

# A Survey on Low-Power GNSS

Antoine Grenier<sup>1</sup>, Graduate Student Member, IEEE, Elena Simona Lohan<sup>2</sup>, Senior Member, IEEE, Aleksandr Ometov<sup>3</sup>, Senior Member, IEEE, and Jari Nurmi<sup>4</sup>, Senior Member, IEEE

**Abstract**—With the miniaturization of electronics, Global Navigation Satellite Systems (GNSS) receivers are getting more and more embedded into devices with harsh energy constraints. This process has led to new signal processing challenges due to the limited processing power on battery-operated devices and to challenging wireless environments, such as deep urban canyons, tunnels and bridges, forest canopies, increased jamming and spoofing. The latter is typically tackled via new GNSS constellations and modernization of the GNSS signals. However, the increase in signal complexity leads to higher computation requirements to recover the signals; thus, the trade-off between precision and energy should be evaluated for each application. This paper dives into low-power GNSS, focusing on the energy consumption of satellite-based positioning receivers used in battery-operated consumer devices and Internet of Things (IoT) sensors. We briefly overview the GNSS basics and the differences between legacy and modernized signals. Factors dominating the energy consumption of GNSS receivers are then reviewed, with special attention given to the complexity of the processing algorithms. Onboard and offloaded (Cloud/Edge) processing strategies are explored and compared. Finally, we highlight the current challenges of today’s research in low-power GNSS.

**Index Terms**—Global navigation satellite system, global positioning system, low-power electronics, digital signal processing, approximate computing.

## I. INTRODUCTION

**S**ATELLITE positioning applications have changed greatly over the last 30 years. Once mostly used for military, research, or surveying purposes, their democratization has led to the integration in almost every modern embedded device, for both positioning and timing purposes. The evident reason is the miniaturization of electronics, which was pushing towards cost, size, and power consumption reductions, thus, new usages of receivers have emerged, followed by new challenges.

Embedded receivers are bound to be used in more challenging environments, with interference, multipath and/or partial sky-view. Jamming and spoofing, once only applicable to military scenarios, are now reaching the general public [1], [2], with possible substantial consequences on crucial applications, e.g., automated driving [3]. Consequently, additional resources

are needed to be spent to counter interference and still provide a positioning solution in line with the application requirements. Figure 1 illustrates the challenges linked to the new GNSS usages with low-power constraints. In this scenario, the devices are placed in an urban canyon environment, which typically implies multipath propagation. Possible intentional (e.g., jamming) and non-intentional (e.g., Radio Detection and Ranging (RADAR)) interference sources might be found in such environment. Users can be multiple: a pedestrian with a connected wearable requiring positioning capacities or a delivery vehicle with geo-tagged packages and possible automated driving capacities.

All these use cases, while placed in the same signal environment, have different constraints in terms of receiver hardware and targeted precision of location. Both need to be designed conjointly, yet assessing these constraints quantitatively for a perfect optimization is not trivial. Moreover, the questions of multi-frequency reception, multi-constellation capabilities, access to assisted data, or best algorithms for the application case might not have straightforward answers.

Power consumption has been a challenge for electronics from the very beginning of the computer age [4]. Nowadays, embedded devices with finite energy resources push back the requirements on energy consumption even further. The growing interest in the Internet of Things (IoT) is leading to a never-ending increase in their numbers, and it is predicted that by 2024 more than 20 billion of all the connected devices will be related to IoT [5]. They include a variety of sensors onboard but only have access to limited energy resources. In the case of GNSS, it is known to be one of the most power-consuming sensors in embedded devices [6]. Research in energy-constrained platforms is referred to as “low-power GNSS”. The key difference with “traditional” GNSS is the attention given to energy consumption to compute a position. The range of applications for GNSS receivers are various (e.g., positioning, timing, monitoring) and interests might be given on different performance indicators (e.g., integrity, reliability, rapidity, accuracy). Thus, energy efficiency might not be part of the parameter set to optimise. Yet, in energy-constrained platforms, it will impact the battery life and consequently the device operations and should be put at the centre of the developments.

In this paper, we focus on understanding the origins of power consumption in GNSS receivers. We hypothesise that algorithm complexity is a driving factor in the energy budget, especially in the early stages of the processing chain such as the Digital Signal Processing (DSP). Improvements in hardware technology should lead to a decrease in energy consumption, as seen in the last decade of research in this area [7], [8],

Manuscript received 5 September 2022; revised 23 February 2023; accepted 4 April 2023. Date of publication 10 April 2023; date of current version 23 August 2023. This work was supported by the European Union’s Horizon 2020 Research and Innovation Programme under Marie Skłodowska Curie Grant 956090 (APROPOS: Approximate Computing for Power and Energy Optimisation, <http://www.apropos-itn.eu/>). (Corresponding author: Antoine Grenier.)

The authors are with the Electrical Engineering Unit, Tampere University, 33720 Tampere, Finland (e-mail: antoine.grenier@tuni.fi; elena-simona.lohan@tuni.fi; aleksandr.ometov@tuni.fi; jari.nurmi@tuni.fi).

Digital Object Identifier 10.1109/COMST.2023.3265841

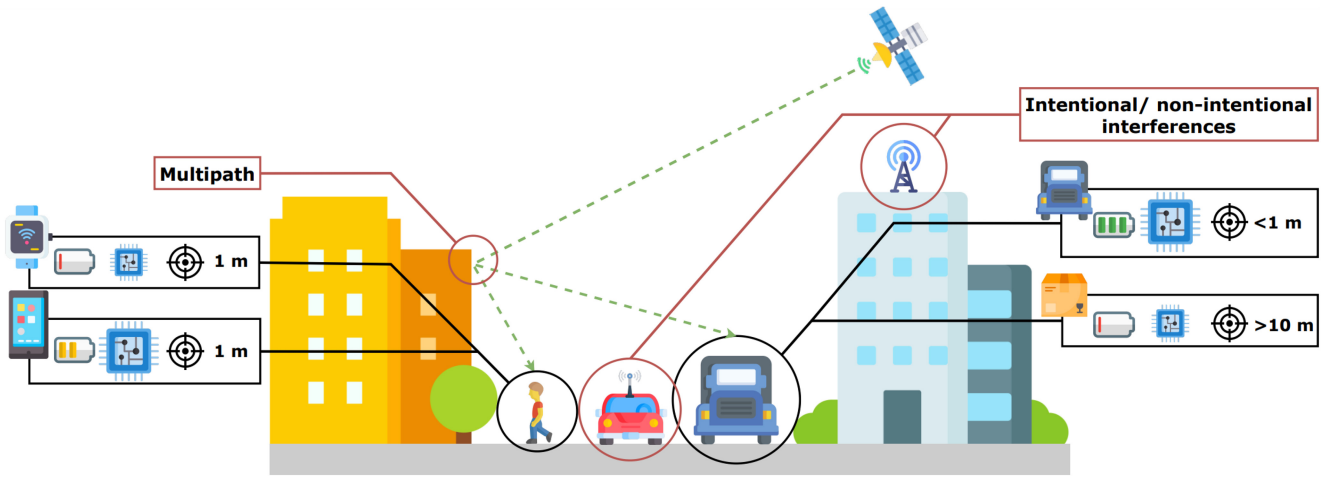


Fig. 1. Examples of low-power GNSS scenarios and their challenges. Necessary accuracies (1 versus 10 m in our examples) are application-dependent.

TABLE I  
SUMMARY OF REVIEWED SURVEYS ON TRACKING TECHNIQUES AND GNSS RECEIVERS ENERGY CONSUMPTION

Ref.	Year	Source type	Title	Contributions
[13]	2015	Thesis	Intermittent GNSS Signal Tracking for Improved Receiver Power Performance.	Analysis of duty cycling strategies in GNSS signal tracking with IMU. Review of factors affecting the energy of a receiver. Duty cycling strategies applied on real scenarios.
[15]	2017	Article	Improving the energy performance of GPS receivers for location tracking applications	Analysis of the energy consumption of the processes in the positioning chain. Experimental analysis based on power measurements on an FPGA implementation.
[11]	2017	Survey	Unambiguous Techniques Modernized GNSS Signals: Surveying the solutions	Review of unambiguous techniques for modern GNSS signal acquisition, with qualitative complexity ranking.
[12]	2020	Article	Unambiguous method for DLL BOC signals tracking	Review of several tracking techniques. Proposed complexity comparison in terms of processing consumed time.
[14]	2021	White Paper	Low-power GNSS for tracking applications	Analysis of power-consumption factors in GNSS. Review on-board and off-loaded architectures to reduce power consumption.
<b>This work</b>	2023	Survey/Tutorial	A Survey on Low-Power GNSS	Survey on low-power GNSS, focusing on the impact of algorithmic complexity of the GNSS processing chain over the energy consumption of receivers.

[9], [10]. While both hardware and software design will drive energy consumption, our review focuses on the software side.

## II. RELATED WORK AND OPEN RESEARCH QUESTIONS

Several surveys on GNSS algorithms have presented summaries for GNSS signal processing techniques. In [11], [12], algorithms for modern signal processing were classified and compared over their performances. Studies defining the causes of power consumption are, however, quite scarce. In [13], the author reviewed several factors that might influence power consumption, e.g., sampling rate, quantization, number of tracked satellites, etc. Yet, the research focuses on duty cycling for tracking, and the study of power consumption factors was mainly limited to this angle.

