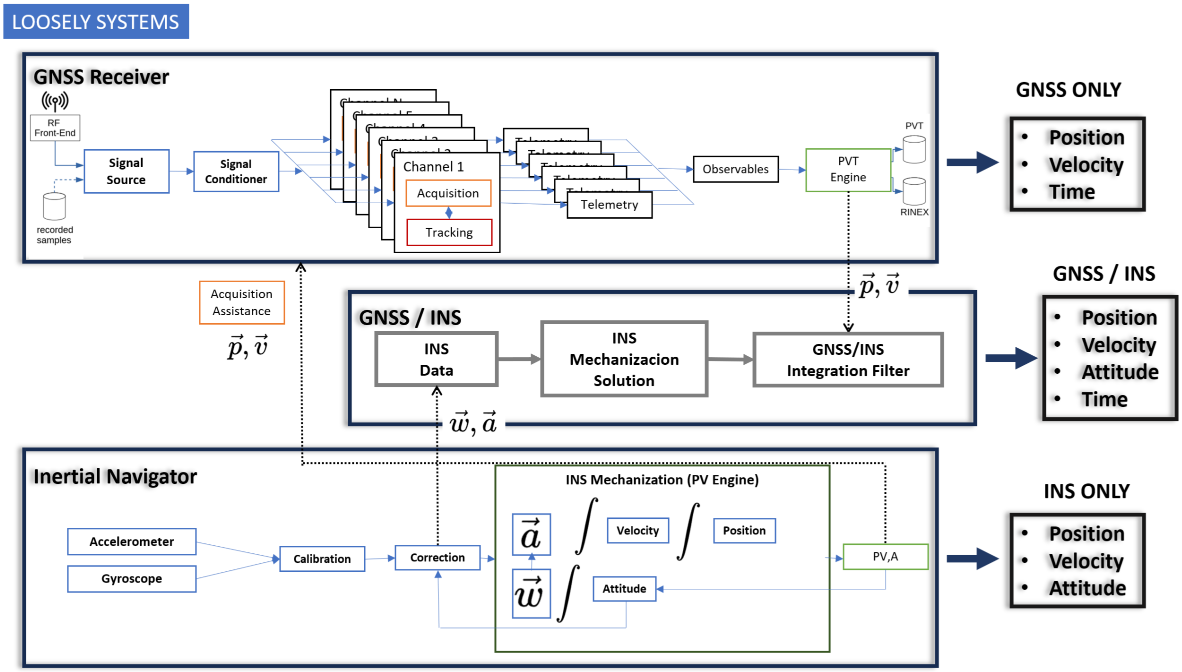


Figure 6.2: Loosely coupled architectures schema.

becomes temporarily unavailable, by maintaining accurate estimates using data from other sensors. Although still tactical grade and above IMUs/INS systems are required, the rate of correction update can be lowered, depending on the application [143]. One main difference from the loosely approach is that the positioning filter is now joined (Figure 6.3). That implies that GNSS measurements are used, even if there is no PVT solution from the receiver available (for example, there are only three SVs on view).

However, the increased complexity of tightly coupled architectures makes them more challenging to design and implement. They require careful synchronization and integration of sensor data [119]. This can even lead to practical issues, as when using COTS systems the raw measurements from the Receiver are not always available. This complexity also places a higher computational load on the system, which can be a drawback for rockets with limited processing capabilities. Furthermore, the reduced modularity of tightly coupled systems means that adding or replacing sensors often requires more extensive modifications.

Ultra-Tight Integration

Deep integration architectures provide several significant advantages in sensor fusion for navigation systems. Chief among these is the synergistic exploitation of sensor data, where inter-sensor synergies are leveraged to extract and utilize complementary information more effectively. This architecture also improves system robustness by addressing specific challenges through complex nonlinear relationships between sensor outputs, allowing sustained navigation performance even under adverse conditions. Furthermore, deep integration allows for optimized performance, as the algorithm can account for and exploit the interdependencies between sensor measurements, resulting in highly refined and accurate solutions.

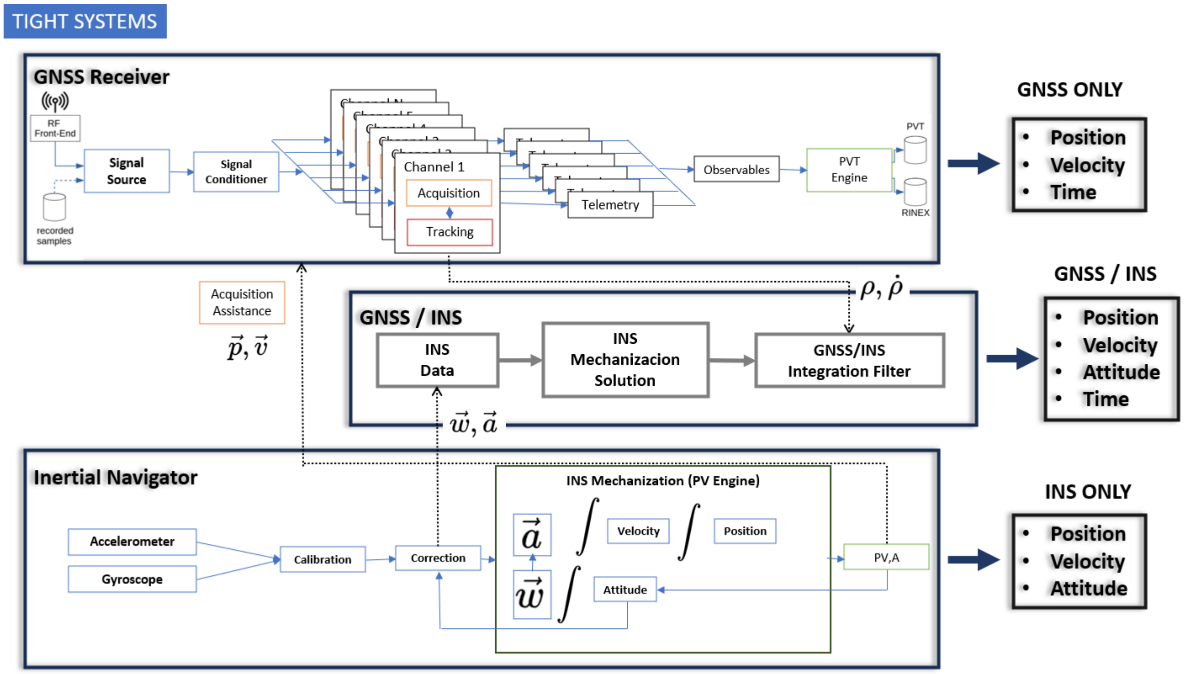


Figure 6.3: Tightly coupled architectures schema.

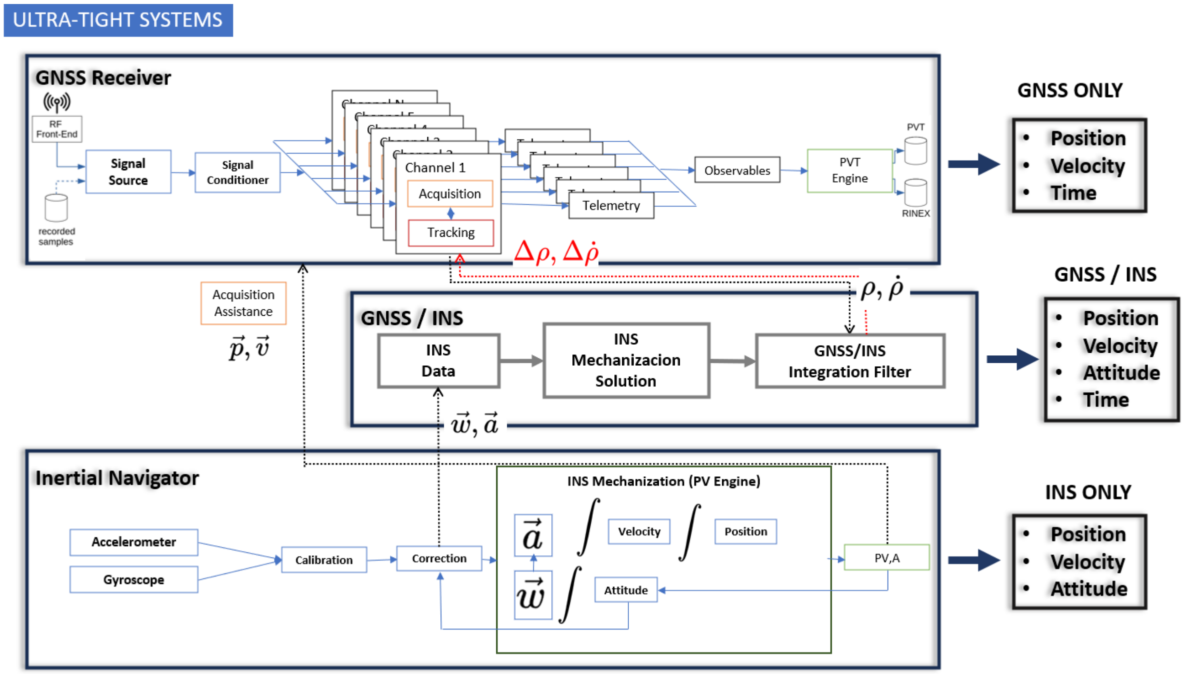


Figure 6.4: Ultra-Tightly coupled architectures schema.

However, deep integration presents several technical challenges. The primary drawback is the increased algorithmic complexity, as the architecture requires sophisticated algorithms capable of fully exploiting the interrelationships between raw sensor outputs, which significantly increases the complexity of the system. In addition, there is a greater development effort involved, as the creation, calibration, and validation of these advanced algorithms require substantial time and rigorous testing. Finally, the architecture places higher computational demands on the system, as the fusion of raw sensor data at deeper levels requires greater processing power, potentially limiting its applicability in systems with constrained computational resources.

6.3 Vector Tracking

In traditional scalar tracking loops, each GNSS satellite signal is tracked independently by separate loops, does not exploit the inherent relationship between signal tracking and the navigation state estimation, and position/velocity estimation occurs separately after signal tracking is completed. These loops do not share information with each other, meaning that each satellite is processed in isolation [84]. In contrast, Vector Tracking Loops (VTL) integrate the processes of signal tracking and position/velocity estimation into a single algorithm. Rather than treating each satellite independently, vector tracking loops use the combined data from all visible satellites to track signals and estimate the position and velocity of the receiver simultaneously. leverage the intrinsic coupling between signal tracking and navigation computation, combining these processes into a single step. With vector tracking, the navigation processor performs both tasks, eliminating the need for separate tracking loops. The navigation processor predicts pseudoranges and pseudorange rates, which are then used in the tracking process. This creates a more interdependent system where errors from one satellite can be mitigated by information from others, leading to potential improvements in robustness, especially in challenging signal environments like urban canyons, areas with weak signals or high-dynamics. The combined information helps stabilize the tracking of weak or obstructed signals. It was first proposed by [121] and has been the subject of a great deal of work in the community. A comprehensive summary can be found in [123], and a recent review can be found in [127].

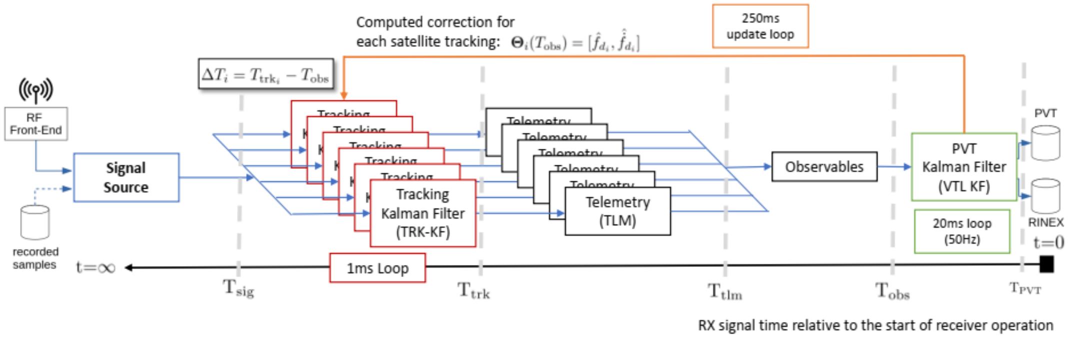


Figure 6.5: VTL algorithm implementation high level block diagram.

Vector Tracking Loops (VTL) offer several advantages over scalar tracking loops, with one of the most frequently cited being their enhanced resistance to interference and jamming. Vector tracking can operate effectively at lower carrier-to-noise power density ratios (C/N_0), improving channel tracking performance, especially in adverse scenarios: Multipath, urban cannon or scintillation ([34] or [148]). Another key benefit is the ability of vector tracking algorithms to bridge signal outages and rapidly re-acquire signals that have been temporarily blocked [75]. Furthermore, the flexibility of the vector tracking architecture allows the receiver's motion to be restricted to specific dimensions. This feature can be particularly useful for applications where movement is predominantly in one or two directions, such as ships or automobiles [122]. Finally, vector tracking loops demonstrate superior immunity to high receiver dynamics, compared to scalar tracking loops [89].

However, VTL are not without drawbacks. One of the primary challenges is their increased computational load and system complexity. The Kalman filter, central to the vector tracking architecture, must be updated at a rate that matches the integrate-and-dump period of the algorithm, typically 50 Hz [165]. Furthermore, the numerically controlled oscillators (NCOs) in each channel must be controlled by the central Kalman filter, which adds to the processing burden of the system. Another potential downside is that faults in one channel can propagate across all channels, leading to overall system instability or loss of signal lock on multiple satellites [167].

6.3.1 Vector Tracking Loop algorithm

Before delving into the operation of vector tracking loops, it is useful to first review how traditional receivers function. Figure 4.2 shows a block diagram of a conventional GPS receiver. In these traditional GNSS receivers, scalar tracking loops are used to estimate the pseudo-ranges and pseudo-range rates for each available satellite. Typically, a Delay Lock Loop (DLL) is used to estimate pseudoranges, while a Phase-Lock Loop (PLL) with a Costas discriminator pseudorange rates or carrier Doppler (Figure 5.7). The pseudoranges and pseudorange-rates are passed to the navigation processor, which then computes the receiver's position, velocity, clock bias, and clock drift (the *navigation states*). In most cases, the navigation processor uses either an iterative least squares algorithm or a Kalman filter to solve for these states. The data flow within the receiver moves strictly from left to right (Figure 4.2). Each receiver channel independently tracks its respective satellite signal without sharing information between channels. Additionally, no feedback from the navigation processor is provided to the tracking loops. The only exception to this might be during satellite acquisition, where the navigation solution is occasionally used to initialize the acquisition process (potentially reducing the acquisition time, but not enhancing the signal tracking capabilities).

In contrast, as introduced previously, VTL uses jointly all the information available from the active satellite channels to estimate not only the user PVT solution, but also the satellite signal synchronization parameters. Extensive work on this architecture has been done recently, as comprehensive examples can be found in [51], [171] or [170]. None of them with real-time code is available to the community.

Figure 6.5 shows a block diagram of the implementation of the proposed VTL algorithm using the GNSS-SDR receiver architecture, based on the one presented in [51]. From left to right, the baseband signal samples are fed to the signal source block either from a front-end, in a real-time operation, or from a binary file storing raw samples, in a post-processing operation. The signal acquisition operation does not change in the proposed VTL implementation. The tracking function in the proposed VTL architecture consists of independent Kalman filters for each visible satellite, handling both carrier phase and code delay (represented as TRK KF boxes in the diagram). A detailed explanation of the Kalman filter implementation per channel was provided in Section 5.3.3.

The source code is released under the General Public License v3.0 and will contribute to GNSS-SDR's open-source code base: `https://github.com/gnss-sdr/gnss-sdr-vtl`, and an application can be found on [67].

6.3.2 Reduced rocket model and accelerometer models

The acceleration of the vehicle can be measured directly, by means of accelerometers of the appropriate ranges embedded in the rocket, or estimated from a model. The modeling of the accelerometer readings is represented as $\vec{a}_G^{B_i}$. The sensor is corrupted by a white noise term $\boldsymbol{\eta_a}$, defined as $\boldsymbol{\eta_a} = (\eta_{a,x}, \eta_{a,y}, \eta_{a,z})$, with mean $E(\eta_{a,i}) = 0$ and variance $\mathrm{Var}(\eta_{a,i}) = \sigma^2_{\eta_{a,i}}$. More details of this can be found in [114]. Notice that $\vec{g}^{B_i}$ represents the components of the gravity vector in the body coordinates:

$$\vec{a}_G^{B_i} = \vec{a}_G^{B_i} - \vec{g}^{B_i} + \boldsymbol{\eta_a} \ . \tag{6.1}$$

Acceleration is simulated by the 5DoF rocket model in the previous section and corrupted in the PVT engine, with $\sigma^2_{\eta_{a,i}} = 5.3mg^2$. This value is extracted as a reference from an ADXL375 [4]. On the other hand, if the usage of an accelerometer is to be avoided, an acceleration model can be developed by a reduced rocket model with 3 DoF that takes into account only a ballistic flight with thrust and aerodynamic drag, C_{D0}, of the vehicle (more details can be found in [98]):

$$\vec{a}_G^{B_i} = -\frac{(\pi\rho \cdot d^2/8)C_{D_0}v^2}{m} \cdot \vec{x} + \vec{g}^{B_i} + \frac{T}{m} \cdot \vec{x} \ . \tag{6.2}$$

6.3.3 Proposed algorithm

This work focuses on a high-dynamics scenario, namely the launch of a rocket. Real-time feedback of the Doppler shift and the Doppler shift rate, since it is the effect that causes loss of tracking of the channels (as shown in [134]).

The geometric PVT problem is defined by the pseudodistance between satellites and users, defined for each i satellite at instant k,

$$\rho_{ik} = \sqrt{(x_{ik}^{Tx} - x_{uk})^2 + (y_{ik}^{Tx} - y_{uk})^2 + (z_{ik}^{Tx} - z_{uk})^2} + c\delta t_{uk}^{(s)} + \sigma_{\epsilon k} \ , \tag{6.3}$$

where $({x^{Tx}}_{ik}, {y^{Tx}}_{ik}, {z^{Tx}}_{ik})$ is the satellite position known from the navigation message, (x_{uk}, y_{uk}, z_{uk}) the user receiver position, $c\delta {t_{uk}}^{(s)}$ the user clock deviation and $\sigma_{\epsilon k}$ gathers other sources of error. Since the receiver must estimate its own 3D position and its clock deviation with respect to the time basis of the satellites, the receiver must see at least four satellites at the same time. Each received satellite signal defines one equation such as the one defined in (6.3), forming a set of nonlinear equations that once solved provides the PVT fix.

Most commercial receivers solve this either by a weighted least squares or by a Bayesian estimation when some *a priori* information is available. Examples of this *a priori* information would be: motions models, statistical characterization of noises or additional information from other systems. In this work, two types of *a priori* information are explored: *a*) the in-advance knowledge of the rocket propulsion profile, and *b*) the presence of an additional single-axis accelerometer to the GNSS receiver within the navigation sensor suite.

The selected tool of the Bayesian framework is the Extended Kalman Filter (EKF) in its error-state version. This algorithm is a standard for real-time recursive problems. As was introduced in [119], the use of error state instead of complete state is suitable due to the legitimate use of linearization in cases where states or measurements are large and small numbers at the same time. The states needed for describing the movement are in this case: Earth-centered Earth-fixed (ECEF) frame position, velocities, accelerations, clock bias, and clock bias rate. These states in their absolute values are in at least three different magnitude orders (ECEF 10^6, velocities, and accelerations from 10^1 to 10^2.)

The EKF needs two models: the system and the measurement model. Regarding the system, the model of 3DoF (Equation 6.2) calculates the acceleration at each time step, leading to a non-linear propagation equation $\boldsymbol{\Delta}\chi_{u_{k+1}} = f(\boldsymbol{\Delta}\chi_k)$. Its linearization,

$$\begin{bmatrix} \Delta x_{uk+1} \\ \Delta y_{uk+1} \\ \Delta z_{uk+1} \\ \Delta \dot{x}_{uk+1} \\ \Delta \dot{y}_{uk+1} \\ \Delta \dot{z}_{uk+1} \\ \Delta \ddot{x}_{uk+1} \\ \Delta \ddot{y}_{uk+1} \\ \Delta \ddot{z}_{uk+1} \\ \Delta c\delta t_{uk+1} \\ \Delta c\delta \dot{t}_{uk+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 & \Delta T^2/2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & \Delta T^2/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & \Delta T^2/2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 0 & 0 & M_x & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & M_y & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & M_z & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_{uk} \\ \Delta y_{uk} \\ \Delta z_{uk} \\ \Delta \dot{x}_{uk} \\ \Delta \dot{y}_{uk} \\ \Delta \dot{z}_{uk} \\ \Delta \ddot{x}_{uk} \\ \Delta \ddot{y}_{uk} \\ \Delta \ddot{z}_{uk} \\ \Delta c\delta t_{uk} \\ \Delta c\delta \dot{t}_{uk} \end{bmatrix} , \tag{6.4}$$

where $M_x = -\frac{(\pi\rho \cdot d^2/8)C_{D_0}}{m}(u_x^2/|u| + |u|)$, $M_y = -\frac{(\pi\rho \cdot d^2/8)C_{D_0}}{m}(u_y^2/|u| + |u|)$ and the z-component: $M_z = -\frac{(\pi\rho \cdot d^2/8)C_{D_0}}{m}(u_z^2/|u| + |u|)$ came from the aforementioned linearization of Equation 6.2. The effect is constraining the relationship between velocity and acceleration. In the case of the acceleremoter-aided version these terms would be zero, leading to a fully kinematic version of the propagation step. This can also be written in matrix notation, analogous to Equation 5.26:

$$\boldsymbol{\Delta}\chi_{u_{k+1}} = \mathbf{F}_{k-1}\boldsymbol{\Delta}\chi_k + \mathbf{v}_k \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q_k}) \ , \tag{6.5}$$

being $\mathbf{v}_k$ the process noise term. This term will the noise variances of the sensors in the accelerometer aided approach (Equation 6.1), whereas will be heuristically tuned to the model uncertainties in the case of the rocket model aided approach. The Jacobian matrix of the measurement equations obtained from the non-linear version measurement Equation 6.3 is rewritten for convenience,

$$\rho_{ik} = d_{ik} + c\delta t_{uk}^{(s)} + \sigma_{\epsilon k} \ , \tag{6.6}$$

where:

$$d_{ik} = \sqrt{(x_{ik}^{Tx} - x_{uk})^2 + (y_{ik}^{Tx} - y_{uk})^2 + (z_{ik}^{Tx} - z_{uk})^2} \ , \tag{6.7}$$

and then,

$$\dot{\rho}_{ik} = (\dot{x}_{ik}^{Tx} - \dot{x}_{uk})a_{x_{ik}} + (\dot{y}_{ik}^{Tx} - \dot{y}_{uk})a_{y_{ik}} + (\dot{z}_{ik}^{Tx} - \dot{z}_{uk})a_{z_{ik}} + c\delta\dot{t}_{uk}^{(s)} + \sigma_{\dot{\epsilon}k} \ . \tag{6.8}$$

Its Jacobian matrix can be written,

$$\begin{bmatrix} \Delta\rho_{1k} \\ \vdots \\ \Delta\rho_{Nk} \\ \Delta\dot{\rho}_{1k} \\ \vdots \\ \Delta\dot{\rho}_{Nk} \\ \Delta\ddot{\rho}_{1k} \\ \vdots \\ \Delta\ddot{\rho}_{Nk} \end{bmatrix} = \begin{bmatrix} a_{x1_k} & a_{y1_k} & a_{z1_k} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{xN_k} & a_{yN_k} & a_{zN_k} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & a_{x1_k} & a_{y1_k} & a_{z1_k} & 0 & 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & a_{xN_k} & a_{yN_k} & a_{zN_k} & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{x1_k} & a_{y1_k} & a_{z1_k} & 0 & \Delta T \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{xN_k} & a_{yN_k} & a_{zN_k} & 0 & \Delta T \end{bmatrix} \begin{bmatrix} \Delta x_{uk} \\ \Delta y_{uk} \\ \Delta z_{uk} \\ \Delta\dot{x}_{uk} \\ \Delta\dot{y}_{uk} \\ \Delta\dot{z}_{uk} \\ \Delta\ddot{x}_{uk} \\ \Delta\ddot{y}_{uk} \\ \Delta\ddot{z}_{uk} \\ \Delta c\delta t_{uk} \\ \Delta c\delta\dot{t}_{uk} \end{bmatrix} , \tag{6.9}$$

where $a_{xi_k} = -\frac{x_{ik}^{Tx} - x_{uk}}{d_{ik}}$, $a_{yi_k} = -\frac{y_{ik}^{Tx} - y_{uk}}{d_{ik}}$, $a_{zi_k} = -\frac{z_{ik}^{Tx} - z_{uk}}{d_{ik}}$. A more detailed derivation can be found in [51]. The EKF measurement equation in matrix notation analogous to (5.18),

$$\begin{bmatrix} \boldsymbol{\Delta}\rho_k \\ \boldsymbol{\Delta}\dot{\rho}_k \\ \boldsymbol{\Delta}\ddot{\rho}_k \end{bmatrix} = \mathbf{H}_k \Delta\chi_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R_k}) \ , \tag{6.10}$$

where $\mathbf{H}_k$ is the linearized observation model, that maps the state space error into the measures error and will be used to calculate $\mathbf{K}_k$ and $\hat{\mathbf{P}}$, $\boldsymbol{\Delta}\chi_k$ is the Position-Velocity-Time (PVT) estimation error at the instant k. The pseudorange rate measurements $\dot{\rho}_k$ can be extracted from the received signal deviation with respect to the nominal carrier frequency f_c,

$$\begin{aligned} f_{d_{ik}} &= -\left(\frac{\dot{x}_{ik}^{Tx} - \dot{x}_{uk}}{d_{ik}} + \frac{\dot{y}_{ik}^{Tx} - \dot{y}_{uk}}{d_{ik}} + \frac{\dot{z}_{ik}^{Tx} - \dot{z}_{uk}}{d_{ik}} \right) \frac{fc}{c} \ , \\ \hat{\dot{\rho}}_k &= -\frac{f_{d_{ik}}}{c/f_c} + c\delta\dot{t}_k + \sigma_{\dot{\epsilon}_k} \ . \end{aligned} \tag{6.11}$$

6.4 Real time implementation in GNSS-SDR framework

Reviewing the algorithmic design of a navigation receiver is of special importance for an SDR receiver because the SDR concept potentially provides the possibility to redesign well-known navigation signal processing techniques to implement more complex algorithms compared to hardware receivers.

The choice of a particular implementation strategy of a VTL algorithm for a real-time software-defined GNSS receiver is tightly coupled to its internal structure and information flow. In this work, we selected the well-known open source receiver implementation GNSS-SDR [64]. We choose it for its modular code structure and for its flexibility to use existing synchronous and asynchronous information exchange mechanisms between signal processing blocks.

GNSS-SDR is internally based on the GNU Radio [65] data streaming transport mechanism between signal processing blocks. GNU Radio is, by design, a concurrent multitasking SDR framework in which each processing block has its own input and output data buffers. Thanks to the buffering system, it is possible to execute concurrently the operations of several chained signal processing blocks. The complete set of interconnected signal processing blocks is called a *flow graph*. Each signal processing block in the flow graph is consequently working in a different receiver signal time, and this fact implies that, for instance, if it is required to send a message from the positioning engine (which is placed at the very end of the GNSS-SDR flow graph) to the tracking blocks, this information will arrive when the tracking is processing signal at some point in a future time with respect to the time of positioning engine. It is not possible to block the flow graph and wait for the message to arrive at the tracking block, as this leads the entire flow graph to enter a deadlock state trying to fill its intermediate buffers, as explained in [50] and [59].

The architecture proposed in this work is summarized in Figure 6.5. Following the tracking blocks, the telemetry decoder blocks, demodulates telemetry and timestamp signal timestamping in the GNSS time frame (Time of Week). After that, all channels are fed to the observables block, which produces code and carrier observables at a common reception time. Finally, the custom implementation of the PVT VTL KF block receives all the observables at receiver time T_{obs} and executes the VTL Kalman filter loop. After each VTL KF epoch, it is required to send the new state information $\boldsymbol{\Theta}_i(T_{\text{obs}}) = [\hat{f}_{d_i}, \hat{\dot{f}}_{d_i}]$ to the $i-$th satellite tracking block using an asynchronous messaging mechanism.

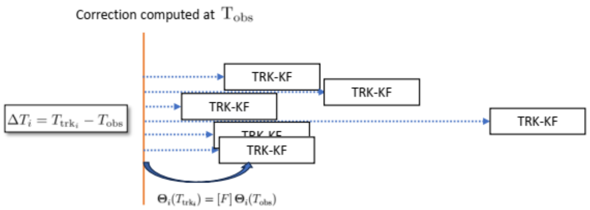


Figure 6.6: Detail of TRK-KF propagation of a vector carrier tracking correction.

However, due to the limitations imposed by the flow graph operation, when this message arrives, the Kalman tracking filter processes a *future* time (Figure 6.6), with a time advance of

$\Delta T_i = T_{\mathrm{trk}_i} - T_{\mathrm{obs}}$. Note that, due to the status of the internal buffers, ΔT_i may be different for each satellite channel. The tracking Kalman filter projects the received state information from T_{obs} to its current signal time T_{trk_i} using the state transition matrix (see Section 5.3.3),

$$\boldsymbol{\Theta}_i(T_{\mathrm{trk}_i}) = \begin{bmatrix} 1 & 0 & \beta\Delta T_i & \frac{\beta\Delta T_i^2}{2} \\ 0 & 1 & 2\pi\Delta T_i & \pi\Delta T_i^2 \\ 0 & 0 & 1 & \Delta T_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{\Theta}_i(T_{\mathrm{obs}}) \ . \tag{6.12}$$

The fact that for each VTL epoch state update, there is a delay in its application to the tracking loops requires the VTL to not execute the next loop epoch until it is ensured that the last correction was applied to all the visible satellites. If this is not taken into account, the VTL loop would be using observables that are not corrected with the KF VTL state updates, accumulating error and sending back new worse corrections to the TRK-KF (orange line in Figure 6.5). Therefore, there is a superior limit for the feedback frequency, directly related to the latency of the implementation. In a regular real-time operation in an x86 architecture, assuming that the system has reached a steady state without signal overflows, the complete loop delay is bounded to $\Delta T_i \leq 250$ ms. By notifying the application of the state corrections to the VTL Kalman Filter algorithm engine, the VTL states can be propagated in time from the last correction to the current time and then execute a new prediction and update cycle.