In a recent white paper [14], factors influencing power consumption by a GNSS receiver were listed, primarily focusing on the user requirements. Various solutions to address the rising questions were proposed, but no details on the data,

processing and results cited were provided. Nevertheless, the paper introduced the concept of offloaded processing, where a part of the processing is performed by a remote server. Such infrastructure is quite novel, as traditionally all processing is performed onboard the receiver. Yet, the solution is claimed to be more energy efficient for specific applications that only require a few positions per day than for applications requiring continuous or frequent positions.

In [15] the power consumption of different GNSS processes was reviewed, by measuring it on a Field-Programmable Gate Array (FPGA) board. Unfortunately, many details on the algorithms and data used for each process were lacking in the paper to understand the results comprehensively. A clear highlight of the study was that most of the energy was consumed by the DSP side of the algorithms (i.e., acquisition and tracking of the signals). A summary of the surveys reviewed is given in Table I.

In this survey, we approach the problem of reducing the energy consumption of a GNSS receiver. The first step is to

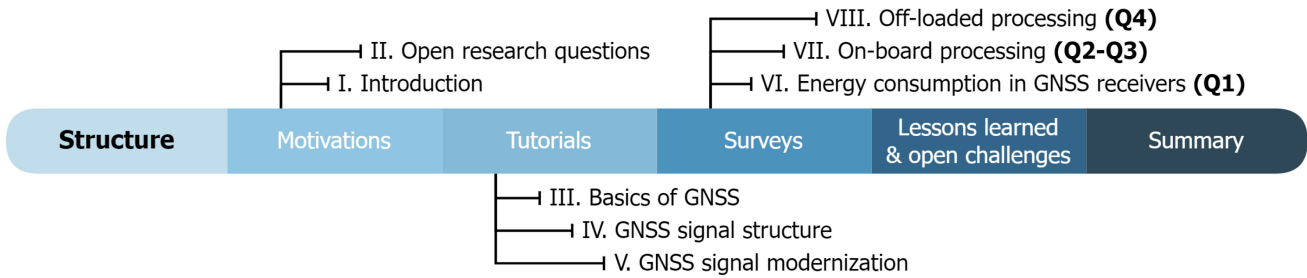


Fig. 2. Overview of the paper's sections.

identify what influence the energy consumption of a receiver and the driving factors (Q1). Since the scope of our review is on the software side, the main objective is to find the processing algorithms requiring the smallest computational complexity, to reduce energy consumption. Based on [15], the DSP algorithms are of prime interest, as they seem to account for most of the energy budget. Thus, a review of the acquisition and tracking algorithms should be performed to highlight the most efficient ones (Q2). Additionally, the impact of the GNSS signal modernization and increased complexity should be also taken into account (Q3). Finally, the growing interest in offloaded architecture makes it quite interesting as another processing approach. This architecture should be reviewed and compared against traditional onboard processing (Q4). Our four research questions can be summarized below:

- Q1** What are the factors driving energy consumption in a GNSS receiver?
- Q2** What are the lowest complexity algorithms for signal acquisition and tracking?
- Q3** What is the cost of tracking modernized GNSS signals?
- Q4** How much more energy-efficient offloaded processing is compared to onboard processing?

To the authors' best knowledge, no comprehensive review of the factors influencing the power and energy consumption in GNSS receivers has been published before. This paper assesses the current state-of-the-art research about low-power GNSS positioning. We review the whole positioning chain, focusing on algorithm complexity in the DSP side. As it is often viewed as the highest complexity phase of the chain, it can be assumed to be responsible for the highest energy consumption [15].

Hardware selection also has a significant effect on power consumption. Yet, the design of processing hardware is more a consequence of other factors, such as algorithmic complexity. Therefore, the rest of this study will not look into optimizations at the hardware level but will focus instead on the DSP to understand the consequences of algorithm complexity on energy consumption. More specifically, we focus on the research made over the last decade on low-complexity methods for modern GNSS signal processing and how they might impact energy consumption compared to legacy signal acquisition. Most of the papers reviewed have been retrieved from searches in several databases (i.e., Scopus, Google Scholar, and Andor) and filtered under the terms "low energy/complexity/consumption".

First, we start with several tutorial sections, to provide an overview of what is GNSS. In Section III, we review the basics of GNSS and positioning. In Section IV, we describe the structure of GNSS signals. In Section V, we introduce the major differences between legacy and modernized signals. Secondly, we seek answers to our four research questions and continue with the survey sections. In Section VI, we provide a review of the factors that influence energy consumption in a GNSS receiver (Q1). In Section VII, we review the processing chain of the onboard processing architecture while listing several algorithms, detailing their complexity and limits (Q2 and Q3). In Section VIII, the processing chain of the offloaded architecture is also reviewed, along with the state-of-the-art status and pending research questions (Q4). Finally, in Section IX, we review the lessons learned throughout the survey sections, in an attempt to find solutions to our research questions (Q1-Q4). We highlight the current limitations in the state-of-the-art research of low-power GNSS and identify the open challenges. Figure 2 proposes an overview of the paper's sections and links to the research questions.

### III. WHAT IS GNSS?

GNSS stands for all the space-borne positioning systems disseminating radio signals to be used globally for positioning or timing purposes. Once dominated by the two legacy systems GPS (USA) and GLONASS (Russia), new constellations have risen over the last two decades. This gave rise to the term "GNSS", which accounts for all the constellations.

#### A. Historical Outlook

While the name "GPS" has been used for quite some time, today it might mislead the general public about what it represents. The Global Positioning System (GPS) was the first satellite positioning system based on the principle of Time of Arrival (TOA). It is constructed and maintained by the USA, its first satellite was launched in 1978, replacing an older satellite positioning system called "TRANSIT". Shortly after, GLONASS was created by the former USSR and today handled by Russia, its first satellite was launched in 1984. For a while, these two constellations rule the satellite positioning world.

Because of the military and economical advantages that GNSS represent today, several countries have decided to develop their positioning systems. Four global constellations are present over the world: GPS, GLONASS, Galileo from the European Union (EU) and BeiDou from China. These last two systems

have been under development for the past two decades and only recently achieved Full Operational Capacity (FOC).

While having all these systems might be redundant for the general public user, it provides many advantages. Besides the enhanced precision and functionalities offered by the modernized signals on the new constellations (see Section V), the increasing number of in-orbit satellites already provides enhanced accuracy. Most GNSS chips available to public devices (e.g., smartphones) not only receive GPS signals but signals from all the GNSS constellations, for increased visibility, measurement redundancy and positioning reliability.

### B. Basics of Navigation

While they differ in their implementations, all the GNSS systems are built around the same positioning principle: trilateration. Let us define a *user* with unknown coordinates and *references* with known coordinates. By providing several range measurements (i.e., at least three for a 3D position) between the user and the references it is possible to compute the position of the user, by using the principle of trilateration. It has been used for centuries by land surveyors and navigators to find their positions. While there exist other positioning principles (e.g., triangulation), trilateration is of interest here because it is the principle used for satellite positioning. Reference points are replaced by satellites with known orbits and coordinates, and range measurements are retrieved using radio signals.

Conceptually, a GNSS spacecraft can be simplified to a very precise clock orbiting the Earth and regularly sending its current time within a radio signal. A receiver on Earth will receive this signal with a delay relative to the speed of light and the time it takes to travel the several thousand kilometres separating the satellite and the receiver. By decoding the time of transmission, comparing it to the time of reception and multiplying their difference by the speed of light, the receiver can reconstruct the range between the satellite and the receiver. This technique is called Time of Arrival (TOA).

Several ranges from different satellites need to be constructed by the receiver to compute a position using the principle of trilateration [16]. In satellite positioning, these range measurements are called *pseudoranges*. Contrary to the previous example of trilateration, which required only three ranging measurements, the satellite case requires a minimum of four, to account for the ambiguity in time synchronicity between the satellites and receiver clocks. The interested reader can find more details of this principle in [16].

## IV. EXPLORING THE GNSS SIGNAL STRUCTURE

For a typical user, the sole purpose of GNSS satellites is to transmit their current clock and position state within a radio signal. Similar to a typical communication system, the GNSS signal is composed of a signal carrier, and a message modulated onto this carrier using a modulation process. In this section, we introduce the structure of the GNSS signals and their more recent modernization concepts. We will limit the definitions only to the basic notions of GNSS signals and the necessary terms needed to understand this paper. The interested reader is invited to review these references for a more comprehensive review [16], [17], [18].

Almost all systems today use Code Division Multiple Access (CDMA) to share the frequency spectrum with all the satellites. The only exception is GLONASS, which uses Frequency Division Multiple Access (FDMA) to divide the spectrum amongst its fleet. As CDMA is used by more constellations and is easier to accommodate with the multiple constellations compared to FDMA, we will focus on the former, explaining its concept and focus on its impact on the signal structure. Note however that the Russian space agency ROSCOSMOS has announced new modernized signals with CDMA channels on its new class of satellite K1/K2/KM [19], and the launch of such satellites and signals was announced for the 2020s. Yet, no ICD describing these new signals has been made publicly available, thus excluding them from this review.

To help with the review of the signal structure, we provide a summary in Table II that will be used throughout the sections of this paper. The table provides details on the properties of all the open signals from GPS, Galileo and BeiDou, which are of interest for this survey. It is based on the tables published by [20], [21] and the ICD of the constellations [22], [23], [24]. It summarizes the encoding scheme used, to enable an understanding of the differences between systems. The table only includes signals accessible by non-governmental agencies, usually referred to as Open Service (OS) and Commercial Service (CS). All constellations also include military signals under various names (e.g., M/P codes for GPS, Public Regulated Service (PRS) for Galileo), but they fall outside the scope of this review.