Algorithm 1 Vector Tracking Loop implemented with an Extended Kalman Filter.

```
Require: List of $N$ satellites with their corresponding measurements at instant $k$: satellites' position and velocity; transition matrix $\mathbf{F}_k$; previous PVT solution $\hat{\chi}_{u_{k-1|k-1}}$ and its covariance matrix $\hat{\mathbf{P}}_{k-1|k-1}$, estimation of covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$.
Ensure: Estimation of $\hat{\chi}_{u_{k-1|k-1}}$ and its covariance.
 1: if PVT solution available then
 2:     Activate the vector tracking loop in the next iteration.
 3: end if
    Vector tracking loop operation:
    Initialization (First time only)
 4: Use the last least squares PVT solution as initial state $\hat{\chi}_0$.
 5: Use the last PVT solution variance as initial $\hat{\mathbf{P}}_{0|0}$ and $\mathbf{Q}_0$.
    Prediction
 6: $\hat{\chi}_{u_k|k-1} = \mathbf{F_k}\hat{\chi}_{u_{k-1|k-1}}$
 7: $\hat{\mathbf{P}}_{k|k-1} = \mathbf{F_k}\hat{\mathbf{P}}_{k-1|k-1}\mathbf{F}_k^T + \hat{\mathbf{Q}}_{\mathbf{k}k}$
 8: for $i = 1$ to $i =< N$ do
 9:     Compute $\hat{\rho}_k(\hat{r}_{ui_{k|k-1}}, x_{ik}^{Tx}, y_{ik}^{Tx}, z_{ik}^{Tx})$ as in 6.6.
10:     Compute $a_{xi_{k|k-1}}, a_{yi_{k|k-1}}, a_{zi_{k|k-1}}$
11:     Compute $\hat{\dot{\rho}}_k(\hat{\dot{r}}_{ui_{k|k-1}}, \dot{x}_{ik}^{Tx}, \dot{y}_{ik}^{Tx}, \dot{z}_{ik}^{Tx})$ as in 6.8.
12: end for
13: Construct $\mathbf{H}_{k|k-1}$ with the results of step 12.
    Update
14: $\mathbf{S}_{k|k} = \mathbf{H}_{k|k-1}\hat{\mathbf{P}}_{k|k-1}\mathbf{H}_{k|k-1}^T + \mathbf{R}_k$
15: $\mathbf{K}_k = \hat{\mathbf{P}}_{k|k-1}H_{k|k-1}^T(\mathbf{S}_{k|k})^{-1}$
16: $\boldsymbol{\Delta}\hat{\rho}_{\mathbf{k|k}} = \rho_{k-1} + \boldsymbol{\Delta}\rho_k - \boldsymbol{\Delta}\hat{\rho}_{\mathbf{k|k-1}}$
17: $\boldsymbol{\Delta}\hat{\dot{\rho}}_{\mathbf{k|k}} = \dot{\rho}_{\mathbf{k-1}} + \boldsymbol{\Delta}\mathbf{f}_k(-\lambda_c) - \boldsymbol{\Delta}\hat{\dot{\rho}}_{\mathbf{k|k-1}}$
18: Compute $\chi_{u_{k|k}} = \chi_{u_{k|k-1}} + K_k[\boldsymbol{\Delta}\hat{\rho}_{\mathbf{k|k}}\boldsymbol{\Delta}\hat{\dot{\rho}}_{\mathbf{k|k}}]^T$
19: $\hat{\mathbf{P}}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_{k|k-1})\hat{\mathbf{P}}_{k|k-1}(I - \mathbf{K}_k\mathbf{H}_{k|k-1})^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T$ (Joseph form)
```

Chapter 7

Results

In this chapter, simulation and field test results that compare different GNSS tracking architectures are presented. On the one hand, analyze traditional Scalar Tracking Loop (STL) architectures, which include the classic PLL-DLL per channel, PLL-DLL-FLL per channel, and Kalman Filter per channel. On the other hand, explore the Vector Tracking Loop (VTL) architecture, which applies a Kalman Filter across all channels in a vectorized approach. As described in Section 5.3, this combination is necessary for vector architecture in a Software-Defined Radio (SDR) receiver. The following sections provide the results of the test conducted to measure the performance comparison between these architectures, highlighting their strengths and differences.

7.1 Materials and Methods

7.1.1 Experimental setup description

The simulation is performed with a Spirent GSS7700 and Spirent GSS9000, GNSS hardware simulator (L1 signal) [146]. The receiver front-end is an USRPX310 [42], and the samples are sent in real time by Gigabit Ethernet to a standard laptop computer running a GNSS-SDR Receiver instance (Figure 7.1).

A representative flight simulation is performed for the rocket, with a range of 14 km and a shooting elevation of 800 mil (45^o), with an apogee altitude of 10 km. The receiver integration time is 1 ms, and the output of the PVT rate is 50 Hz. Before launch, 3 minutes of static waiting is used to ensure an initial converged PVT solution. The model used for this simulation has been described in Chapter 2.

The satellites vehicles (SV) in view can be seen in Figure 7.2. To ensure repeatability of the experiments, each channel of the SDR is forced to track a specific SV (Table 7.1).

For clarity, the dynamics are summarized as follows: The rocket starts with a strong acceleration of one hundred times gravity (100 g, equivalent to 980 m/s^2). (caused by the thrust rocket motor) for a few seconds (1 s - 2 s), followed by a flight affected only by gravity and air friction (fundamentally a ballistic flight). After the rocket motor runs out, the vehicle

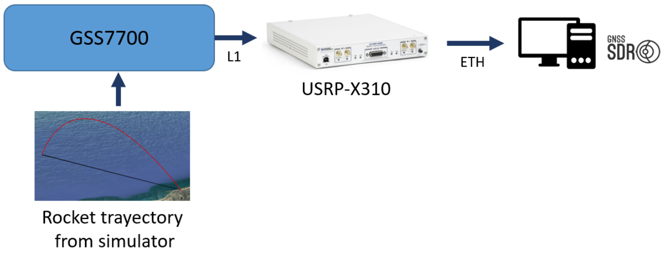


Figure 7.1: Experimental Setup.

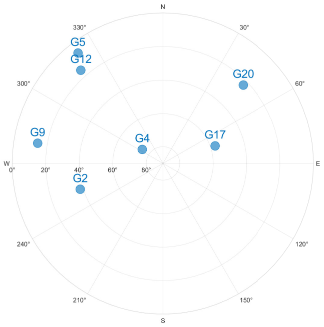


Figure 7.2: Satellites present during launch in the simulations and their position in the sky.

is rapidly decelerated, since it happens to be flying at high speed without propulsion, but within the atmosphere, at a rate 20 times that of gravity (20 g or 196 m/s^2). The flight will last about 60 seconds. The trajectories taken are at launches with an elevation of 33.708° and a range of 20 km. Figure 7.3 shows the simulated dynamics of the rocket.

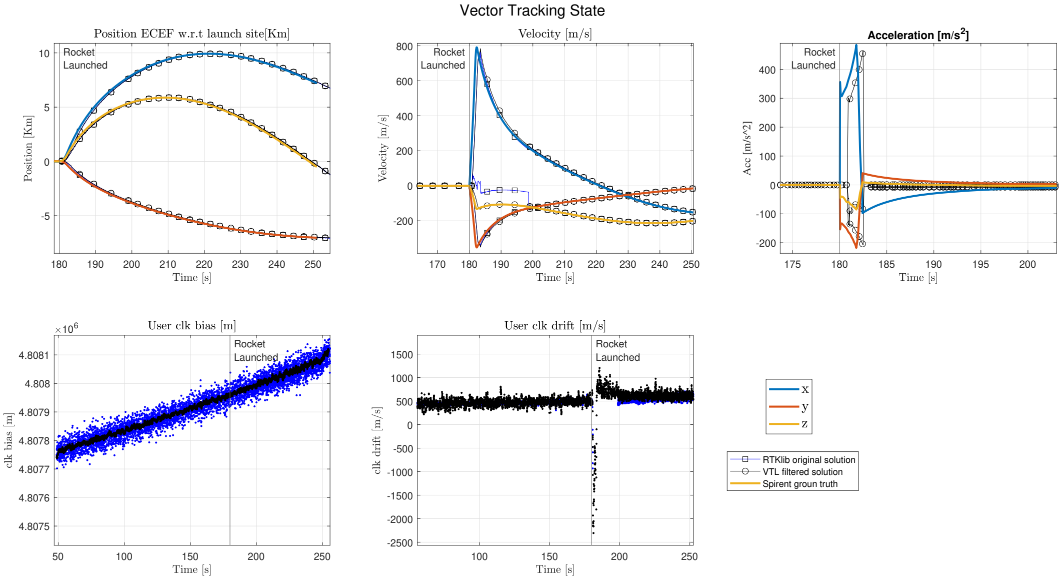


Figure 7.3: Vehicle states accounted in the PVT Kalman Filter: position, velocity, acceleration, clock bias and clock drift.

Table 7.1: GNSS-SDR tracking channels and associated SV PRN.

GNSS-SDR tracking Channel	**SV PRN**
0	20
1	17
2	12
3	9
4	5
5	4
6	2

Figure 7.4: Field tests elements. MC-25 rocket (left) and tracking radar (right) used in field tests (CEDEA).

7.2 Architecture comparison

The Figure 7.7 and the following ones are organized as explained in Chapter 5: the top left is the constellation. To its right are the information bits, with I shown in blue and Q shown in orange. Below are the values of the magnitudes of the correlations. Bottom left are the values of the PLL discriminator (Equation 5.4) and the DLL discriminator (Equation 5.3), with and without filtering applied.

7.2.1 PLL/DLL tracking performance

A typical GNSS receiver uses a combination of Phase-Locked Loop (PLL) and Delay-Locked Loop (DLL) to accurately track the carrier phase and code phase of GNSS signals.

The performance of the proposed implementation is compared to the existing two-step plus classical Least Squares implementation in the original GNSS-SDR implementation, based on the RTKLIB libraries [151] (in the legend as *RTKLIB solution*). The latter was unable to follow the launch dynamics, accumulating errors during launch. This is especially relevant to the Z velocity of the rocket.

7.2.2 PLL/DLL-Frequency assisted tracking performance

The previous architecture proved not to be enough, due to the significant relative motion between the satellite and receiver. As showed in the previous section, the PLL must rapidly and accurately adjust to changes in the carrier frequency and phase caused by Doppler shifts. This could not be achieved by increasing the bandwidth (the only option in that architecture). In this architecture this is solved by adding a Frequency Loop (FLL). This FLL provides a feedback loop that continuously adjusts the local oscillator to match the phase of the incoming signal, ensuring precise carrier phase tracking even under high dynamic conditions.

The Delay-Locked Loop (DLL) is used to track the code phase of the GNSS signal. It ensures that the locally generated replica of the PRN code is in synchronization with the incoming satellite signal. In high dynamics scenarios, the DLL must quickly adapt to changes in the propagation delay caused by rapid changes in the receiver's position. This is critical for maintaining accurate pseudorange measurements, which are essential for precise positioning.

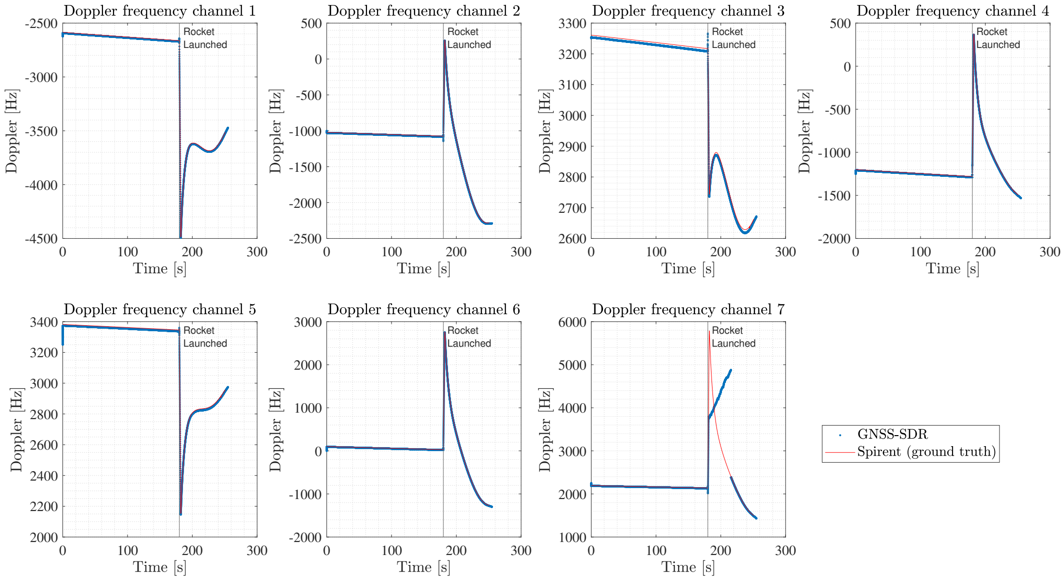


Figure 7.5: Doppler results on channel classic PLL/DLL-Frequency assisted tracker.

As can be seen in Figures 7.5 and 7.6, it is capable of following the satellites, even in phases of flight with very large accelerations and jerks, providing a position solution. However, in those channels subjected to exceptionally difficult conditions (channel 7, SV 2), signal tracking is lost.

The lock in the satellite signal is lost along the boost phase, as can be seen in Figure 7.7: both in the constellation, which coalesces to a single point, and in the phase and code discriminators,

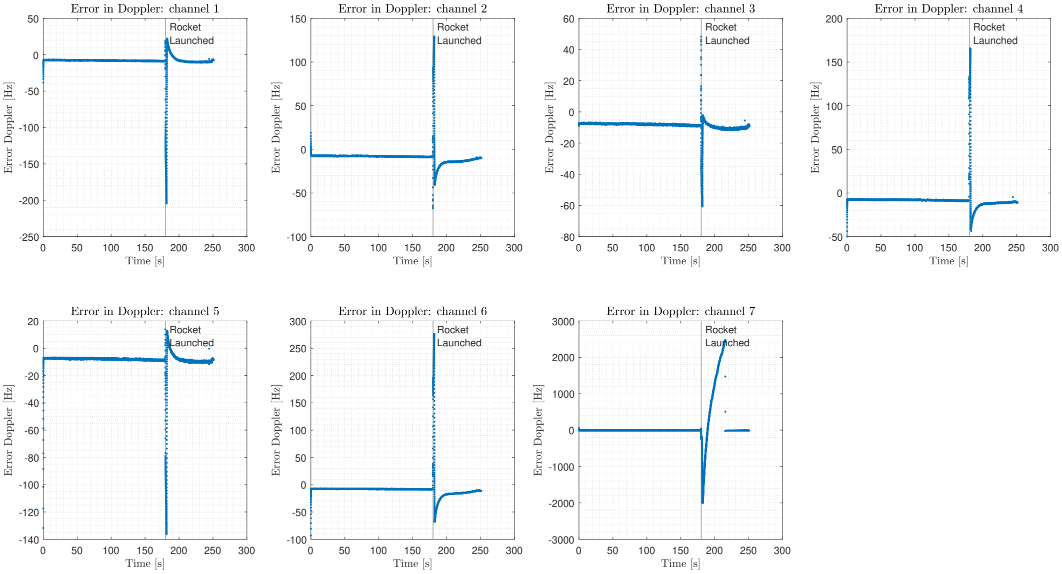


Figure 7.6: Doppler Error on channel classic PLL/DLL-Frequency assisted tracker.

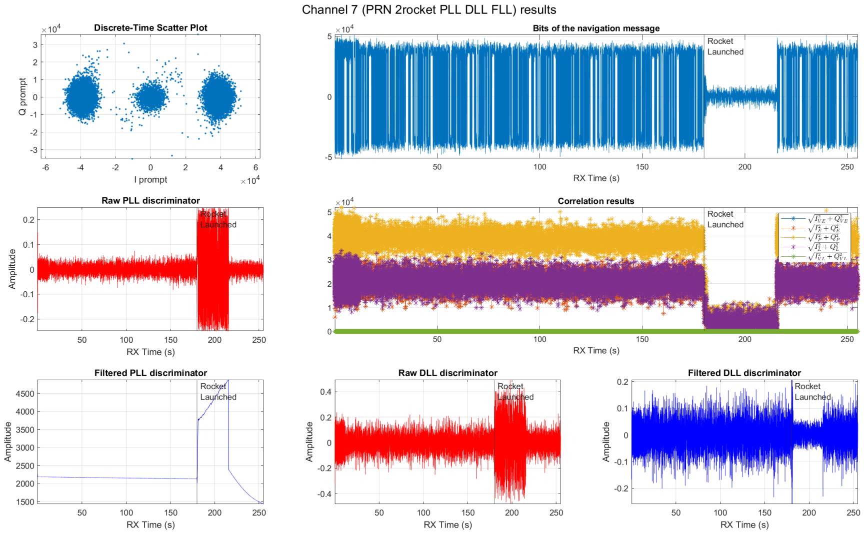


Figure 7.7: Tracking results on channel classic PLL/DLL-Frequency assisted tracker.

which increase above the lock thresholds. This requires entering in the acquisition stage, which takes more than 30 s to recover the channel lock.

This architecture is the one used in [66] in a real launch of the proposed rocket, obtaining a PVT solution throughout the flight. This would also be the architecture analyzed in the Field test scenario, and its behavior will considerably worsen during an actual flight. During a launch, effects such as clock acceleration sensitivity due to the rocket motor, vibrations, or the rotation of the vehicle itself, are added to the Doppler shift and lead to the loss of lock on more satellites.

7.2.3 Kalman filter Tracking performance

This architecture differs from the previous one in the use of a Kalman Filtering (KF) in each of the channels. This is still a Scalar Tracking Loop (STL) architecture. These KF tracking systems are more resistant to variable conditions due to their adaptive nature (reflected in equations 5.20 for the model and 5.21 for measurements).