In this paper, we sometimes refer to signals as *legacy* or *modern*. However, their exact differentiation might be unclear in GNSS literature. We define legacy signals as the original GPS signals (i.e., L1 C/A, L1P and L2P), with a “simple” signal structure. We define modern signals as the ones transmitted by the newer generation of GNSS satellites, like Galileo, BeiDou or GPS 2nd/3rd generation, as they have a more complex signal structure with specific functionalities (see Section V). In this section, we will use the example of the legacy GPS L1 C/A signal, the most widely used signal in GNSS, for illustrating purposes.

We start with the case of the legacy signal. A typical legacy GNSS signal is often represented as composed of three components [16], [18]:

- a **signal carrier**, used to propagate the message over the dedicated frequency;
- a **ranging code**, used to identify the satellite and to compute the range measurement. It can also be called *spreading*, *primary* or Pseudo Random Noise (PRN) code;
- a **navigation message**, containing the necessary information to compute a position (transmission time, orbits, etc.)

The code and navigation message are imprinted onto the carrier using a specific modulation process, described in the following sections.

### A. Signal Carrier

GNSS satellites transmit their radio signals over several dedicated frequency bands. Historically, satellite positioning systems have used the L-band (i.e., 1-2 GHz) frequencies



TABLE II  
SUMMARY TABLES OF GNSS SIGNAL CHARACTERISTICS (OPEN/COMMERCIAL SERVICES ONLY). GPS RECEIVED POWER VALUES ARE BASED ON BLOCK III SATELLITES. DNF STANDS FOR DATA NOT FOUND. DETAILS ABOUT THE NAVIGATION MESSAGE DATA TYPES CAN BE FOUND DIRECTLY IN THEIR RESPECTIVE ICD [22], [23], [24]

System		GPS							
Signal		C/A	L1C		L2C		L5		
Frequency [MHz]		1575.42	1575.42		1227.60		1176.45		
In activity		Operational	Experimental		Operational		Operational		
Channel combination		-	Quadrature		Code multiplexing		Quadrature		
Channel (In-phase / Quadrature)		Data	Data (I)	Pilot (Q)	M (Data)	L (Pilot)	Data (I)	Pilot (Q)	
Sub-carrier frequency [MHz]		-	1.023	1.023	-	-	-	-	
Min. bandwidth [MHz]		2.046	4.092	4.092	1.023	1.023	20.46	20.46	
Modulation		BPSK	BOCs	TMBOCs	BPSK	BPSK	BPSK	BPSK	
		(1)	(1,1)	(6, 1, 1/11)	(1)	(1)	(10)	(10)	
Primary code	Rate [Mchip/s]	1.023	1.023	1.023	511.5	511.5	10.23	10.23	
	Length [chip]	1023	10230	10230	10230	767250	10230	10230	
	Length [ms]	1	10	10	20	15000	1	1	
Secondary code	Rate [bits/s]	-	-	100	-	-	1000	1000	
	Length [bits]	-	-	1800	-	-	10	20	
	Length [ms]	-	-	18000	-	-	10	20	
Navigation message	Rate [bit/s]	50	50	-	25	-	50	-	
	Rate [symbol/s]	50	100	-	50	-	100	-	
	Type	LNAV	CNAV-2	-	CNAV	-	CNAV	-	
	Frame size [bits/s]	300 / 6s	900 / 18s	-	300 / 12s	-	300 / 12s	-	
Min. Received Power [dBW]		-158.5	-157		-158.5		-157		
System		GALILEO							
Signal		E1 OS	E6 CS		E5a		E5b	E5	
Frequency [MHz]		1575.42	1278.75		1176.45		1207.14	1191.795	
In activity		Operational	Experimental		Operational		Operational	Operational	
Channel combination		Code multiplexing		Code multiplexing		Quadrature		-	
Channel (In-phase / Quadrature)		B (Data)	C (Pilot)	B (Data)	C (Pilot)	Data (I)	Pilot (Q)	Data (I)	
Sub-carrier frequency [MHz]		1.023	1.023	-	-	-	-	-	
Min. bandwidth [MHz]		6.138	6.138	-	-	-	-	15.345	
Modulation		4.092	4.092	10.23	10.23	20.46	20.46	20.46	
		CBOCs(+)	CBOCs(-)	BPSK	BPSK	BPSK	BPSK	BPSK	
		(6, 1, 1/11)	(6, 1, 1/11)	(5)	(5)	(10)	(10)	(10)	
Primary code	Rate [Mchip/s]	1.023	1.023	5.115	5.115	10.23	10.23	10.23	
	Length [chip]	4092	4092	5115	5115	10230	10230	10230	
	Length [ms]	4	4	1	1	1	1	1	
Secondary code	Rate [bits/s]	-	250	-	1000	1000	1000	1000	
	Length [bits]	-	25	-	100	20	100	4	
	Length [ms]	-	100	-	100	20	100	4	
Navigation message	Rate [bit/s]	125	-	500	-	25	-	125	
	Rate [symbol/s]	250	-	1000	-	50	-	250	
	Type	I/NAV	-	C/NAV	-	F/NAV	-	I/NAV	
	Frame size [bits/s]	250 / 1s	-	-	-	500 / 50 s	-	250 / 1s	
Min. Received Power [dBW]		-157	-155		-155		-155	-155	
System		BEIDOU							
Signal		B1I	B1C		B3I	B2		B2a	B2b
Frequency [MHz]		1561.098	1575.42		1268.52	1207.14		1176.45	1207.14
In activity		Operational	Operational		Operational	Operational		Operational	Operational
Channel combination		Quadrature		Code Multiplexing		-	Quadrature		-
Channel (In-phase / Quadrature)		I (Data)	Q (Pilot)	Data	Pilot	Data	I (Data)	Q (Pilot)	
Sub-carrier frequency [MHz]		-	-	1.023	1.023	-	-	-	
Min. bandwidth [MHz]		-	-	1.023	6.138	-	-	-	
Modulation		4.092	4.092	4.092	4.092	20.46	4.092	4.092	
		BPSK	BPSK	BOCs	QMBOCs	BPSK	BPSK	BPSK	
		(2)	(2)	(1, 1)	(6, 1, 4/33)	(10)	(2)	(10)	
Primary code	Rate [Mchip/s]	2.046	2.046	1.023	1.023	10.23	2.046	2.046	
	Length [chip]	2046	2046	10230	10230	10230	2046	2046	
	Length [ms]	1	1	10	10	10	1	1	
Secondary code	Rate [bits/s]	1000	1000	-	100	1000	1000	1000	
	Length [bits]	20	20	-	1800	20	20	5	
	Length [ms]	20	20	-	18000	20	20	5	
Navigation message	Rate [bit/s]	50/500	-	50	-	50/500	50/500	-	
	Rate [symbol/s]	DNF	-	100	-	DNF	DNF	-	
	Type	D1 / D2	-	B-CNAV1	-	D1 / D2	-	B-CNAV2	
	Frame size [bits/s]	NAV	-	1800 / 18 s	-	NAV	-	NAV	
		300 / 6s or 0.6s	-	1800 / 18 s	-	300/6s or 0.6s	-	288 / 3s	
Min. Received Power [dBW]		-163	-159 / -161		-163		-163	-156/158	
								-160/162	

for carrier signals, thanks to their great propagation properties through the atmosphere [25]. The first ones transmitted were L1 (1575.42 MHz) and L2 (1227.60 MHz). With the new

generation of satellites, additional frequencies have emerged: L5 (1176.0 MHz) and L6 (1278.75). The different GNSS signals are distributed across these bands, depending on their

usage and properties. As L1 is the legacy frequency, many signals co-exist on this band to be received by mono-frequency antennas. The L5 frequency has been part of the recent modernization of the GNSS system and is especially adapted for aviation purposes [26]. Multi-frequency reception allows many positioning advantages, the most crucial being ionospheric mitigation. However, it is outside the scope of this paper and the interested reader can review the book [16].

### B. Code

Codes are also often referred to as *ranging* codes in literature. Indeed, their primary purpose is the computation of the range measurement from satellite to receiver. But as they also serve additional purposes within the system, they can also be named Pseudo Random Noise (PRN) codes, *spreading* code or *primary* code, depending on the context. All these names refer to the same code in the GNSS signal structure.

All constellations (except GLONASS) use a spectrum-sharing technique called CDMA. This allows all the satellites to transmit on the exactly same frequency without their signals interfering with each other. Its main benefit is the simplified antenna design, which can be optimized for one central frequency.

Each satellite is given a specific code, often referred to as the PRN code. This code can be seen as the “fingerprint” of the satellite within the signal. These “apparently random” codes are used for their correlation properties and their Auto-Correlation Function (ACF). The auto-correlation of a random code will result in a single high peak when the two codes are aligned, a well-known triangular ACF. Originally, GPS PRN codes are based on Gold code principles [18], known for their small cross-correlation properties similar to a randomly generated code. Yet the generation of such codes is systematic and allows replicas to be easily re-generated within the receiver without the need to save them in memory. In their final form, PRN codes are binary strings of various lengths, depending on the signal definition. While these strings are actually “bits”, a bit from the code is usually referred to as a “chip”, to avoid confusion with the bits from the navigation message. In Table II, the code length for each signal is reported in their “Primary code” row. The distinction between *primary* and *secondary* codes will be detailed in Section V-B, as it related to modern GNSS signals.

The code can also be referred to as the *spreading* code, being used to spread the energy of the signal over a larger part of the frequency spectrum. This is done by alternating the code at a high frequency (i.e., MHz level). A wider bandwidth helps with interference mitigation, making the system less prone to intentional or non-intentional jamming. For example, the GPS L1 C/A uses a code frequency of 1.023 MChips, i.e., 1023 code bits per millisecond. This frequency is thus often taken as a reference for the definition of other signals.

### C. Navigation Message

The navigation message, also known as the *data* component of the signal, contains all the necessary information to

compute a position. The main information are: the transmission time, also known as the Time of Week (TOW); the orbits of the satellite (i.e., satellite ephemeris); corrections on the satellite’s clock; atmospheric corrections, etc. The protocol for the message structure is different for each constellation and is described in detail in their respective ICD [22], [23], [24].

The data component is transmitted at a low rate, due to the long transmission distance that might induce errors during its reception. Legacy signals such as the GPS L1 C/A use a rate of 50 bits per second, while modern signals with higher code rates and enhanced error checks have an increased transmission rate (see Section V).

### D. Modulation

Once the code and the data components are prepared, they are assembled and modulated on top of the carrier. In GNSS signals, the most common modulation used is the Binary Phase Shift Keying (BPSK) modulation. BPSK is the simplest form of Phase Shift Keying (PSK) modulations and the original modulation used for all legacy signals. Data bits are symbolized in the signal by a 180° phase inversion in the signal carrier [18], as seen in Figure 3. When the BPSK modulation is used in combination with the spreading (i.e., PRN) code, the resulting signal is called a Direct Sequence Spread Spectrum (DSSS) [18]. Among its many properties, such a signal has most of its energy in a single main lobe centred on its carrier frequency. An example of the Power Spectral Density (PSD) of a BPSK modulated signal can be found in Figure 4.

### E. Transmission and Signal Recovery

Once the code and the data have been modulated on the carrier, the resulting signal is transmitted by the satellite. GNSS signals arriving on Earth have a very faint power, as they were transmitted over more than 20 000 km from satellite to the surface of Earth. In fact, the received signal is way below the ambient noise floor, approaching -160 dBW. The signal recovery is possible thanks to the PRN code, as it is a recognizable structure within the noise.

Depending on the signal quality, receiver architecture, and current state, a typical receiver might need several seconds up to several tens of seconds to provide a first position fix. The time taken for the first fix is called the Time To First Fix (TTFF). It is an important comparison metric between receivers (see Section VI).

We do not review the processes needed for successful recovery of the GNSS signal by a receiver, as this will be further detailed in Section VII.

## V. MODERNIZATION OF THE GNSS SIGNALS

Modernization of GPS signals has been ongoing since the beginning of the satellite positioning era. Today, GPS has reached its 3<sup>rd</sup> generation (block III); the first satellite of it launched in December 2018, with currently six in orbit. BeiDou and Galileo have been under preparation for more than two decades. BeiDou reached its fleet completion in June 2020 [27], with currently 44 operational satellites. Galileo reached Full Operational Capacity in December 2021 [28] and

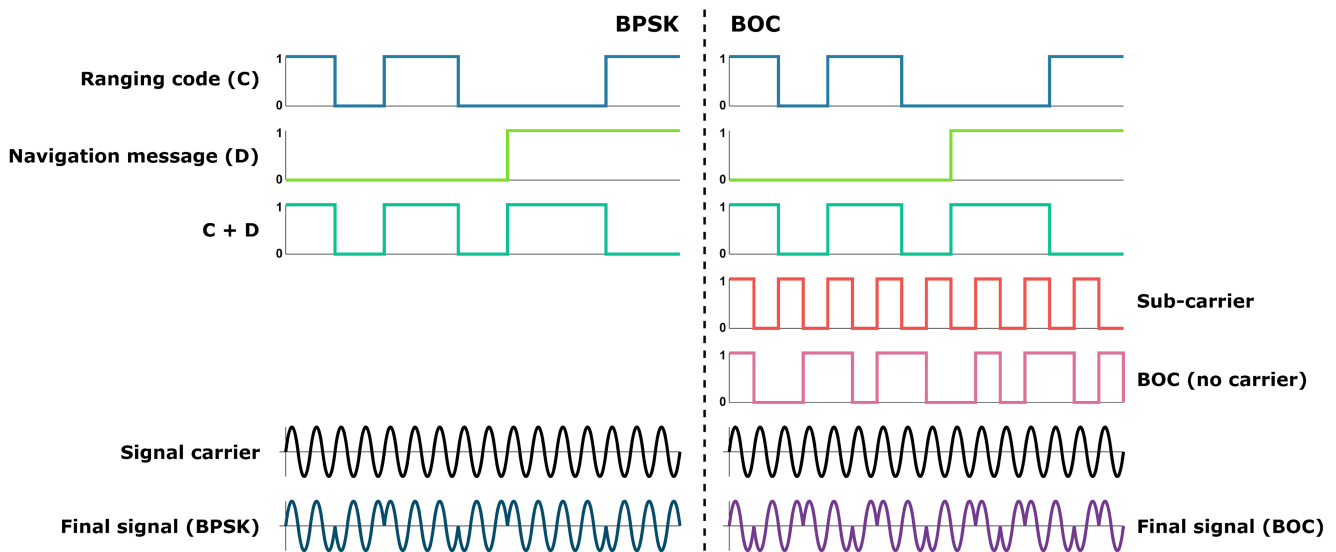


Fig. 3. Structure of legacy BPSK and modern BOC signals.

currently has 23 operational satellites. Every GNSS constellation transmits *modern* GNSS signals, which use more complex modulation schemes compared to the legacy GPS signals. They come with several benefits: better inter-system cross-correlation and increased resistance to signal interference, but also their own set of challenges. A higher sampling rate and more complex algorithms are required to receive these signals, leading to an increase in computation complexity [21]. Signal modernization and its consequences on acquisition/tracking complexity are among the most researched topics over the last decades (see Section VII).

The number of open signals available has increased significantly over the last decade and can be found in Table II. Some signals (i.e., GPS L1C and Galileo E6) are still under experimental usage only, thus flagged as unhealthy and carrying a dummy message. Their operation capacity is planned for the coming years. This signal diversity implies a revision of the standard low-power GPS receiver with L1 C/A capacity only, as these signals have been designed to answer the evolved needs of the users.

In this section, we review the features of the modern signals compared to the legacy structure explained in Section IV. We highlight the improvement in the signal properties while pointing out the increase in the computational complexity of the tracking algorithms. The detailed properties of the modern signals can be found in Table II.

#### A. New Modulation

Early in 2000s, the number of planned GNSS constellations rose, as well as the need for improved GNSS signals. As an answer, the Binary Offset Carrier (BOC) modulation was created by [29], specifically designed for GNSS systems interoperability. This modulation applies a *sub-carrier* in the form of a sine/cosine wave on top of the usual carrier. The frequency of this wave is a multiple of the reference frequency 1.023 MHz. This is represented in the Figure 3. The effect on the spectrum is the separation of the main lobe into side

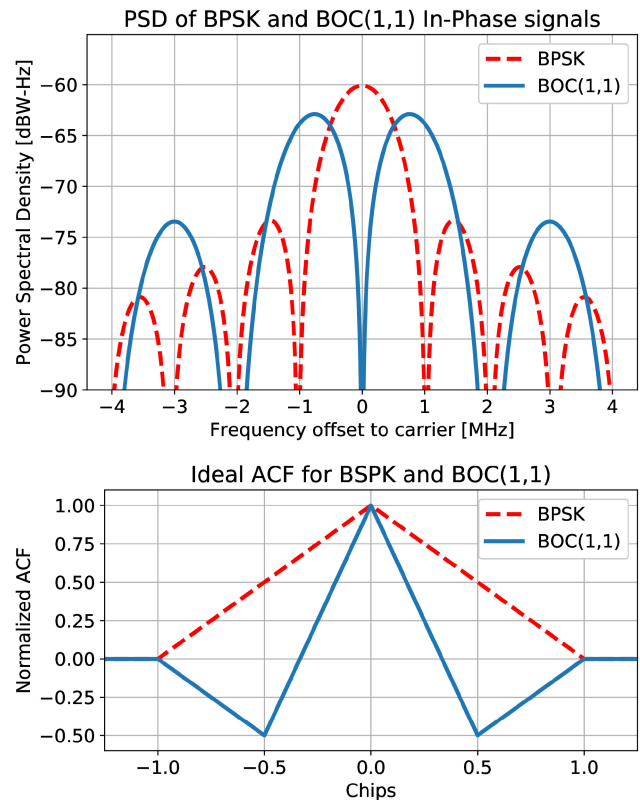


Fig. 4. Comparison of BPSK and BOC(1,1) signals.

lobes located around the carrier frequency. Consequently, the Auto-Correlation Function (ACF) of a BOC modulated signal is different from BPSK, creating a sharper central peak and additional side peaks, smaller in magnitude.