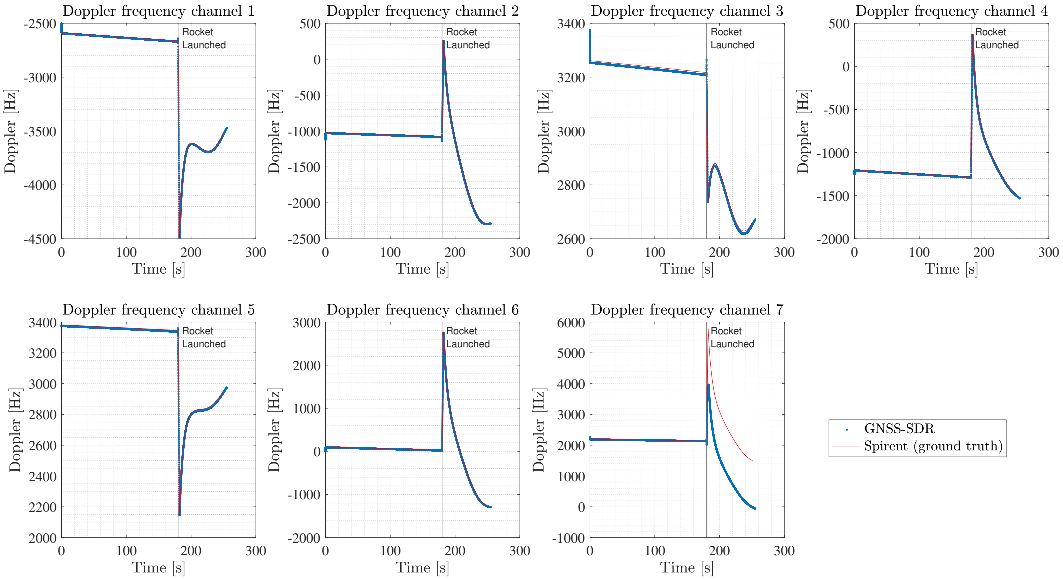


Figure 7.8: Doppler results on channel Kalman Filter.

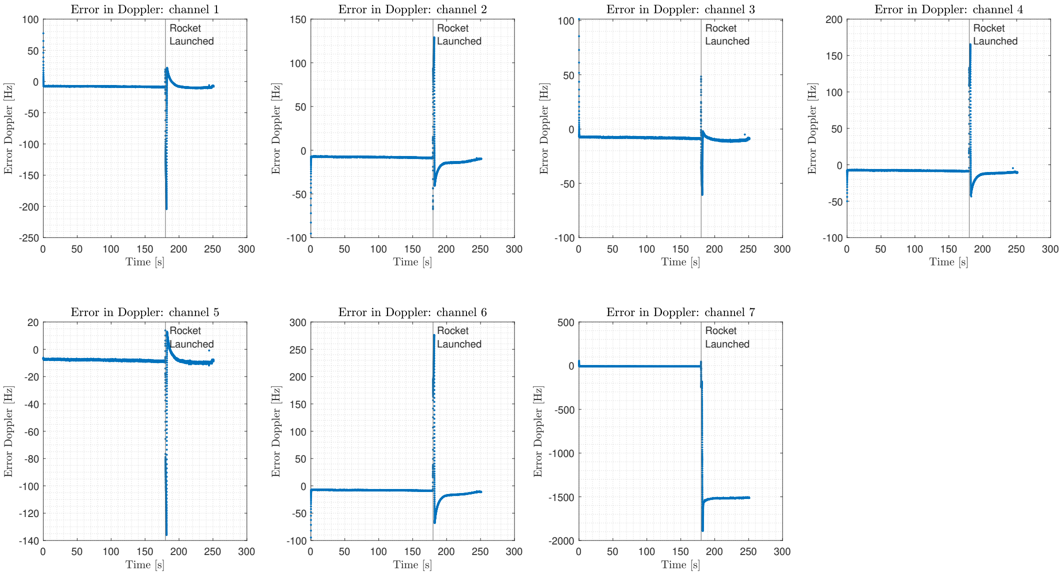


Figure 7.9: Doppler Error on channel Kalman Filter.

This architecture improves the Doppler estimation with respect to the classical approach (Figure 7.8) in the estimation of Dopplers per satellite. In this case, compared to the previous estimates, Doppler is estimated throughout the flight. This improvement in Doppler shift estimation allows for better SNR than in the previous case (constellation in Figure 7.10), although it is not enough for a clear satellite and barely maintains the phase and code discriminators at appropriate thresholds.

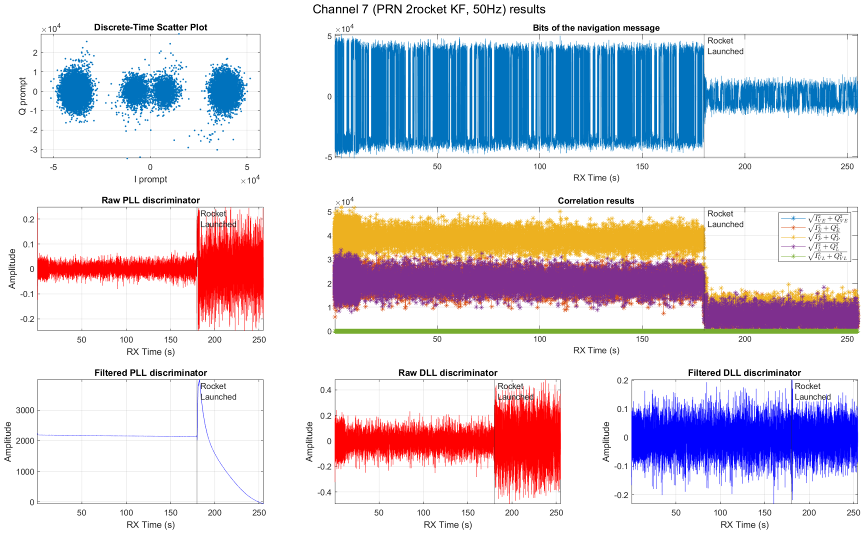


Figure 7.10: Tracking results on channel Kalman Filter.

As can be seen in Figure 7.10, the KF-based correlator trusts the model more than the measurements. When the launch is performed, the C/N_0 decreases and, therefore (equation 5.20), the variances grow, and the filter trusts the model.

However, this model does not account for the acceleration variation experienced at launch (since it cannot measure any acceleration). For this reason, it presents a constant error of 1.5 kHz with respect to the real Doppler (Figure 7.9). It is not possible to decode the message bits, but the lock is not lost. Since the rocket does not stop accelerating, it is not possible to know if the channel would return to the proper lock when this movement ceases. The variances of this KF model (diagonal elements of the Q matrix) are adjusted for all satellites together, considering the expected dynamics for all of them. Adjusting the filter to track the relative dynamics of this specific SV separately would worsen the tracking conditions for the remaining SVs in view.

7.2.4 Vector tracking loop performance

Vector tracking loop with rocket model

In Figure 7.11, it can be observed how feeding back the PVT solution, using a rocket 3DoF model to calculate the acceleration, improves Doppler feedback estimation compared to the open-loop versions for SV PRN 2. In this case, compared to the previous estimations, the Doppler is estimated along the entire flight, contrary to the classical architecture, and better than the open-loop KF tracking.

This improvement in Doppler shift estimation allows for better SNR than in the previous case (constellation in Figure 7.13), although it is enough for a clear satellite bit navigation message, and maintains the phase and code discriminators at appropriate thresholds.

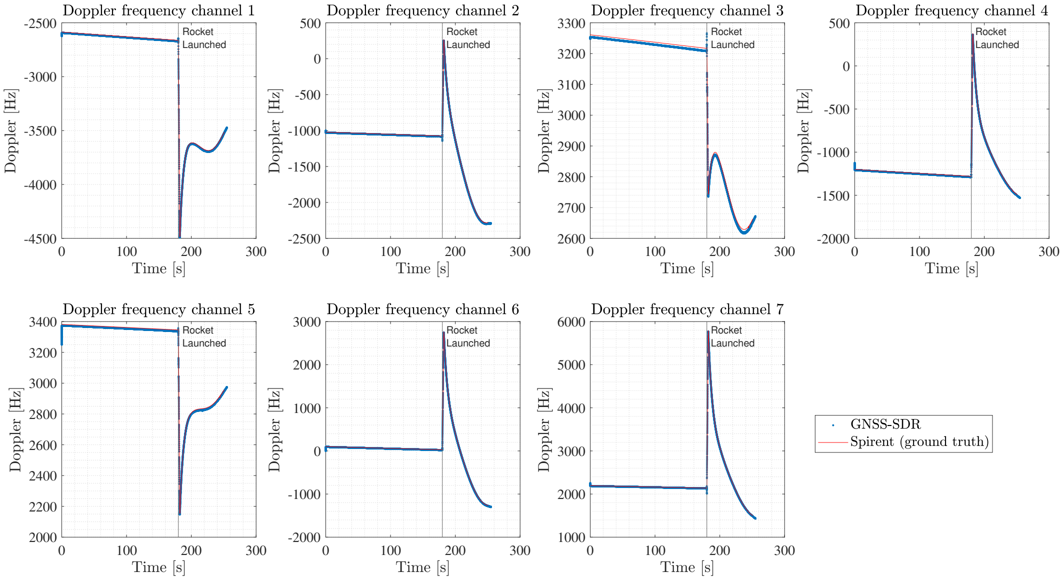


Figure 7.11: Doppler results on VTL aided by the rocket model.

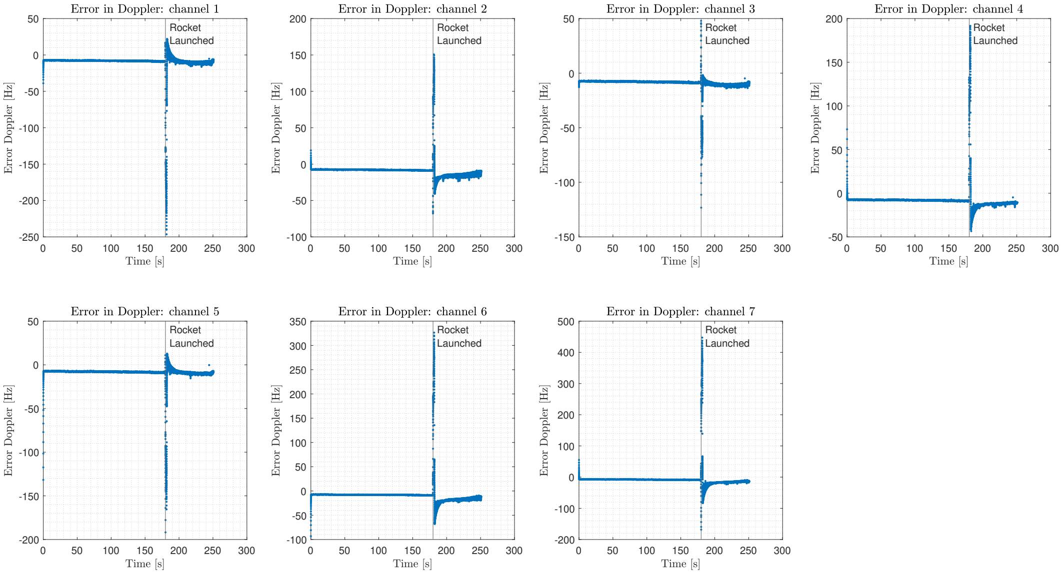


Figure 7.12: Doppler error on VTL aided by the rocket model.

This architecture provides an accurate estimation of the Doppler within the whole flight. This information can be used to calculate the rocket's trajectory, adjust its flight path, and ensure that it reaches its target accurately and safely on guided rockets. The summary of the behavior of the different architectures is recapitulated in Figure 7.14

Moreover, the use of GNSS for providing information on the rocket's attitude, including roll, pitch, and yaw, requires an adequate lock on the SV signals. and it is used for stabilizing

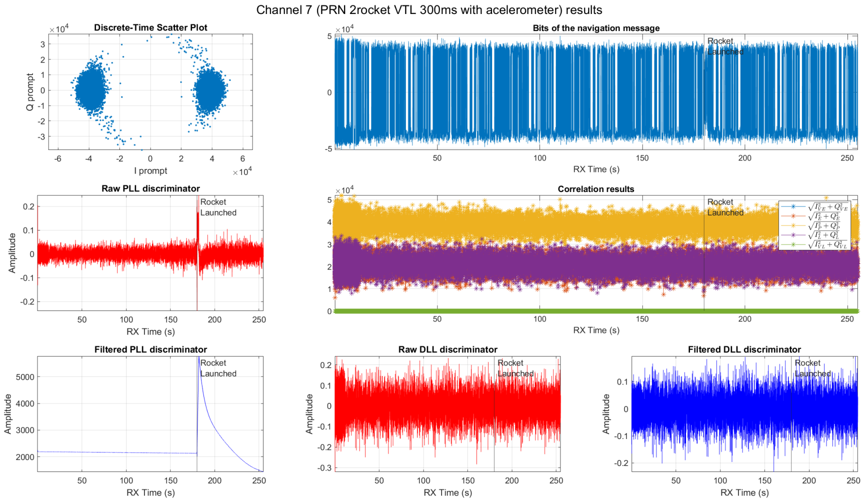


Figure 7.13: Tracking results on VTL aided by the rocket model.

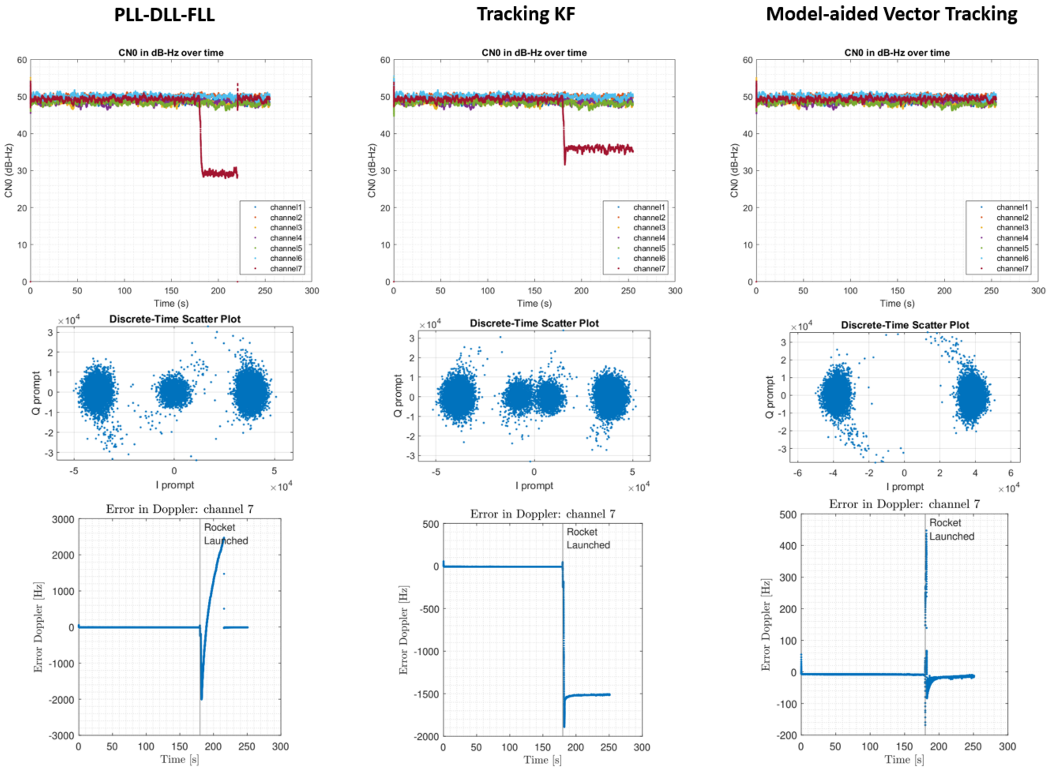


Figure 7.14: Doppler error and constellation plot of the signal on the different architectures for SV 2.

the rocket and ensuring it maintains its orientation during flight. In addition, it is worth mentioning that the stress caused to the GNSS receiver frequency oscillators in high acceleration environments is perceived as an additional Doppler drift. This effect could be significant, depending on the quality of the oscillator and how it is mounted in the rocket. If the receiver is not properly mounted on the rocket, it could even be a dominant process. This change in frequency and phase noise dependence with acceleration can be difficult to model, as some recent works present in [79]. The inclusion of the mechanic stress on the clock in real time architectures should be included as part of the model in the future.