The code frequency can be multiplied to increase the spreading further. The notation  $\text{BOC}(n, m)$  provides information on these multiples:  $n$  represents the sub-carrier frequency multiple and  $m$  the code frequency multiple. For example, the  $\text{BOC}(1,1)$  modulation applies a sub-carrier of  $1 \cdot 1.023$  MHz, creating

two side lobes centred on  $\pm 1.023$  MHz, while the spreading of each lobe is left unchanged (i.e., 2.046 MHz). Figure 4 represent the effect of BPSK and BOC modulations on the signal, with the example of BOC(1,1). The BOC(1,1) modulation is used for L1C and the conception of the B1C signal. More “exotic” combinations are possible, like the BOC(6,1,1/11) modulation used for the E1 signals, where two signals in BOC(1,1) and BOC(6,1) are generated and mixed. Details of each signal modulation are given in Table II. The interested reader can read these references for a more comprehensive review of the BOC modulation [18], [30].

There are two main advantages with BOC modulated signals: (1) the signal is spread over a larger spectrum, therefore rendering it less prone to interference; (2) a sharper ACF, leading to better performance in tracking (i.e., more precise measurements) and resilience to multipath [30].

### B. Additional Channels and Codes

Many modernized signals are divided into two channels: one containing the navigation data (i.e., *data* channel) and one data-less, containing only the PRN code (i.e., *pilot* channel). These channels could be either multiplexed (e.g., L2C, E1, B1C) or in quadrature, with the data in-phase and the pilot in quadrature-phase, rotated  $90^\circ$  (e.g., L1C, L5, E5a/b, B1I, etc.). Dataless channels are interesting for acquisition and tracking procedures, as they allow longer integration and easier tracking without the risk of a data bit change.

Pilot channels also contain a *secondary* code, also known as *tiered* code, on top of the usual PRN code. These codes are designed for better cross-correlation properties and to help with the data bit synchronization [30]. Consequently, in modernized signals, the PRN codes are renamed *primary* codes.

All the modernized signals target longer primary code. Longer codes allow again better cross-correlation properties and resistance to multipath, which are both of interest for challenging environments [30]. Moreover, some signals transmit their navigation messages with an increased bit rate, thanks to advanced techniques such as Cyclic Redundancy Check (CRC) and Forward Error Correction (FEC) [30].

### C. New Services

With the evolving needs of satellite positioning, new generations propose additional services in response. Given our scope on low power, two services are of particular interest: (1) anti-spoofing protection; (2) improved accuracy.

1) *Anti-Spoofing Protection: Jamming and spoofing* of GNSS signals are other challenges related to the wireless environment. Although they do not directly increase energy consumption, they will delay or even prevent positioning. Both of these techniques refer to the act of a nearby transmitter. A jammer sends a high-power waveform on the appropriate frequency, while a spoofer sends a signal with similar characteristics as the satellite signals but with false information inside [31]. While such techniques were first reserved for military applications, their use by the general public for individual applications has been growing. Today a simple jammer can be bought for 15\$ on the Internet [31]. The conception of

a jammer is straightforward and has existed since the start of the GPS era. A spoofer requires more sophistication to fool a receiver. Yet, recent studies [2], [32] have shown that Do-It-Yourself projects can easily be implemented at a low cost ( $< 500\$$ ) to create a basic spoofing device. With the ever-growing usage of satellite positioning in sensitive applications and easy access to spoofing hardware, the fight against these attacks is getting closer to the civilian world and needs to be considered for the new generations of receivers.

While jamming is easy to spot in signals, it is harder to detect a spoofed signal as it is constructed to mimic a real satellite signal and fool the receiver. Both GPS and Galileo have included anti-spoofing techniques within their modernized signals. For GPS, this service is called Chip-Message Robust Authentication (CHIMERA) and consists of both Navigation Message Authentication (NMA) and Spreading Code Authentication (SCA) techniques. For Galileo, these two authentications are separated into two services: respectively, the Open Service Navigation Message Authentication (OSNMA) located in the I/NAV message from E1 (E1-B), and the Commercial Authentication Service (CAS) located on the E6 pilot component (E6-C). Reference [33] have shown the benefits of using both CHIMERA and OSNMA conjointly for spoofing protection.

Given the vast number of critical applications relying on the GNSS signals today, the new services are aimed to improve positioning safety and reliability.

2) *Improved Accuracy Services:* Galileo will also offer the High Accuracy Service (HAS) located in the C/NAV message from E6 (E6-B). This service aims to enable real-time Precise Point Positioning (PPP) by providing additional corrections to the user (e.g., precise satellite ephemeris and clock modelling). Using this information, the receiver can compute a position with an accuracy of 20 cm (95%) horizontal / 40 cm (95%) vertical in less than 300 s. The Initial Service phase should start to be declared through 2022, with a Full Service phase by 2024 [34]. Yet to achieve this accuracy, almost continuous decoding of the navigation message will be required, as corrections are valid only for a short time (e.g., 10s for satellite clock corrections). Its applicability for low-power systems and challenging environments will need to be further reviewed. Note that a similar service should be provided by the BeiDou constellation over the B3 signal, yet no details of the service could be found.

### D. Increase in Signal Complexity

All the new functionalities cited in the last sections come with one major cost: a larger frequency bandwidth that requires an increased sampling rate to be recovered. Compared with the GPS L1 C/A, the GPS L1C requires four times more bandwidth, whereas the GPS L5 requires 10 times more. It is an important drawback to consider for the design of low-power receivers, as sampling frequency substantially impacts the operations and memory requirements. Additionally, features like secondary codes and longer primary codes also lead to an increase in algorithm complexity, further reviewed in Section VII.



The BOC modulation also leads to an increase in the algorithm complexity. The side peaks, located around the narrow central peak, create an ambiguity during the signal processing phases. Algorithms originally designed for legacy signals have difficulties differentiating the main peak from the side peaks. Locking onto one of these side peaks will lead to a significant bias in the measurements and the receiver position. While the sharpness of the central peak enables a great increase in measurement precision, modern algorithms must overcome this ambiguity. The solutions to this issue and the consequent increase in complexity will be detailed in Section VII.

Nevertheless, we have seen that modern signals offer numerous advantages over the legacy L1 C/A signal. While the complexity of decoding modernized signals is set to be greater than GPS L1 C/A, the impact on the energy budget should be reviewed and compared against the improvement in terms of precision/accuracy, positioning reliability, etc.

The basics of GNSS have been presented in this section, along with the definition of modern GNSS signals. They offer higher performances in response to the new challenges of their modern usage and additional functionalities compared to legacy GPS signals. However, this comes with an extra price in computational complexity to track them, which is expected to lead to higher energy requirements.

## VI. ENERGY CONSUMPTION IN GNSS RECEIVERS

As GNSS receivers get more and more integrated into embedded devices, “low-power GNSS” has become a growing field of interest over the last decade. Understanding what dictates the energy consumption of such a sensor is important for devices with limited energy resources (e.g., IoT).

### A. Power, Energy and Complexity

Before diving into the energy consumption of a GNSS receiver, it seemed necessary to recall the difference between power and energy consumption, as they are key concepts in this analysis and are often misused in literature by non-electronic experts. Additionally, we take the opportunity here to explain our assumptions on energy consumption and algorithmic complexity, as our review will focus on the software side of GNSS receivers.

1) *Power Versus Energy Consumption:* The implementation of GNSS receivers for embedded devices is usually marked as “low-power”, yet these words fail to represent the ideas behind such an implementation. Electrical power is a rate unit and defines the amount of electrical energy consumed over time. It is usually expressed in Watt, corresponding to one Joule per second. Energy consumption, defined in Joule, is therefore the power consumption integrated over a certain time.

As the limiting factor in an embedded device today is the finite amount of energy available onboard, the difference is fundamental. Comparing the energy consumption of two processes requires that both the power consumption and the processing time are defined. Consider the example of a high-power consuming process and a low-power process. Estimating which process will consume more energy also

requires knowing how long it takes to perform the processing. Thus, a high-power-consuming process might consume less energy than a low-power process, depending on the time the process has to run. This will be of interest during the presentation of the acquisition and tracking processes in Section VII.

Given that we aim to reduce the amount of energy consumed, as it is directly related to the battery draining on mobile battery-operated devices, both the power and time factors should be reviewed to minimize the final energy consumption.

2) *Relating Energy-Consumption and Algorithm Complexity:* According to the trend of the last decades, hardware technology is meant to continue evolving, leading to an overall energy reduction in newer implementations. However, the complexity of an algorithm is not bound to hardware technology and can be viewed as a metric common to any processing, as well as another angle to continue improving processing efficiency. Studies in computer science focusing on software benchmarking use complexity analysis to classify an algorithm’s efficiency [35], [36]. Similarly to our review of power and energy, would an algorithm running twice as fast (in time complexity domain) than another algorithm have a lower energy consumption when compared over the same hardware technology?

The energy consumption is not only influenced by how long an algorithm will run, but also by the type and number of arithmetic operations, the amount of memory needed, etc. Thus providing additional complexity metrics is important to complement the algorithm analysis on a theoretical level. The most used ones are time and space complexity, respectively the number of elementary operations or “steps” and memory used by an algorithm [37]. Other complexity metrics can sometimes be found in papers, such as arithmetic complexity (number of basic operations such as addition/multiplication/etc.) and bit complexity (number of bit operations needed).

### B. Low-Power GNSS

Today GNSS receivers are derived into a large variety of chipsets. Those designed for low-power devices are referred to as “low-power GNSS”. They come with fewer functionalities and reduced precision compared to the high-precision receiver, in exchange for significant power consumption reduction. Comparing power consumption from different receivers, it is clear they are not meant for the same range of applications. Several examples of GNSS receiver chipsets are provided in Table III, along with their application, capacities and power consumption. Large differences in power consumption can be seen depending on the receiver, showing that the power consumption of GNSS receivers is essentially related to the user requirements. They will be dictating the hardware, the number of tracking channels, receiver frequencies, and constellations embedded in the receiver, which will drive the device’s power consumption.

To better review these numbers, we derived the theoretical energy consumption depending on the receiver state. *Cold start* is defined when the receiver has no information of his current state. This is usually the case when the receiver has

TABLE III  
 EXAMPLES OF GNSS RECEIVER CHIPSETS AND THEIR NOMINAL POWER CONSUMPTION (G: GPS, R: GLONASS, E: GALILEO, C:BEIDOU)

Company	Novatel	U-blox	U-blox	Broadcom	Unicorecomm
<b>Product</b>	PIM222A	ZED-F9H	ZOE-M8B	BCM4778	UC6226
<b>Application</b>	High-precision	High-precision	Low-power	Low-power	Low-power
<b>Reference</b>	[38]	[39]	[40]	[41]	[42]
<b>Constellation</b>	G / E / C	G / R / E / C	G / R / E / C	Not provided	G / R / E / C
<b>Frequencies</b>	L1 / L2 / L5	L1 / L2 / L5	L1	L1 / L5	L1
<b>Number of channels</b>	Not provided	184	72	Not provided	64
<b>TTFF Cold / Hot start [sec]</b>	40s / 10s	24s / 2s	26s / 2s	DNF	29s / 1s
<b>Power consumption [mW]</b>	1200	200	72 / 12 (super-E mode)	4 (single) / 6 (dual)	80

been offline for a prolonged duration (e.g., several months). The receiver needs to recover the navigation message within the signals, increasing the TTFF. *Hot start* is defined when the receiver has already precise knowledge of its location and the satellite ephemeris. Such starts happen when the receiver is initiated with the navigation data recovered through an external channel (e.g., radio or LTE) or just momentarily switched off (e.g., a few seconds). In this case, the receiver does not need to recover the navigation message, reducing the TTFF. As discussed in Section VI-A1, shorter TTFF should mean reduced energy consumption.

The results are represented in Figure 5, based on the parameters from Table III. As no TTFF numbers could be retrieved for the BCM4778 chip, we used 25s (cold start) and 2s (hot start), based on median values from the other receivers. As expected, receivers defined as low-power tend to outperform high-precision receivers in terms of energy consumption by several orders of magnitude. Note that those numbers are coarse approximations of the energy consumption, only based on theoretical values provided by the manufacturers. The actual energy numbers will vary as influenced by several factors (see Section VI-C).

The diversity of applications also leads to new ways of defining a GNSS receiver. Both public research and industry have started looking into moving the processing away from the GNSS receiver and performing the expensive computations on a remote server (e.g., Cloud computing). However, while being attractive and in line with the current processing strategies in embedded electronics, this solution leads to several challenges and might not be suitable for all applications. Onboard computing and offloaded computing are reviewed respectively in Sections VII and VIII.

C. A Multi-Factor Energy-Consumption Problem

Energy consumption in a GNSS receiver is a problem driven by many factors. It has been present from the beginning of the satellite positioning era but is often viewed as an unavoidable consequence of a GNSS design. While the need for reduced energy consumption in embedded devices is rising, few studies have looked into the factors that lead to high energy consumption.

In [13] and [14], a review was performed and used to draft a first picture of the energy consumption in a GNSS

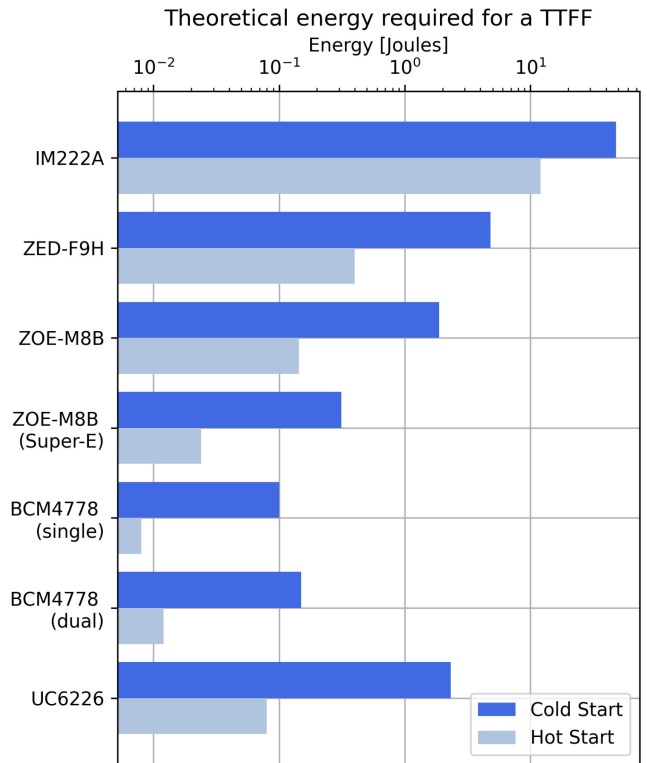


Fig. 5. Theoretical energy required for a TTFF for a cold start (deep blue) and hot start (light blue).

receiver. Three major factors were extracted: hardware complexity, algorithm complexity, and activity time. Each of these components is related to each other, and they are influenced by four major causes: the hardware technology, the user's environment, the user's requirements, and the external aiding. Depending on the application, these causes will change, and different receiver architectures should be considered. Figure 6 provides a diagram to represent this segmentation.

1) *User Requirements*: User requirements are the most evident reason for an increase in power consumption. The positioning precision/accuracy is often the driving factor, but the TTFF, the position update frequency, or even the reliability of the positioning might also be of interest. Those requirements are application-specific and need to be tailored for each case. They need to be transposed into a lower level

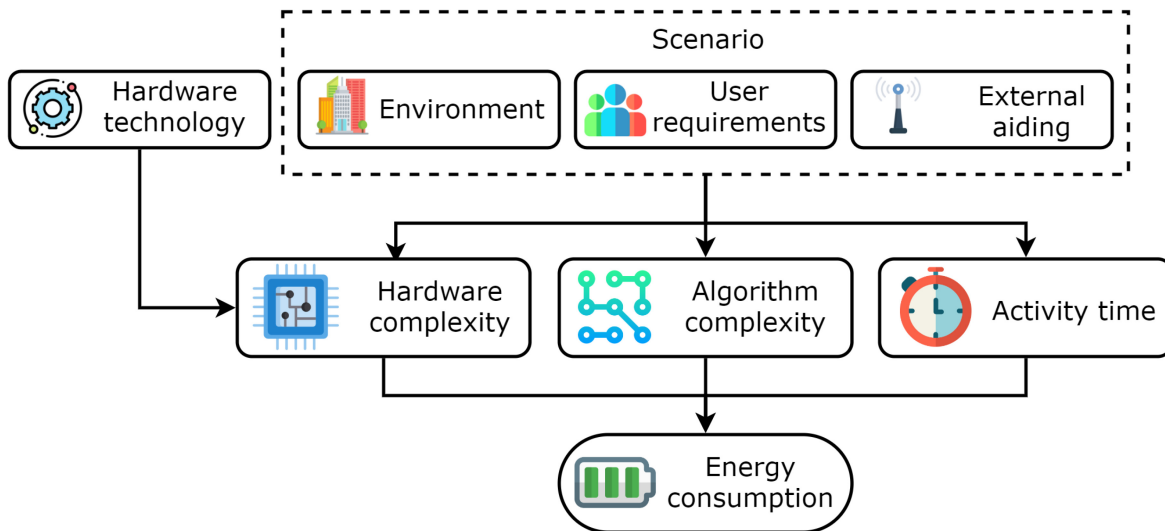


Fig. 6. Factors driving the energy consumption of a GNSS receiver.

for a receiver implementation, such as the number of tracking channels, constellations/frequencies, and antenna design, leading to hardware and software decisions. Consequently, the overall energy budget will be influenced by their definition. Therefore, establishing the test scenarios and assumptions is mandatory for proper analysis and comparison when looking at receiver performances.

2) *Wireless Environment*: The signal environment is a major factor driving the consumption of a GNSS receiver. For embedded and small-size devices, the effects from the environment have an even larger impact on the positioning, as they are more likely to be used in challenging conditions.

We define the environment as the physical and signal conditions where a receiver might be placed. For example, taking the case of a user in a city, its environment would be attributed to an “urban canyon” case. The consequence of such an environment on the signals are possible SNR reductions, multipath, interference, and physical blocking (e.g., buildings), with a potential decrease in the receiver performance. Longer integration time to recover the signals and more challenging tracking will undoubtedly lead to positioning precision reduction and longer TTFF. The consequences are not directly on power but on energy consumption. Moreover, using a specific interference-resistant algorithm would most probably increase complexity and power consumption.

Additional challenges from a wireless environment are the jamming and spoofing of GNSS signals, reviewed in Section V-C of this paper.

3) *External Aiding*: External aiding refers to additional components that can be set up along the GNSS chipset to provide further information about its environment, dynamics, the current constellation status, etc. It can be from additional sensors integrated alongside the receiver or communication links for information exchange.

Assisted-GNSS is a type of external aiding, where the GNSS receiver receives information via another communication link (e.g., terrestrial radio communication, mobile network, etc.) to help with the acquisition of signals [43].