Vector tracking loop with accelerometer assistance

In Figure 7.15, it can be observed that feeding back the acceleration experienced by the rocket during flight improves Doppler feedback estimation compared to Kalman Filtering tracking, and equals the behavior of the model-aided version. This is clearer in the estimation of SV PRN 2. In this case, compared to the STL estimations, the Doppler is accurately estimated throughout the entire flight. This improvement in Doppler shift estimation allows the best SNR in the channel (constellation in Figure 7.17) and maintains the phase and code discriminators at appropriate thresholds.

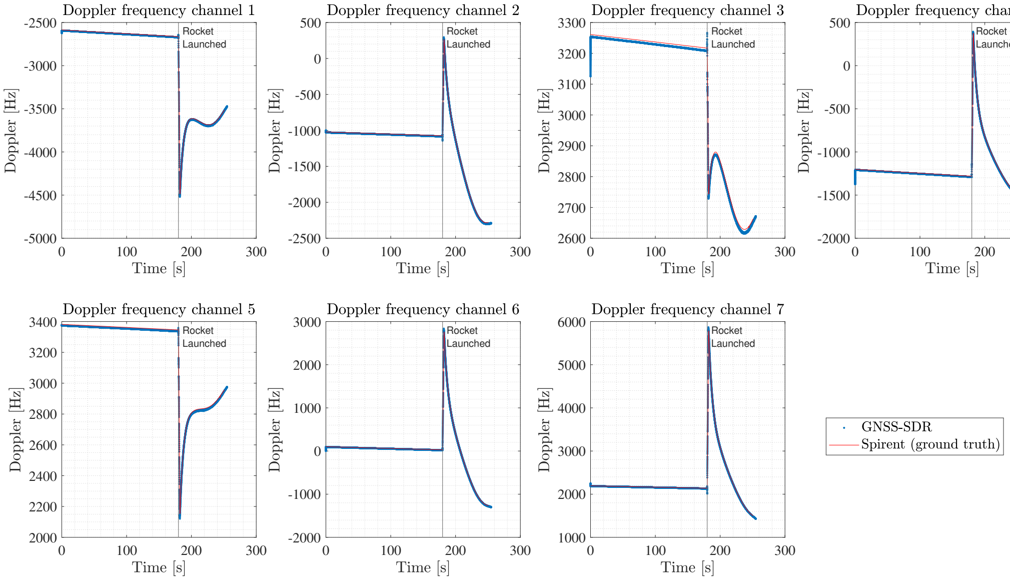


Figure 7.15: Doppler results on VTL aided by the accelerometer sensor.

This architecture presents the same behavior and results that are comparable to those of the model version. Introducing an accelerometer as an aid would be *a priori* the same as introducing it as a help in STL tracking loops.

However, it is easier for the acceleration in a real-time architecture to be introduced as presented. This aid in a STL architecture would require projecting the acceleration of the mobile in each of the tracking channels, at a different synchronization moment for each of them.

Noise in the Carrier Doppler: comparison

In the context of applying GNSS receivers to sounding rockets, it is worth highlighting that STL designs capable of handling acceleration levels of 40-50g are both feasible and well

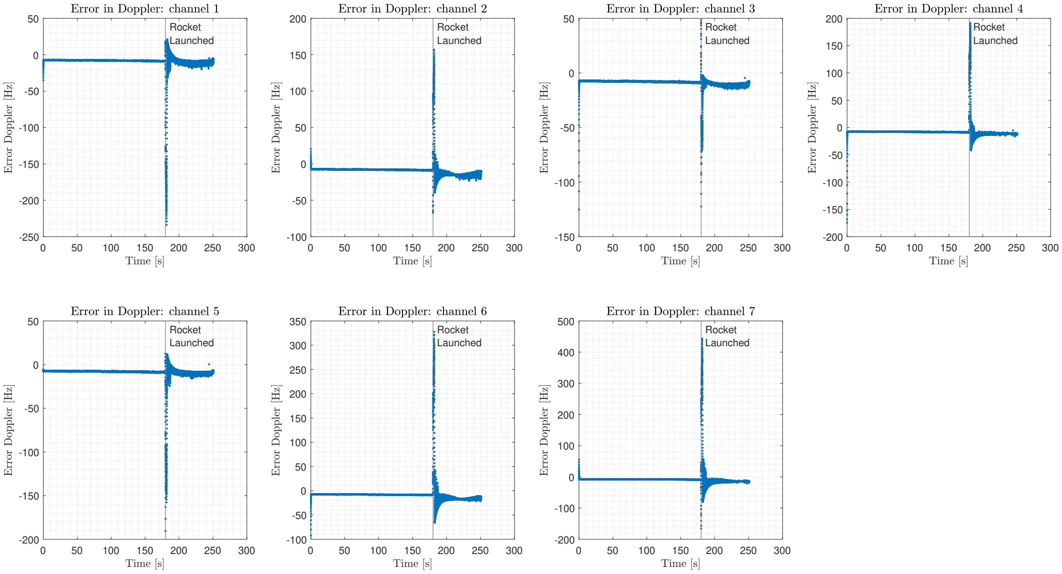


Figure 7.16: Doppler error on VTL aided by the accelerometer sensor.

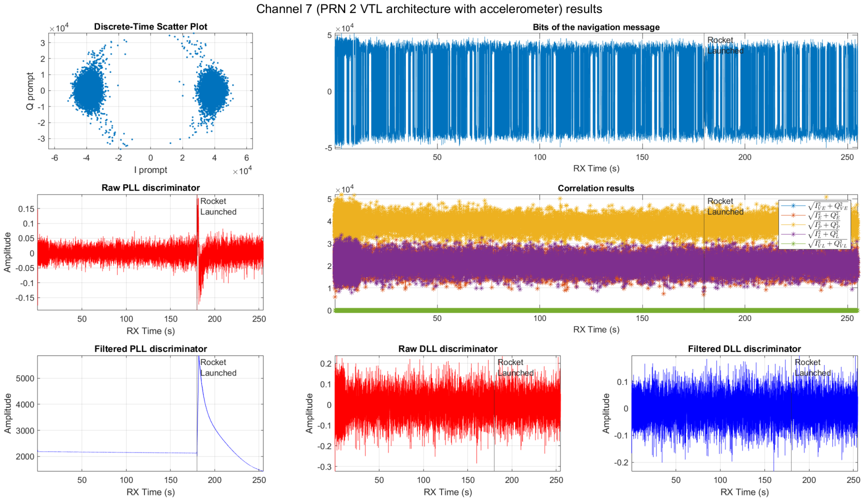


Figure 7.17: Tracking results on VTL aided by the accelerometer sensor.

documented in the literature (for example, see [103], [133], [161]). However, these designs come with trade-offs: in addition to requiring meticulous loop filter design, they necessitate the use of larger loop bandwidths, which result in noisier navigation measurements and higher tracking thresholds.

These drawbacks are mitigated in a VTL implementation. As illustrated in Figure 7.18, the PLL-FLL used under static conditions shows a variance of 5Hz, whereas Figure 7.19 demonstrates that the VTL achieves a much lower variance of 1Hz.

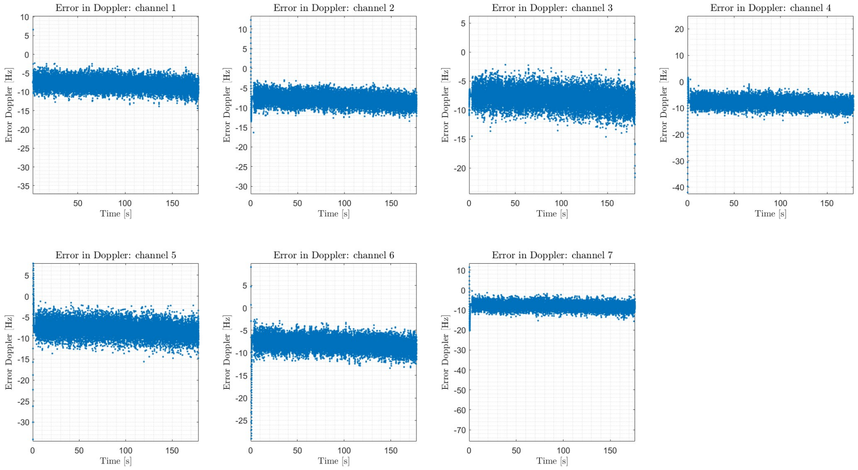


Figure 7.18: Noise of the Doppler measurements of the different SV, in the STL architecture.

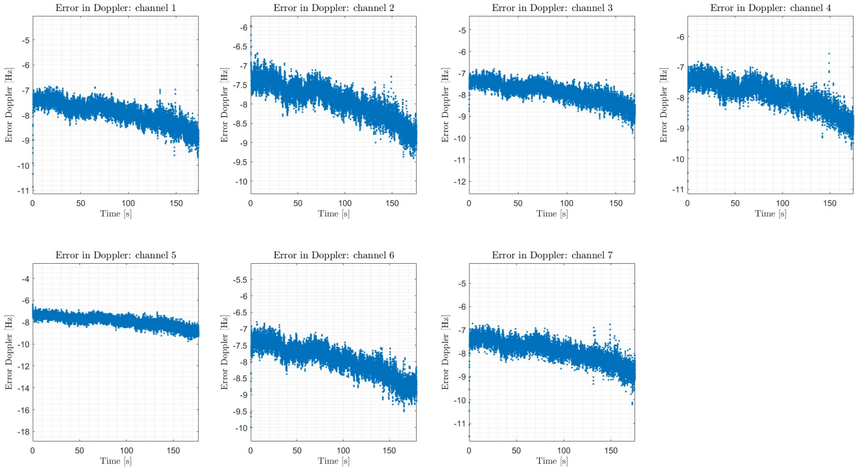


Figure 7.19: Noise of the Doppler measurements of the different SV, in the VTL architecture.

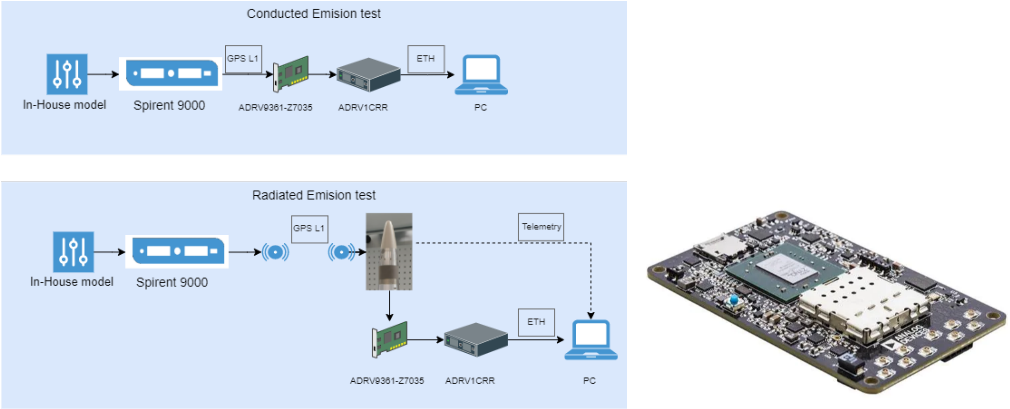
7.3 Laboratory Real Time Hardware tests

Simulations are performed with the characteristics of Table 7.2. These are the characteristics used later on the real rocket launch at the test site. In addition, the unavailability of the signal in the launch tube is emulated, since it is one of the major issues faced by the receiver. This allows us to check if the reacquisition time of the receiver will be or not enough for this scenario.

With this scenario, tests were performed (Figure 7.20). In the conducted tests, the aims were: checking the algorithm architecture, measuring the expected TTFF, validating the tracking and acquisitions loops behavior for Doppler experienced by those GNSS signals, and measuring re-acquisition time after launch tube exit.

Table 7.2: Simulated scenario parameters.

Max Acceleration	Elevation	Max Velocity	Max Roll Angular Velocity	Range
980 m/s^2	33.708 °	800 m/s	18000 deg/s	20 Km



(a) Laboratory tests configurations (b) ADRV9361 card used

Figure 7.20: Laboratory trials configuration.

The hardware used is in Figure 7.20 b), the ADRV9361 card. This card has an AD9361 with a Xilinx FPGA Z7035. The GNSS-SDR receiver is separated into the Processing System (PS) and Programmable Logic (PL). The acquisition and tracking channels are embedded in the PL, whereas the remaining parts of the receiver are embedded in the PS (Figure 4.2). In this case a Spirent 9000, with a configuration equal to the Spirent 7700 is used.

7.3.1 Acquisition Results

The TTFF is measured, which will determine the time to wait with the electronics on before introducing the rocket into the launcher. The time said for the configurations used and the architecture of the receiver (with 8 GPS L1 tracking channels and an integration time of 1 ms) is on average 2 minutes and 32 seconds with a variance of 10 seconds. After this, wait about 5 more minutes before cutting off the emitted signal, so that the receiver has acquired and is tracking at least 6 satellites. After 20 minutes with the signal cut off, the launch simulation is performed. At this moment, the receiver has a signal again, and it takes between 11 and 15 seconds to have a PVT solution again.

Table 7.3: Simulated acquisition time results.

TTFF	Time without GNSS	Reacquisiton time
212 ±10 s	1800 s	15 s (worst case)

7.3.2 Tracking Results

Figure 7.21 shows partial results of the Signal Tracking block of a specific channel for the scenario. The rocket launch is about 170th seconds, as can be seen from the Raw PLL discriminator graph. The rocket flight continues until the end of the simulation. As can be seen in the outputs of the correlators, in the middle graph right in Figure 7.21, even in the moments of the greatest acceleration the correct signal tracking is not lost. In the filtered output of the PLL (lower left graph in Figure 7.21) it can be seen how the system is able to follow the frequency changes throughout the stage.

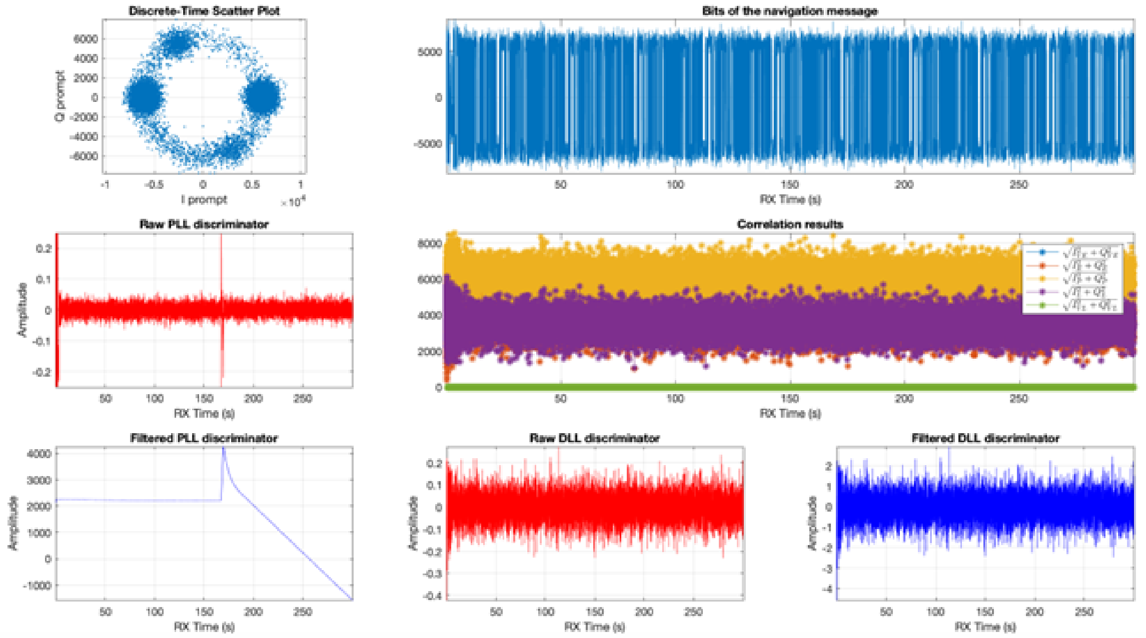


Figure 7.21: Intermediate results of the Signal Tracking block for Scenario.