Provided with a priori knowledge, such as an approximate position of the receiver or the almanac/ephemeris of the satellites, a receiver can reduce its search space significantly, leading to smaller TTFF and smaller energy consumption. It is related to the application and the capacities of the embedded devices, as it requires additional hardware besides the GNSS chipset to receive these other signals. It is also necessary to be in a connected environment, where reception of these assisting data is possible.

Similarly, other sensors (inertial, camera, etc.) can also be used to provide information about the receiver and help with its positioning. In [13], the use of Inertial Measurement Unit (IMU) for efficient duty cycling is reviewed along with the possible benefits for the energy budget, showing the potential of these additional sensors for reducing energy consumption.

4) *Hardware Technology*: In embedded devices, power consumption has been highly correlated with the technology node of electronics. Each new generation of processors has made them more efficient over the last decades. Therefore, one side of GNSS research is adapting new processor technology and proposing new architectures, which has led to significant power consumption reduction even in the last decade. While this is not the only reason for power reduction, evidence from these papers still shows a considerable correlation [7], [8], [9], [10].

However, comparing receiver electronics to highlight the best design/architecture is not straightforward. Research that focuses on RF Front-End (RFFE) or GNSS receivers usually provides a comparison table. The recurrent items are the type of Complementary Metal-Oxide Semiconductor (CMOS) process, GNSS/frequencies receivable, the area used, the current/voltage, and the power consumption. The components in these tables tend to vary depending on what is necessary to highlight, leading to incomplete comparison when the item cannot be recovered from previous studies. The first difficulty is a lack of a clear item list that should be present in any paper that would allow a fair comparison of each design to understand what might lead to an increase in

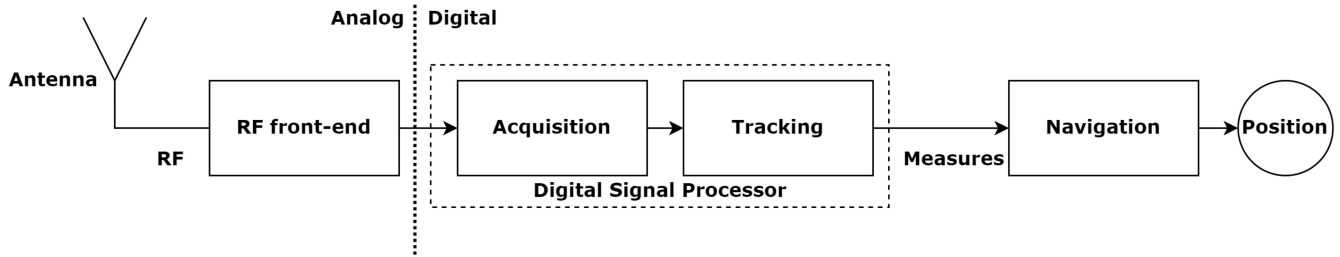


Fig. 7. Overview of the standard GNSS processing chain.

power consumption. In a recent comparison, the difference in tracked frequencies/constellations also leads to unfair comparison. New papers tend to compare multi-GNSS receivers with GPS-only receivers, justifying the excess consumption in their work to enable more constellations. While this sounds logical and probably true, it does not allow concrete comparison that could help to understand the benefits of their architecture among others.

#### D. Summary and Lessons Learned

In this section, we have reviewed the different papers looking into the factors influencing a receiver's power/energy consumption. The user requirements and operating environment are unalterable driving factors that will dictate the requirements for the receiver. Specific hardware or software will be needed to be implemented on the receiver to fit the requirements. The complexity of the hardware and software will dictate the power consumption. The overall implementation will have a certain activity time, finally leading to the receiver's energy consumption.

Finding the root cause for an increase in energy consumption inside a GNSS receiver is far from straightforward. An increase in energy consumption can be linked to many factors rooted in the receiver architecture, the hardware technology, or the scenario's environment. While these impacts can be qualitatively seen, quantifying each would require an environment with complete control of all the parts. Such control is typically not accessible for commercially manufactured receivers based on the limited information provided in their datasheets. Thus, we are restricted to the energy analysis performed in Table III and Figure 5. To figure out the individual effect of each factor on the energy consumed by a receiver, a fully controlled environment is needed and should be reviewed in future research.

A more abstract comparison metric is needed to escape from hardware implementations, such as algorithmic complexity estimation. When using such a metric, absolute energy consumption derivation is impossible. However, an algorithm can be used to estimate the relative differences between compared algorithms. While it may seem impractical to use these metrics to provide energy comparison on a theoretical level, we will see in Section VII that all GNSS research proposing new algorithms today rely on such high-level comparisons when deriving the new generation of "low-complexity" algorithms. In this review, we have assumed that algorithmic time complexity is sufficient to understand the relative energy

consumption cost between algorithms. Thus, in the following section, we will estimate the algorithmic complexity of the surveyed methods.

We have presented a list of factors that impact the energy consumption of a GNSS receiver. We will now dive into different architectures to compute a position, divided into onboard and offloaded processing.

## VII. ONBOARD PROCESSING

By onboard GNSS, we refer to the standard way of performing GNSS positioning: a fully integrated chipset with receiving, acquisition, tracking and navigation functionalities, all performed using the available hardware on the circuit board. The process chain overview is represented in Figure 7. The different processes present in the diagram will be reviewed in this section, explaining their implication on the processing chain and their requirements in terms of computational complexity.

In this section, "algorithm" regroups the acquisition, tracking, and navigation computations present in a GNSS receiver. Over the last decade, a significant part of GNSS research has been evaluating ways of reducing their complexity to save time and resources during the positioning. In parallel, the rise of new signals/constellations (Galileo, BeiDou) has created additional challenges compared to GPS signals. This section delves deeper into the algorithms presented as low complexity, with a specific focus on modernized signals tracking. Because acquisition and tracking are often highlighted as the processes leading to the highest computational requirements and are directly impacted by the modernization of signals, they will be reviewed extensively.

### A. GNSS Signal Acquisition

The acquisition phase is the first stage of the digital process chain. It has two objectives: (1) to find which satellites are present in the signal; (2) to find a coarse estimation of the Doppler frequency shift and code phase of each satellite's signal. To perform this, an unaided receiver must generate and test a wide range of signal replicas with different Doppler shifts and code phases. Due to this search space, the complexity of this process is high, and it is often highlighted as the most complex and resource-consuming phase of the digital process chain. Consequently, many publications looking into reducing computation complexity in GNSS receivers focus on the acquisition phase [44], [45], [46], [47], [48]. We limit the



scope of the survey to acquisition algorithms published during the last ten years. Moreover, we give special attention to algorithms focusing on acquiring the modernized signals.

Note that algorithm complexity is often reviewed based on the number of signal samples  $n$  in the papers surveyed, as more processed samples will necessarily lead to more computations. The definition is presented in [44]: for example, a GNSS receiver acquiring an L1 C/A signal needs at least 1 millisecond of data, the length of a full PRN code. Let us take a sampling frequency of 5 MHz, thus  $n = 5000$  for 1 millisecond of data. In case the sampling frequency increase, the time complexity of the algorithm increases accordingly.

1) *Acquiring Legacy Signals*: Based on the stated problem, the straightforward solution is to generate the signal replica with different Doppler and phase shifts and mix (i.e., convolve) it with the incoming signal, performing these computations in the time domain. The algorithm is referred to as Serial Search (SS); it can be parallelized easily to accelerate the search process, depending on the number of resources available on the hardware. This solution has a complexity of  $\mathcal{O}(n^2)$  [44], where  $n$  is the number of RF samples, as defined earlier.

An alternative method to this search was introduced by [49] and is known today as Parallel Code Phase Search (PCPS). The major improvement provided by this solution is the introduction of “circular cross-correlation” thanks to the shifting from time to frequency domain. Instead of generating a different replica for each code phase and performing an independent mixing, the Fourier Transform of the code and the incoming signal are computed and mixed, allowing to test the whole code phase in one go. These operations need to be performed for each frequency shift. Moreover, convolution in the time domain is a multiplication in the frequency domain. Therefore, the mixing operations are greatly reduced. The overall complexity is reduced to  $\mathcal{O}(n \log n)$  [44].

This frequency-domain method is still today the one with the most reduced complexity. Over the last decade, new algorithms based on the same principle have been proposed to reduce complexity further. In [44], the “QuickSync” algorithm is detailed, which proposes a two-step search space based on sparse Fast Fourier Transform (FFT). The sparse aspect is defined by aliasing the incoming signal, leading to a coarse correlation peak, spreading over multiple Doppler/phase values. Yet the search is reduced to only these few values, and fine correlation can be performed only on these values. This “divide and conquer” technique allows to reduce the complexity of the acquisition algorithm to  $\mathcal{O}(n\sqrt{\log n})$ , with similar performance as PCPS. The complexity can even be made linear by using a more aggressive aliasing parameter, reducing the performance for low SNR but also the complexity to  $\mathcal{O}(n)$ .

Finally, in [50], the reduction of the sampling frequency below the Nyquist rate was reviewed based on band-pass theory. It concerns superheterodyne receivers with an Intermediate Frequency (IF) stage, yet the study also shows improvements in baseband signals datasets. The algorithm used for acquisition is similar to the PCPS technique, and the reduction in complexity comes from the lower sampling rate. The algorithm introduces a *resampling* stage, where the incoming signal is down-sampled to lower the sampling frequency.