7.3.3 Navigation Solution Results

With the performance of the tracking loops shown, the SDR receiver is able to demodulate the navigation messages received from the satellites and to calculate the observables (pseudorange and carrier phase), allowing one to calculate the navigation solution (position, speed, and time). This is shown in Figure 7.22.

7.4 Field tests

The focus of the tests is (*i*) checking when the solution is available, (*ii*) in the case of losing track of the navigation solution, how long it takes the system to get a new one, and (*iii*) measuring how accurate it is, whenever it is available.

Throughout the flight, key data points, including the PVT solution (position, velocity, time), rocket rotation speed, acceleration, C/N_0 and Doppler measurements from 6 of the 8 possible

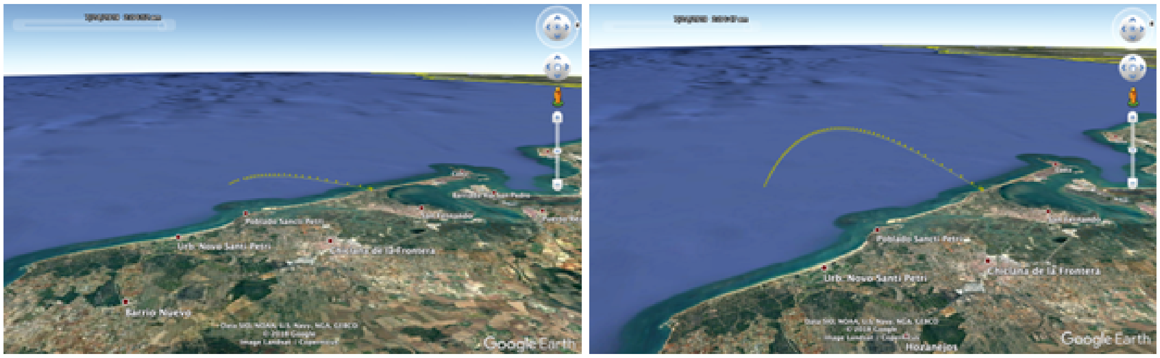


Figure 7.22: Navigation solution obtained in Scenario.

channels, were continuously recorded and transmitted by telemetry to the launch site. This approach is necessary because the rocket is launched over the sea, making it impossible to recover the electronics on board after launch. Due to the constraints of the telemetry bandwidth, no additional parameters were transmitted.

7.4.1 Acquisition Results

7.4.2 Tracking Results

The carrier-to-noise density ratio (C/N_0) of 6 satellites is recovered, as can be seen in Figure 7.23. Four of the channels are recovered approximately at the same time, around 9 seconds after launch. The other two are tracked 3 seconds later. All of them exhibit a good C/N_0 value (around 40 to 45 dB-Hz)

The Doppler of each of the six recorded satellites is shown in Figure 7.23. As can be seen, around 18 seconds after launch, the Doppler seems to remain constant. This coincides with the apogee of the flight (Figure 7.24).

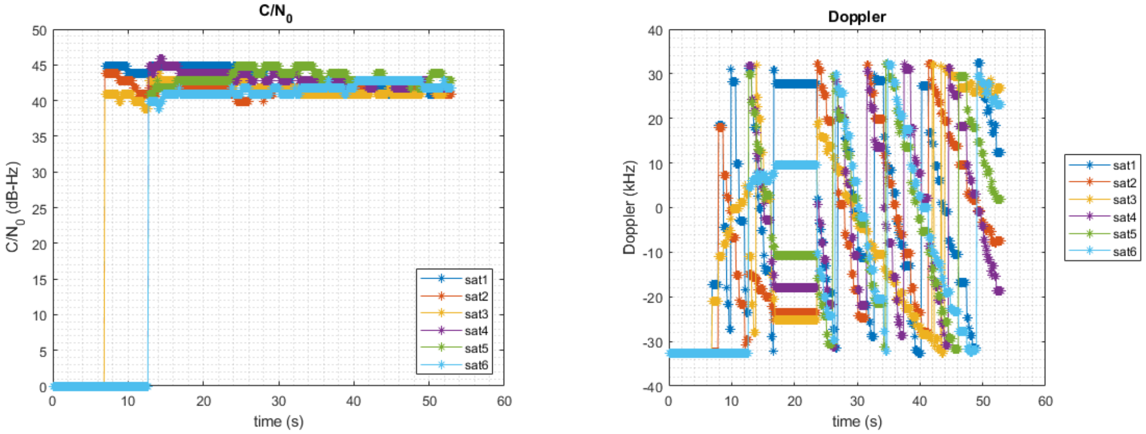


Figure 7.23: Field trials results, C/N_0 and Doppler.

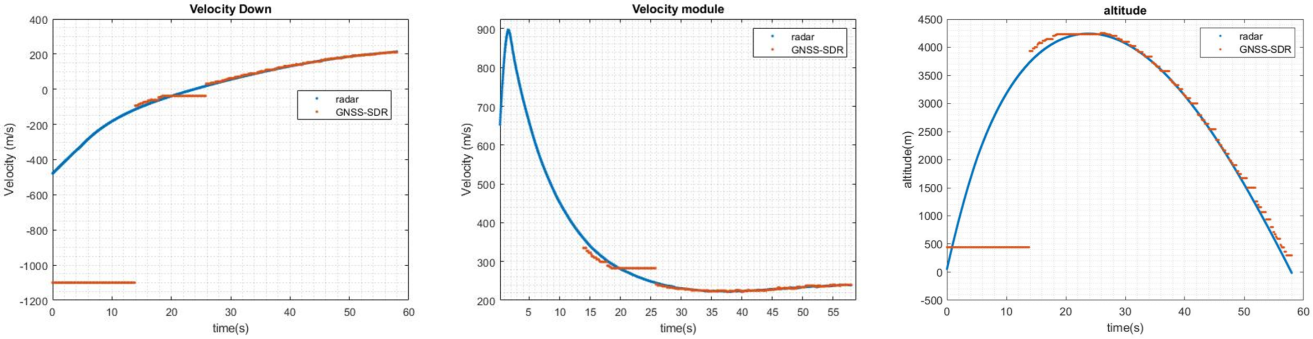


Figure 7.24: Velocity magnitude, altitude and vertical velocity of the field tests.

7.4.3 Navigation Solution Results

The GNSS-SDR trace agrees with the trajectory provided by the tracking radar in altitude, Figure 7.24), and in velocity calculated by the receiver (the same figure, vertical component and magnitude). The time it takes the receiver to achieve a valid navigation solution is around 14 seconds from the launch, which is consistent with the simulation tests (Section 5.3.2). Although it is true that later after the re-acquisition, around the apogee, there are about 5 seconds where the receiver is not able to maintain an adequate PVT solution. This, together with the Doppler values shown in the previous section, leads us to think that it is the effect of the local oscillator in the receiver, which has a drift in its frequency due to the acceleration of the rocket (both longitudinal and rotational).

Chapter 8

Conclusions

This work explored the navigation systems required for the guidance of rockets and projectiles, focusing primarily on GNSS (Global Navigation Satellite Systems) and GNSS-IMU sensor fusion methodologies. The aim is to ensure precision and reliability in the high-dynamic environments. The study has contributed to the understanding of projectile navigation, introducing both theoretical advances and practical implementations.

The research emphasizes that the distinct challenges posed by projectile dynamics, such as extreme acceleration, rapid rotational motion, and the constraints imposed by the compact size of the munitions, demand specific tailored receivers and sensors. Traditional approaches have proven inadequate, requiring the development of specific sensors or architectures. Addressing these distinct challenges led us to understand the change in the search space for signal acquisition or that special attention of the oscillator used in the receivers is needed. Moreover, new tracking architectures that can deal with intense velocities, accelerations, and jerks were explored and compared. The correct tracking loops behavior ensure that GNSS signals are captured and maintained even during the most dynamic portions of the flight, providing insight into the mechanisms through which projectiles can be guided with precision over long distances, overcoming the inherent limitations of previous systems.

The comparative analysis of different Scalar Traking Loops (STL) solutions against Vector Tracking Loop (VTL) filtered solutions shows the efficacy of these tracking algorithms in mitigating high-dynamics signal scenarios. The VTL filtered solution consistently demonstrates better performance in the SV under the worst accelerations and jerks, compared to the STL output, especially during the launch phase. This improvement in performance suggests that the VTL algorithm effectively compensates for the high dynamic environment of a rocket launch, and would be a correct choice for these scenarios. The agreement of the VTL solution with the true simulated value further validates the behavior of the VTL filtering approach in tracking precision. In terms of trajectory, the raw least squares PVT solution, from the STL measurements, exhibits a notable deviation during the early phase of the launch. This discrepancy can be attributed to the limitations of the least-squares algorithm in handling the rapid motion and complex dynamics of the rocket under high acceleration forces. In contrast, the VTL solution, designed to handle more complex tracking conditions, demonstrates much closer alignment with the ground truth, especially as the rocket moves into a more stable flight regime. The filtering process appears to significantly reduce the error margin, probing its ability to manage high frequency noise and the rapid dynamics present during the initial stage of the rocket's trajectory. The launch data also reflect the tracking errors and signal deviations inherent in real-world rocket launch conditions, with some deviation between the predicted and actual paths, particularly in the early stages of flight. Despite these challenges, the receiver provides an accurate solution, although not during the entire flight.

A critical contribution of this thesis is the examination of GNSS-SDR (Software-Defined Radio) receiver implementation, which provides a flexible and scalable solution that enhances signal processing capabilities during flight. The use of SDR in this context allows for adaptable mission-specific configurations that improve the robustness of navigation systems. Through empirical data gathered from field tests and simulations, this work shows that GNSS-SDR systems, when tuned correctly, can maintain navigation integrity throughout various flight phases: from launch to terminal guidance. In this regard, sensor fusion has emerged as an essential solution, where the combination of GNSS receivers and inertial sensors allows real-time adjustments and corrections that mitigate errors caused by external factors such as signal degradation, mechanical stress, or environmental interferences. Furthermore, research validates the effectiveness of sensor fusion navigation in modern projectile systems. By integrating GNSS and accelerometers, the system continuously estimates the position and trajectory of the projectile, aids the GNSS tracking loops in an ultra-tight integration, providing functional navigation system capable of deliver PVT solution even under the extreme conditions experienced.

In conclusion, this thesis highlights the transformative potential of GNSS-SDR and sensor fusion technologies in rocket and projectile navigation. By tackling both theoretical and practical challenges in high-dynamic environments, it not only summarizes the current state-of-the-art but also opens new pathways for innovation in the field of precision-guided munitions. Although significant progress has been made in trajectory tracking and acceleration management during rocket launches, certain aspects still require further refinement. This is particularly true in the launch phase, where discrepancies in velocity and position remain more pronounced. The VTL filtering solution has proven to be a valuable enhancement for signal tracking, but additional efforts in algorithmic optimization will be essential to ensure precise position tracking throughout the entire flight in future launches.

8.1 Future work

Looking ahead, several pathways are possible for future research and development. One of the most promising directions is the exploration of alternative signal processing techniques that could further enhance trajectory prediction and correction in real time. Signal resilience and sensor fusion will also be key enablers, particularly in environments where GNSS signals are compromised. As reliance on GNSS navigation systems grows, so does the need to mitigate vulnerabilities such as signal loss in GNSS-contested environments, where adversaries employ jamming or spoofing techniques (commonly referred to as NavWar scenarios). The development of machine learning algorithms for trajectory prediction and real-time error correction is another critical area of exploration. By leveraging neural networks and other

AI-driven techniques, navigation systems could be trained to better recognize and adapt to dynamic flight conditions. These systems would anticipate disturbances such as wind gusts or mechanical instabilities, enabling more sophisticated and autonomous trajectory adjustments during flight. Additionally, the integration of AI could significantly optimize the real-time fusion of multiple sensors, leading to more accurate and reliable navigation.

In terms of resilience to signal jamming and interference, exploring alternative signals is crucial. Low Earth Orbit (LEO) satellite constellations provide navigation capabilities, either by design as Iridium STL signal, or as a Signal-Of-Oportunity (SoOP) as Starlink. These improvements could ensure accurate, continuous navigation even when traditional GNSS signals are unavailable or compromised.

Bibliography

[1] D. Akopian, P. K. Sagiraju, and S. Turunen, “Performance of two-stage massive correlator architecture for fast acquisition of GPS signals,” in *2006 IEEE Region 5 Conference*, 2006, pp. 137–140. DOI: 10.1109/TPSD.2006.5507444.

[2] D. M. Akos, “A software radio approach to Global Navigation Satellite System receiver design,” 1997. [Online]. Available: https://web.stanford.edu/group/scpnt/gpslab/pubs/theses/DennisAkosThesis97.pdf.

[3] D. M. Akos *et al.*, “GNSS software defined radio: History, current developments, and standardization efforts,” *ION GNSS+, The International Technical Meeting of the Satellite Division of The Institute of Navigation*, 2022.

[4] Analog Devices, Inc., *ADXL375 Datasheet - 3-Axis, ±200g Digital MEMS Accelerometer*, 2014.

[5] B. Anderson and J. B. Moore, *Optimal filtering*. Englewood Cliffs, New Jersey, USA: Prentice-Hall, 1979.

[6] G. Andreas, B. Braun, and M. Markgraf, “The Kodiak GNSS Receiver for Microlaunchers and Sounding Rockets,” *42nd AAS Annual Guidance and Control Conference*, pp. 831–842, Feb. 2019.

[7] J. Arribas, “GNSS array-based acquisition: Theory and implementation,” Ph.D. dissertation, Universitat Politecnica de Catalunya, Castelldefels, Spain, 2012. [Online]. Available: https://theses.eurasip.org/media/theses/documents/arribas-javier-gnss-array-based-acquisition-theory-and-implementation.pdf.

[8] F. E. M. Avram K. Tetewsky, “Carrier phase wrap-up induced by rotatin GPS antennas,” GPS World, Tech. Rep., 1997.

[9] S. R. Babu and J. Wang, “Ultra-tight GPS/INS/PL integration: A system concept and performance analysis,” *GPS Solutions*, vol. 13, pp. 75–82, Jul. 2009. DOI: 10.1007/s10291-008-0097-9.

[10] J. Beuchert, A. Matthes, and A. Rogers, “SnapperGPS: Open hardware for energy-efficient, low-cost wildlife location tracking with snapshot GNSS,” *ArXiv*, vol. abs/2207.06310, 2022.

[11] A. A. Birklykke, Y. Le Moullec, R. Prasad, and L. Alminde, “Statistical re-acquisition method for GPS receivers on satellites in low earth orbit,” in *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)*, 2011, pp. 1–5. DOI: 10.1109/WIRELESSVITAE.2011.5940913.

[12] D. Borio, N. Sokolova, and G. Lachapelle, "Doppler measurement accuracy in standard and high–sensitivity Global Navigation Satellite System receivers," *IET Radar, Sonar & Navigation*, vol. 5, no. 6, pp. 657–665, Jul. 2011, doi: 10.1049/iet-rsn.2010.0249.

[13] K. Borre, D. Akos, N. Bertelsen, P. Rinder, and S. Jensen, *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach* (Applied and Numerical Harmonic Analysis). Birkhäuser Boston, 2007.

[14] L. L. Boyer, "Carrier-phase wrap-up caused by rotating a Global Positioning System antenna and its effect on measurements," 1999.

[15] J. A. C. Bradford S. Davis and T. G. Brown, "Flight demonstration results of an Inertial Measurement Unit and Global Positioning System translator telemetry system," Army Research Laboratory, Tech. Rep., 2001.

[16] A. C. Bragado, A. Solera-Rico, and M. A. Gómez, "Implementation of trajectory propagator for artillery projectiles based on artificial neural networks," in *New Technologies and Developments in Unmanned Systems*, Springer International Publishing, 2023, pp. 187–192. DOI: 10.1007/978-3-031-37160-8_29.

[17] T. G. Brown and B. S. Davis, "Dynamic high-g loading of MEMS sensors: Ground and flight testing," in *Photonics West - Micro and Nano Fabricated Electromechanical and Optical Components*, 1998.

[18] T. G. Brown *et al.*, "Strap-down microelectromechanical (MEMS) sensors for high-g munition applications," *IEEE Transactions on Magnetics*, vol. 37, pp. 336–342, 2001.

[19] B. Burchett and M. Costello, "Specialized Kalman filtering for guided projectiles," in *39th Aerospace Sciences Meeting and Exhibit*. 2001. DOI: 10.2514/6.2001-1120.

[20] C. Fernández–Prades, J. Vilà–Valls, J. Arribas, and A. Ramos, "Continuous reproducibility in GNSS signal processing," *IEEE Access*, vol. 6, no. 1, pp. 20 451–20 463, Apr. 2018. DOI: 10.1109/ACCESS.2018.2822835.

[21] J. S. C.G. Maple, "Aerodynamic symmetry of projectiles," *Quarterly of Applied Mathematics*, vol. 6, no. 4, pp. 345–366, Jan. 1949. [Online]. Available: https://www.jstor.org/stable/43633687.

[22] C. Cahn, D. Leimer, C. Marsh, F. Huntowski, and G. LaRue, "Software Implementation of a PN Spread Spectrum Receiver to Accommodate Dynamics," *IEEE Transactions on Communications*, vol. 25, no. 8, pp. 832–840, 1977.

[23] R. de Celis and L. Cadarso, "Attitude determination algorithms through accelerometers, GNSS sensors and gravity vector estimator," *International Journal of Aerospace Engineering*, vol. 2018, pp. 1–14, 2018.

[24] R. de Celis and L. Cadarso, "Development of a neural network-based controller for short-range rockets," *2021 Winter Simulation Conference (WSC)*, pp. 1–12, 2021.

[25] R. de Celis, L. Cadarso, and J. Sánchez, "Guidance and control for high dynamic rotating artillery rockets," *Aerospace Science and Technology*, vol. 64, pp. 204–212, 2017. DOI: 10.1016/j.ast.2017.01.026.

[26] R. de Celis, P. Solano-López, and L. Cadarso, "A neural network for sensor hybridization in rocket guidance," *2020 Winter Simulation Conference (WSC)*, pp. 2006–2017, 2020.

[27] K. Chiang, M. Psiaki, S. Powell, R. Miceli, and B. O'Hanlon, "GPS-based attitude determination for a spinning rocket," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 4, pp. 2654–2663, 2014.

[28] T.-Y. Chiou, D. Gebre-Egziaber, T. Walter, and P. Enge, "Model analysis on the performance for an intertial aided FLL-assisted PLL carrier tracking loop in the presence of ionospheric scintillation," in *Proc. of the ION NTM*, Jan. 2007.

[29] Committee on the History of Japan's Trade and Industry Policy RIETI, "Long-Term Depression and Economic Structural Reform (1991-2000)," *Dynamics of Japan's Trade and Industrial Policy in the Post Rapid Growth Era (1980–2000)*, pp. 203–308, 2020.

[30] G. R. Cooper, F. E. Fresconi, and M. Costello, "Flight stability of an asymmetric projectile with activating canards," *Journal of Spacecraft and Rockets*, vol. 49, pp. 130–135, 2012.

[31] I. Cortes *et al.*, "Robust Tracking Strategy for Modern GNSS Receivers in Sounding Rockets," *2022 10th Workshop on Satellite Navigation Technology (NAVITEC)*, pp. 1–7, 2022.

[32] M. Costello and R. Letniak, "A nonlinear model predictive observer for smart projectile applications," 2007.

[33] H. Cramér, *Mathematical Methods of Statistics*. Princeton, New Jersey, USA: Princeton Univ. Press, 1946.

[34] J. T. Curran, M. Bavaro, and J. Fortuny-Guasch, "An open-loop vector receiver architecture for GNSS-based scintillation monitoring," in *Proc. of the European Navigation Conference, ENC-GNSS*, Dec. 2014.

[35] W. P. D'amico, W. H. Clay, L. W. Burke, and E. H. Baur, "In-flight instrumentation and launch system for anti-personnel grenades.," 1993.

[36] B. Davis, "Using low-cost MEMS accelerometers and gyroscopes as strapdown IMUs on rolling projectiles," in *IEEE 1998 Position Location and Navigation Symposium (Cat. No.98CH36153)*, 1998, pp. 594–601. DOI: `10.1109/PLANS.1998.670218`.

[37] B. S. Davis, J. A. Condon, and T. G. Brown, "Flight demonstration results of an inertial measurement unit and global positioning system translator telemetry system," 2001.

[38] G. C. Diaz, "Organización y empleo de una ubeaca basada en artillería de cohete en el contexto de una operación WAS," 2016.

[39] D. Díaz, "The SILAM program," Revista Ejércitos, Tech. Rep., 2024.

[40] EDA, "Course correction fuze integration into artillery," European Defence Agency, Tech. Rep., 2017. [Online]. Available: `https://www.eda.europa.eu/docs/default-source/brochures/course-correction-fuze-(ccf)-project_factsheet.pdf`.

[41] ESA, "Galileo open service signal in space interface control document - draft 1," European Space Agency / Galileo Joint Undertaking, Tech. Rep., 2008.

[42] *Ettus Research USRP X310 Product Specification*, 2023. [Online]. Available: `https://support.spirent.com/SC_KnowledgeView?id=DOC12114`.

[43] L. Fairfax, F. Fresconi, and M. Costello, "Affordable state estimation using Inertial Navigation Sensor arrays," in *AIAA Atmospheric Flight Mechanics Conference*. 2011. DOI: `10.2514/6.2011-6267`.

[44] L. D. Fairfax, Fresconi, and F. E., "Loosely-coupled GPS/INS state estimation in precision projectiles," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, 2012, pp. 620–624. DOI: `10.1109/PLANS.2012.6236936`.

[45] R. Falone, C. Stallo, E. Gambi, and S. Spinsante, "Sdr GNSS receivers: A comparative overview of different approaches," *2014 IEEE Metrology for Aerospace (MetroAeroSpace)*, pp. 326–331, 2014.

[46] J. Farrell, *The Global Positioning System & Inertial Navigation*. McGraw-Hill Education, 1999.

[47] J. Farrell, *Aided Navigation: GPS with High Rate Sensors* (McGraw-Hill professional engineering: Electronic engineering). McGraw Hill LLC, 2008.

[48] X. Feng, T. Zhang, T. Lin, H. Tang, and X. Niu, "Implementation and performance of a deeply-coupled GNSS receiver with low-cost MEMS inertial sensors for vehicle urban navigation," *Sensors*, vol. 20, no. 12, 2020. DOI: `10.3390/s20123397`.

[49] E. M. Ferguson, L. W. Burke, M. Holis, C. Myers, and E. S. Irwin, "Hardened subminiature telemetry and sensor systems (HSTSS) subsystems flight test results.," Tech. Rep., 1996.

[50] C. Fernandez-Prades, J. Arribas, and P. Closas, "Accelerating GNSS software receivers," in *Proc. of the ION GNSS+*, Portland, Oregon, Sep. 2016, pp. 44–61. DOI: `10.33012/2016.14576`.

[51] C. Fernandez-Prades *et al.*, "Assisted GNSS in LTE-advanced networks and its application to vector tracking loops," in *Proc. of the ION GNSS*, vol. 2, Nashville, TN, Sep. 2012, pp. 1462–1476.

[52] C. Fernández-Prades, L. Lo Presti, and E. Falletti, "Satellite radiolocalization from GPS to GNSS and beyond: Novel technologies and applications for civil mass market," *Proceedings of the IEEE*, vol. 99, no. 11, pp. 1882–1904, Nov. 2011. DOI: `10.1109/JPROC.2011.2158032`.

[53] C. Fernández-Prades, J. Arribas, and P. Closas, "Turning a television into a GNSS receiver," in *ION GNSS+ 2013*, 2013.

[54] F. Fresconi, "Guidance and control of a fin-stabilized projectile based on flight dynamics with reduced sensor and actuator requirements," Army Research laboratory, Tech. Rep., Feb. 2011. DOI: `10.21236/ada539868`.

[55] F. Fresconi, G. Cooper, and M. Costello, "Practical assessment of real-time impact point estimators for smart weapons," *Journal of Aerospace Engineering*, vol. 24, no. 1, pp. 1–11, Jan. 2011. DOI: `10.1061/(asce)as.1943-5525.0000044`.

[56] F. E. Fresconi and P. Plostins, "Control mechanism strategies for spin-stabilized projectiles," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 224, pp. 979–991, 2009.

[57] F. E. Fresconi and J. Rogers, "Flight control of a small-diameter spin-stabilized projectile using imager feedback," *Journal of Guidance Control and Dynamics*, vol. 38, pp. 181–191, 2015.

[58] G. Frost and M. Costello, "Linear theory of a rotating internal part projectile configuration in atmospheric flight," *Journal of Guidance Control and Dynamics*, vol. 27, pp. 898–906, 2004.

[59] G. Kahn and D. B. MacQueen, "Coroutines and networks of parallel processes," in *Proceedings of Information Processing*, Toronto, Aug. 1977, pp. 993–998.

[60] E. Gagnon and A. Vachon, "Efficiency analysis of canards-based course correction fuze for a 155-mm spin-stabilized projectile," *Journal of Aerospace Engineering*, vol. 29, no. 6, p. 04 016 055, 2016. DOI: `10.1061/(asce)as.1943-5525.0000634`.

[61] A. Gamble and P. Jenkins, “Low cost guidance for the Multiple Launch Rocket System (MLRS) artillery rocket,” in *IEEE 2000. Position Location and Navigation Symposium (Cat. No.00CH37062)*, Mar. 2000, pp. 193–199. DOI: 10.1109/PLANS.2000.838302.

[62] G. Gao and G. Lachapelle, “A novel architecture for ultra-tight HSGPS-INS integration,” *Journal of Global Positioning Systems*, vol. 01, 2008. DOI: 10.5081/JGPS.7.1.46.

[63] E. K. A. Gill and D. Akos, “Snapshot GNSS receivers for low-effort, high-gain space situational awareness,” *Advances in Space Research*, 2023.

[64] GNSS-SDR. “GNSS-SDR documentation.” (2021), [Online]. Available: https://gnss-sdr.org/docs/.

[65] *GNU Radio*, Online: https://www.gnuradio.org/. Accessed: September 29, 2024, 2023.

[66] M. Gomez, D. Galindo, J. Arribas, and C. Fernandez-Prades, “GNSS receiver developed on a SDR platform withstands high accelerations and speeds,” in *Proc. of the ION GNSS+*, doi: 10.33012/2022.18396, Denver, CO, Sep. 2022, pp. 3333–3339.

[67] M. Gomez, A. Solera-Rico, E. Valero, J. Arribas Lazaro, and C. Fernandez-Prades, “Enhancing GNSS Receiver Performance with Software-Defined Vector Carrier Tracking for Rocket Launching,” *Results in Engineering*, pp. 101 310–101 325, 2023.

[68] “GPS signal specification,” Global Positioning System Standard Positioning Service, Tech. Rep., 1993. [Online]. Available: https://www.%7BGPS%7D.gov/technical/ps/1993-SPS-signal-specification.pdf.

[69] A. Grenier, E. S. Lohan, A. Ometov, and J. Nurmi, “An Open-source Software-Defined Receiver for GNSS Algorithms Benchmarking,” *14th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 31–38, 2022.

[70] M. S. Grewal, L. R. Weill, and A. P. Andrews, Eds., *Global Positioning Systems, inertial navigation and integration*, 2nd. John Wiley and Sons, Inc., 2007.

[71] M. S. Grewal, A. P. Andrews, and C. G. Bartone, “Global Navigation Satellite Systems, Inertial Navigation and Integration,” pp. 249–292, 2020.

[72] P. Groves, C. Mather, and A. Macaulay, “Demonstration of non-coherent deep INS/GPS integration for optimized signal-to-noise performance,” in *Proc. ION GNSS*, vol. 3, Sep. 2007, pp. 2627–2638.

[73] L. C. Hainz and M. Costello, “Modified projectile linear theory for rapid trajectory prediction,” *Journal of Guidance, Control and Dynamics*, vol. 28, no. 5, pp. 1006–1014, Sep. 2005. DOI: 10.2514/1.8027.

[74] T. E. Harkins, “Understanding body-fixed sensor output from projectile flight experiments,” Army Research Laboratory, Tech. Rep., 2003.

[75] P. Henkel, K. Giger, and C. Gunther, “Multifrequency, multisatellite vector phase-locked loop for robust carrier tracking,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 4, pp. 674–681, Aug. 2009. DOI: 10.1109/JSTSP.2009.2025637.

[76] A. E. Hodapp, “Passive means for stabilizing projectiles with partially restrained internal members,” *Journal of Guidance Control and Dynamics*, vol. 12, pp. 135–139, 1989.

[77] W. Hurd, J. Statman, and V. Vilnrotter, “High dynamic GPS receiver using maximum likelihood estimationand frequency tracking,” *IEEE Transactions on Aerospace and*

Electronic Systems, vol. AES-23, no. 4, pp. 425–437, 1987. DOI: 10.1109/TAES.1987.310876.

[78] Ifen, GmbH, *SX3 GNSS Receiver*, Datasheet, Poing, Germany, Apr. 2023. [Online]. Available: `https://www.ifen.com/fileadmin/documents/pdf/SX3/SX3-Flyer_Mar2023.pdf`.

[79] A. Jain and S. Schon, "Influence of receiver clock modeling in GNSS-based flight navigation: Concepts and experimental results," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 208–218. DOI: 10.1109/PLANS46316.2020.9109925.

[80] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York, USA: Academic Press, 1970.

[81] E. Johnson, A. Calise, and J. Corban, "A Six Degree-of-Freedom Adaptive Flight Control Architecture for Trajectory Following," *AIAA Guidance,Navigation and Control Conference*, Oct. 2002.

[82] J. Juang, "Multi-objective approach in GNSS code discriminator design," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 481–492, Apr. 2008, doi: 10.1109/TAES.2008.4560201.

[83] J. Juang, Y.-H. Chen, T.-L. Kao, and Y.-F. Tsai, "Design and implementation of an adaptive code discriminator in a DSP/FPGA-based Galileo receiver," *GPS Solutions*, vol. 14, no. 3, pp. 255–266, Jun. 2010. DOI: 10.1007/s10291-009-0141-4.

[84] E. D. Kaplan and C. Hegarty, Eds., *Understanding GPS: Principles and applications*, 2nd. Artech House, 2006.

[85] E. D. Kaplan and C. Hegarty, Eds., *Understanding GPS/GNSS: Principles and applications*, 3rd. Norwood, MA: Artech House, 2017.

[86] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, New Jersey, USA: Prentice-Hall, 1993.

[87] S. Khambra, C. Kumar, D. Chatterjee, and B. Mondal, "Predicting the flight behaviour of a guided projectile through a six degrees of freedom trajectory model," in *Fluid Mechanics and Fluid Power, Volume 6*, K. M. Singh, S. Dutta, S. Subudhi, and N. K. Singh, Eds., Singapore: Springer Nature Singapore, 2024, pp. 515–529.

[88] R. R. Khan, A. Hanif, and Q. Ahmed, "Threat analysis of position, navigation, and timing for highly automated vehicles," in *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2023, pp. 647–659. DOI: 10.1109/PLANS53410.2023.10140072.

[89] M. Lashley, D. M. Bevly, and J. Y. Hung, "Performance analysis of vector tracking algorithms for weak GPS signals in high dynamics," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 4, pp. 661–673, Aug. 2009, doi: 10.1109/JSTSP.2009.2023341.

[90] P. Lian, "Improving tracking performance of PLL in high dynamic applications," Ph.D. dissertation, University of Calgary, Calgary, Canada, 2004.

[91] R. F. Lieske and M. L. Reiter, "Equations of motion for a modified point mass trajectory," *Ballistics Research Laboratory Reports*, no. 1314, 1966. DOI: 10.21236/ad0485869.

[92] K. Lloyd and D. Brown, "Instability of spinning projectiles during terminal guidance," *Journal of Guidance and Control*, vol. 2, no. 1, pp. 65–70, 1979. DOI: 10.2514/3.55833.

[93] C. D. V. López, “The FGK (Fuze Guidance Kit) program,” Revista Ejércitos, Tech. Rep., 2023.

[94] C. O. M.G. Petovello and G. Lachapelle, “Ultra-Tight GPS/INS for Carrier Phase Positioning in Weak-Signal Environments,” *Meeting Proceedings RTO-MP-SET-104*, pp. 30-1–30-18, Sep. 2007.

[95] J. A. MA Gomez David Galindo and C. Fernandez-Prades, “Receptor GNSS para dinámicas con altas aceleraciones y velocidades, desarrollado sobre una plataforma SDR,” in *Proc. DseI+D 2019*, Cadiz, Spain, Nov. 2019.

[96] M. Markgraf, “The Phoenix GPS Receiver for Rocket and Satellite Applications: An Example for the Successful Utilization of COTS Technology in Space Projects,” in *Radiation Effects on Integrated Circuits and Systems for Space Applications*, R. Velazco, D. McMorrow, and J. Estela, Eds. Springer International Publishing, 2019, pp. 347–367.

[97] C. McConaghy, “Doppler Tracking,” *QEX-ARR March/April 2010*, pp. 30–38, Apr. 2010.

[98] R. McCoy, *Modern exterior ballistics: The launch and flight dynamics of symmetric projectiles*, 1st. Atglen, PA: Schiffer Publishing, 1999.

[99] R. A. H. Michael J. Wilson and M. Ilg, “Onbord (On-board Navigation of Ballistic ORDnance): Gun-launched munitions flight controller,” Army Research Laboratory, Tech. Rep., 2004.

[100] Microchip, Inc., *TX-552 Datasheet - Temperature Compensated Crystal Oscillator*, 2024.

[101] Microchip, Inc., *TX-802A Datasheet - Temperature Compensated Crystal Oscillator*, 2024.

[102] M. Milla, H. Boeglen, L. Bernard, D. Schmoltzi, and R. Vauzelle, “Analysis of the doppler shift due to pendulation and static spinning for projectile antennas,” *2016 IEEE 27th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6, 2016.

[103] N. S. Miller *et al.*, “SNAP: A Xona space systems and GPS Software-Defined Receiver,” in *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2023, pp. 897–904. DOI: 10.1109/PLANS53410.2023.10139956.

[104] R. Minor and D. W. Rowe, “Utilization of GPS/MEMS-IMU for measurement of dynamics for range testing of missiles and rockets,” *IEEE 1998 Position Location and Navigation Symposium (Cat. No.98CH36153)*, pp. 602–607, 1998.

[105] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2006.

[106] J. Mitola, “Software radio architecture evolution: Foundations, technology tradeoffs and architecture implications,” *IEICE Transactions on Communications*, vol. E83B, pp. 1165–1173, Jun. 2000.

[107] O. Montenbruck *et al.*, “Position-Velocity Aiding of a Mitel ORION Receiver for Sounding-Rocket Tracking,” *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000)*, pp. 2003–2008, Jan. 2000.

[108] O. Montenbruck *et al.*, “High Dynamics and False Lock Resistant GNSS Carrier Tracking Loops,” *Proceedings of the 20th International Technical Meeting of the*

Satellite Division of The Institute of Navigation (ION GNSS 2007), pp. 2364–2375, 2007.

[109] P. H. Morrison, “A lesson learned about cannon-launched guided projectiles,” 1978.

[110] P. H. Morrison and D. S.Amberntson, “Guidance and control of a cannon-launched guided projectile,” *Journal of Spacecraft and Rockets*, vol. 14, pp. 328–334, 1977.

[111] C. Murphy, “Instability of controlled projectiles in ascending or descending flight,” *Journal of Guidance and Control*, vol. 4, no. 1, pp. 66–69, 1981. DOI: `10.2514/3.19716`.

[112] C. H. Murphy, “Influence of moving internal parts on angular motion of spinning projectiles,” *Journal of Guidance Control and Dynamics*, vol. 1, pp. 117–122, 1977.

[113] H. Nichols, M. J. Murrian, and T. E. Humphreys, “Software-defined GNSS is ready for launch,” *ION GNSS+, The International Technical Meeting of the Satellite Division of The Institute of Navigation*, 2022.

[114] NSO, “Open system architecture interface to enable simulator laboratory test of integrated global positioning system (GPS),” North Atlantic Treaty Organization, STANAG 4572, 2007.

[115] NSO, “An engineering model to estimate aerodynamic coefficients,” North Atlantic Treaty Organization, STANAG 4655, 2010.

[116] D. Ollerenshaw and M. Costello, “Model predictive control of a direct fire projectile equipped with canards,” *Journal of Dynamic Systems, Measurement and Control*, vol. 130, no. 6, 2008. DOI: `10.1115/1.2957624`.

[117] D. Ollerenshaw and M. Costello, “On the swerve response of projectiles to control input,” Army Research laboratory, Tech. Rep., 2008.

[118] D. Ollerenshaw and M. Costello, “Simplified projectile swerve solution for general control inputs,” *Journal of Guidance, Control and Dynamics*, 2008.

[119] P. D. Groves, *Principles of GNSS, Inertial and Multisensor Integrated Navigation Systems, 2nd Ed.* Norwood, United States: Artech House, 2013.

[120] K. B. Pamadi, J. Ohlmeyer, and T. R. Pepitone, “Assessment of a GPS guided spinning projectile using an accelerometer-only IMU,” 2004.

[121] B. Parkinson and J. Spilker, *Global Positioning System: Theory and Applications.* American Institute of Aeronautics & Astronautics, 1996.

[122] M. Petovello and G. Lachapelle, “Comparison of vector-based software receiver implementations with application to ultra-tight GPS/INS integration,” Proc. of the 19st International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS, Fort Worth, TX, Sep. 2006, pp. 1790–1799.

[123] M. Petovello, “What are vector tracking loops and what are their benefits and drawbacks?” *Inside GNSS*, 2009.

[124] M. G. Petovello, “Real-time integration of a tactical-grade IMU and GPS for high-accuracy positioning and navigation,” 2003.

[125] L. L. Presti, P. di Torino, E. Falletti, M. Nicola, and M. T. Gamba, “Software defined radio technology for GNSS receivers,” in *2014 IEEE Metrology for Aerospace (MetroAeroSpace)*, 2014, pp. 314–319. DOI: `10.1109/MetroAeroSpace.2014.6865941`.

[126] F. Principe, G. Bacci, F. Giannetti, and M. Luise, “Software-Defined Radio Technologies for GNSS Receivers: A Tutorial Approach to a Simple Design and Implementation,” *International Journal of Navigation and Observation*, vol. 27, Jan. 2011.

[127] A. Qi, L. Chuanjun, and A. Hongfei, “Review on vector tracking application to GNSS receiver,” in *2015 12th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, vol. 01, 2015, pp. 304–309. DOI: 10.1109/ICEMI.2015.7494184.

[128] L. Qian, G. Chen, M. Tong, and J. Tang, “General design principle of artillery for firing accuracy,” *Defence Technology*, vol. 18, no. 12, pp. 2125–2140, 2022. DOI: 10.1016/j.dt.2022.09.001.

[129] B. Ristic, S. Arulampalam, and N. Gordon, Eds., *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, Boston, 2004.

[130] J. Rogers and M. Costello, “Flight dynamics and control authority of a projectile equipped with a controllable internal translating mass,” 2007.

[131] J. Rogers and M. Costello, “A low-cost orientation estimator for smart projectiles using magnetometers and thermopiles,” *NAVIGATION*, vol. 59, no. 1, pp. 9–24, 2012. DOI: 10.1002/navi.5.

[132] J. Rogers, M. Costello, T. Harkins, and M. Hamaoui, “Effective use of magnetometer feedback for smart projectile applications,” *Navigation*, vol. 58, Sep. 2011. DOI: 10.1002/j.2161-4296.2011.tb02581.x.

[133] P. A. Roncagliolo, C. E. De Blasis, and C. H. Muravchik, “GPS Digital Tracking Loops Design for High Dynamic Launching Vehicles,” *2006 IEEE Ninth International Symposium on Spread Spectrum Techniques and Applications*, pp. 41–45, 2006.

[134] P. A. Roncagliolo, J. G. García, and C. H. Muravchik, “Optimized carrier tracking loop design for real-time high-dynamics GNSS receivers,” *International Journal of Navigation and Observation*, vol. 2012, pp. 1–18, 2012. DOI: 10.1155/2012/651039.

[135] E. Roussel, “Control moment gyro actuator for a guided projectile,” *Journal of Guidance, Control and Dynamics*, 2023.

[136] A. Roux, S. Changey, J. Weber, Lauffenburger, and Jean-Philippe, “LSTM-based projectile trajectory estimation in a GNSS-denied environment,” *Sensors*, vol. 23, no. 6, 2023. DOI: 10.3390/s23063025.

[137] L. S, L. S, Z. J, F. Q, and Y. Y, “Estimator Based on the Adaptive Strong Tracking Kalman Filter for GNSS Vector Receivers,” *Sensors (Basel)*, Jan. 2020.

[138] B. S., Davis, M. Clymer, and G. Graves, “Development of a projectile spin counter and orientation sensor,” in *Defense, Security and Sensing*, 1996.

[139] E. Saltzman, K.-S. Wang, Iliff, and Kenneth, “Aerodynamic assessment of flight-determined subsonic lift and drag characteristics of seven lifting-body and wing-body reentry vehicle configurations,” Dec. 2002.

[140] G. Schmidt, “Navigation sensors and systems in GNSS degraded and denied environments,” *Chinese Journal of Aeronautics*, vol. 23, Dec. 2014. DOI: 10.1016/j.cja.2014.12.001.

[141] G. T. Schmidt, “INS/GPS Technology Trends,” *Meeting Proceedings RTO-EN-SET-116*, pp. 1-1–1-24, Sep. 2011.

[142] G. T. Schmidt and R. E. Phillips, “INS/GPS Integration Architectures,” *Meeting Proceedings RTO-EN-SET-116*, pp. 5-1–5-18, Sep. 2010.

[143] G. T. Schmidt and R. E. Phillips, “INS/GPS Integration Architectures Performance Comparisons,” *Meeting Proceedings RTO-EN-SET-116*, pp. 6-1–6-22, Sep. 2010.

[144] L. Scott, A. Jovancevic, and S. Ganguly, “Rapid signal acquisition techniques for civilian & military user equipments using DSP based FFT processing,” in *Proceedings*

of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2001, 2001.

[145] N. Shen and X. Zhang, "The design for high dynamic GPS receiver in a combinated method of FLL and PLL," in *Emerging Technologies for Information Systems, Computing and Management*, New York, NY: Springer New York, 2013, pp. 3–11.

[146] *SimGEN Software Datasheet with Product Specification*, 2024. [Online]. Available: `https://www.ettus.com/all-products/x310-kit/`.

[147] R. M. Snow and E. A. Bonney, "Aerodynamic characteristics of wings at supersonic speeds," Johns Hopkins University, Applied Physics Laboratory, Tech. Rep. 55, 1947.

[148] F. M. G. Sousa and F. D. Nunes, "Performance analysis of a VDFLL GNSS receiver architecture under ionospheric scintillation and multipath conditions," in *Proc. of the IEEE/ION PLANS*, May 2014, pp. 602–611.

[149] C. Srinu and L. Parayitam, "Performance of GNSS-SDR for IRNSS L5 signals using a low-cost RF front-end," *NAVIGATION: Journal of the Institute of Navigation*, 2023.

[150] M. Susi and D. Borio, "Kalman filtering with noncoherent integrations for Galileo E6-B tracking," *NAVIGATION*, vol. 67, no. 3, pp. 601–618, 2020, doi: 10.1002/navi.380.

[151] T.Takasu, *RTKLIB: An Open Source Program Package for GNSS Positioning*, 2023. [Online]. Available: `https://github.com/tomojitakasu/RTKLIB`.

[152] A. K. Tetewsky and F. Mullen, "Carrier phase wrap-up induced by rotating GPS antennas," 1996.

[153] S. Theodoulis and P. Wernert, "Flight dynamics & control for smart munition: The isl contribution," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 512–15 517, 2017, 20th IFAC World Congress. DOI: `10.1016/j.ifacol.2017.08.2127`.

[154] S. Theodoulis *et al.*, "Guidance and control design for a class of spin-stabilized fin-controlled projectiles," *Journal of Guidance, Control and Dynamics*, vol. 36, no. 2, pp. 517–531, 2013. DOI: `10.2514/1.56520`.

[155] X. Tian, W. Xie, and C. G. Wang, "Attitude estimation for guided projectiles based on magnetometers and accelerometers," *Journal of Physics: Conference Series*, vol. 1069, 2018.

[156] *u-blox M8 concurrent GNSS modules datasheet*, Thalwil, Switzerland, Feb. 2023. [Online]. Available: `https://www.u-blox.com/`.

[157] "U.S. Standard Atmosphere," U.S. Government Printing Office, Tech. Rep., 1976. [Online]. Available: `https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19770009539.pdf`.

[158] J. Vilà-Valls, P. Closas, M. Navarro, and C. Fernández-Prades, "Are PLLs dead? A tutorial on Kalman filter-based techniques for digital carrier synchronization," *IEEE Aerospace & Electronic Systems Magazine*, vol. 32, no. 7, pp. 28–45, 2017, 10.1109/MAES.2017.150260.

[159] R. F. W.P. D'amico L.W Burke, "The hardened subminiature telemetry and sensor systems demonstration phase," Army Research Laboratory, Tech. Rep., 1996.

[160] R. F. W.P. D'amico L.W Burke, "Above Us only stars: Exposing GPS spoofing in Russia and Syria," C4ADS, Tech. Rep., 2019. [Online]. Available: `https://c4ads.org/wp-content/uploads/2022/05/AboveUsOnlyStars-Report.pdf`.

[161] X. Wang, X. Ji, S. Feng, and V. Calmettes, “A high-sensitivity GPS receiver carrier-tracking loop design for high-dynamic applications,” *GPS Solutions*, vol. 19, pp. 225–236, 2015.

[162] Y. Wang, W. Song, D. Fang, and Q.-w. Guo, “Guidance and control design for a class of spin-stabilized projectiles with a two-dimensional trajectory correction fuze,” *International Journal of Aerospace Engineering*, vol. 2015, pp. 1–15, 2015.

[163] P. W. Ward, “Performance comparison between FLL, PLL and a 11l FLL-assisted PLL carrier tracking loop under RF interference conditions,” in *Proc. of the ION GPS*, Sep. 1998.

[164] J.-H. Won, T. Pany, and B. Eissfeller, “Characteristics of Kalman filters for GNSS tracking loops,” *IEEE Trans. on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3671–3681, Oct. 2012.

[165] J.-H. Won, D. Dötterböck, and B. Eissfeller, “Performance comparison of different forms of Kalman filter approaches for vector-based GNSS signal tracking loop,” *Navigation*, vol. 57, pp. 185–199, Sep. 2010. DOI: `10.1002/j.2161-4296.2010.tb01777.x`.

[166] J. Wright and M. Costello, “Use of ground-based signals of opportunity for smart projectile navigation,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, no. 1, pp. 63–76, 2011. DOI: `10.1243/09544100JAERO751`.

[167] B. Xu, Q. Jia, and L.-T. Hsu, “Vector tracking loop-based GNSS NLOS detection and correction: Algorithm design and performance analysis,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 4604–4619, 2020. DOI: `10.1109/TIM.2019.2950578`.

[168] Y. Yi and D. A. Grejner-Brzezinska, “Tighly-coupled GPS/INS integration using unscented Kalman filter and particle filter,” Proc. of the 19st International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS, Fort Worth, TX, Sep. 2006, pp. 2182–2191.

[169] P. Zarchan, H. Musoff, A. I. of Aeronautics, and Astronautics, *Fundamentals of Kalman Filtering: A Practical Approach* (Progress in astronautics and aeronautics). American Institute of Aeronautics and Astronautics, 2009.

[170] X. Zhang, X. Zhan, and J. Shen, “An open-source real-time software vector tracking GNSS receiver - Design, tests and results,” in *Proc. of the ION ITM*, San Diego, CA, Jan. 2013, pp. 351–485.

[171] S. Zhao and D. Akos, “An open source GPS/GNSS vector tracking loop - Implementation, filter tuning and results,” in *Proc. of the ION ITM*, San Diego, CA, Jan. 2011, pp. 1293–1305.