By lowering the number of samples to process, the number of operations is by definition reduced as well. The sensitivity of the acquisition was reduced, yet not significantly when compared to the results without resampling. Unfortunately, the complexity analysis was only provided in the number of additions/multiplications required. As this resampling strategy is based on the IF and the original sampling frequency, computation of the complexity is more difficult. It is to be noted that other sampling strategies such as Compressive Sensing have also been looked into for GNSS acquisition but are more related to offloaded computing. Thus, they will be presented in Section VIII.

2) *Acquiring Modernized Signals*: The techniques presented before can be used for both legacy and modernized signals. Yet modernized signals being more complex in their design, the straightforward application of these algorithms can lead to a considerable increase in complexity compared to legacy signals. This is due to their longer primary codes, secondary codes, and the higher sampling rates required, which directly increase the number of operations.

In [21], the PCPS technique is used to compare the L1/L5 and E1/E5a acquisition complexities. What is highlighted is that modernized signals such as L5/E1/E5a lead to a great increase in acquisition complexity if traditional methods are used. Even if Assisted-GNSS data are used for modernized signals, the complexity is reduced but still higher than for an L1 C/A acquisition. Note that in a specific case, with assisted-data and low sensitivity requirements, L5 acquisition might be less complex than L1 as it might require less memory. Yet, for E5a, the acquisition is always more complex than E1. In this context, sensitivity refers to the capacity of the receiver to successfully acquire a signal based on the signal’s SNR and receiver environment. The paper provided the most thorough complexity analysis, with time and memory complexity analyses for each signal parameter. It also considered many different parameters to realistically simulate the complexity, namely, the number of coherent/non-coherent integration, the zero-padding involved, and the latency of the FFT. Unfortunately, the analysis does not directly compare the complexities of E1 and L1 C/A. It also does not include a complexity analysis for other modernized signals like L1C, L2C, E6, or any of the BeiDou signals.

New techniques are used in the acquisition to deal with these more complex signals. The first challenge is the presence of secondary codes on pilot channels, which can happen between any primary codes and negate the acquisition, similarly to data bit transitions. For signals that only have a secondary code on their pilot component (e.g., L1C, E1, B1C), the acquisition can be realized with the data component and then switched to the pilot component during the tracking procedure for the secondary code lock. For the other signals (e.g., L5, E5, B1I, etc.), the most straightforward solution is to acquire the signal based on their secondary code instead of primary code. Yet secondary codes are much longer, resulting in a tremendous increase in computation requirements.

Another technique to counter this effect is based on zero-padding [52]. Zero-padding originally refers to padding the samples and code with zeros to increase the efficiency of the

FFT algorithm (i.e., RADIX-2 or 4), also known as Single Block Zero-Padding (SBZP). In our case, we refer to Double Block Zero-Padding (DBZP), where the code is split into small portions, and the correlation is realized on two consecutive portions of the signal. This requires that the code replica portions are padded with zero. In [51], DBZP is introduced for acquisition of Galileo E1 signals. Simulations showed that compared to traditional PCPS, DBZP can divide the number of operations by 2.2. Yet, the zero-padding induces a lot of unnecessary operations. The authors also propose a new method called Double Block Zero-Padding Transition Insensitive (DBZPTI) to mitigate the impact of bit transition. The complexity of this method is similar to DBZP, but its sensitivity strongly depends on the Doppler frequency.

Yet, both of these techniques suffer from the zero-padding, which induces a significant amount of redundant and unnecessary operations. A technique called “Split FFT” has been reviewed extensively in [47], [52] to mitigate the impacts of zero-padding with new signals. By splitting the FFT procedures into smaller-size FFTs, the effects over time and memory complexity can be mitigated and reduced compared to the single FFT solution. Reference [47] added to its complexity analysis the impact on Adaptive Logic Module (ALM), memory usage, and multipliers usage (DSP blocks) for an FPGA implementation. The 3-FFT algorithm leads to a 21% reduction in the number of computations, a 50% reduction in memory usage, and a 39% reduction in processing time. Reference [52] even further reduced the memory consumption with the 5/9/15-FFT solution, but with an increase in multipliers usage.

However, as pointed out by [48], such DBZP techniques are limited to non-coherent integration due to the possible secondary code sign transition, and the acquisition sensitivity is then limited by non-coherent loss. One solution is to go back to Serial Search, but this third dimension search is time-consuming. In [48], a SBZP solution is proposed using the Partial Correlation Method (PCM) introduced by [20]. The main idea is to combine the results of multiple partial correlations to increase the coherent integration process leading to a higher acquisition sensitivity. Further details are provided in the article and will not be explored here due to the algorithm’s complexity. The paper provides a complexity analysis, which unfortunately only compares the algorithm to other PCM approaches. The algorithm is claimed to have a complexity of  $\mathcal{O}(6 \cdot m \cdot n \log(n/m))$ , where  $n$  is the total number of samples divided into  $m$  correlation blocks. It is higher than the PCPS method, given the analysis from [44], yet the sensitivity of acquisition can be theoretically significantly increased thanks to the coherent integration.

Code folding, introduced in the Extended Replica Folding Acquisition Search Technique (XFAST) algorithm for GPS P-Code decoding [53], has also been suggested to deal with the long codes and secondary code transitions. In [45], a modified XFAST method is presented, with a reduced complexity claimed  $\mathcal{O}(n/2 \log n/4)$ . Yet, the SNR is significantly impacted due to cross-correlation interference appearing with code folding. Thus, its interest in difficult signal environments is limited.

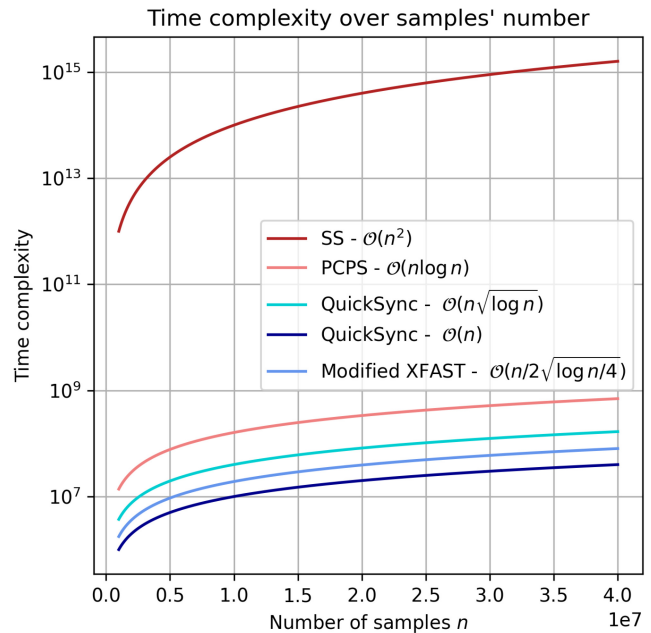


Fig. 8. Time complexity based on the number of samples.

Another challenge of modernized signal acquisition is due to the BOC ACF shape, leading to side peaks around the central peaks. Depending on the signal quality and user environment, it can lead to a wrong estimation of the code phase during acquisition and a wrong lock during tracking. A workaround is to transform back the ACF to a single-peak shape for the acquisition. In [46], a method is proposed by applying a seven nonzero-tap triangular filter over the signal before performing the acquisition. The complexity is analyzed in a number of operations and showed significant reductions compared to the research from [54] based on the principle with a different method. Unfortunately, it is not compared to other well-known methods, complicating its performance analysis.

3) *Complexity Analysis:* All the research papers presented provided a form of complexity analysis, as they focused on reducing the complexity of the acquisition phase. Table IV summarizes the different methods presented, along with their mathematical time complexity expressions when available. Methods providing a time complexity have been represented in Figure 8. The SBZP is not represented as it also depends on the number of correlation blocks  $m$ . The range of samples’ number  $n$  was chosen based on typical sampling frequency (e.g., 5-40 MHz), as defined before. Since the PCPS method was introduced, most research focusing on complexity reduction used FFT-based algorithms. However, high sensitivity is necessary for acquisition in a challenging environment, and the trade-off between complexity and performance should be reviewed for an optimized algorithm selection.

Table IV also shows that complexity analysis can be heterogeneous, from mathematical time complexity analysis to a number of elementary operations. Both metrics are valid for the absolute analysis of algorithm complexity. Yet, for relative comparison between algorithms, this heterogeneity leads to difficulties highlighting the differences between two studies, as they might not be compared over the same reference algorithm.

TABLE IV  
SUMMARY OF ACQUISITION ALGORITHMS FOR MODERNIZED SIGNAL

Technique	Source	Complexity analysis	Expression	Notable features
SS	[44]	Time complexity	$\mathcal{O}(n^2)$	Straightforward implementation but high complexity.
PCPS	[44], [49]	Time complexity	$\mathcal{O}(n \log n)$	Complexity reduced by translating to the frequency domain. All following algorithms are based on this method.
QuickSync	[44]	Time complexity	$\mathcal{O}(n\sqrt{\log n})$ or $\mathcal{O}(n)$	Complexity of PCPS further reduced by performing a coarse and fine search.
Modified XFAST	[45]	Time complexity	$\mathcal{O}(\frac{n}{2}\sqrt{\log \frac{n}{4}})$	Code folding applied to modern signal acquisition. Complexity further reduced but SNR decreased.
DBZP / DBZPTI	[51]	Number of operations	N/A	Complexity analysis of DBZP, reduced by half compared to PCPS. Proposed a new method resistant to bit transition (DBZPTI).
Side Peak Cancellation	[46]	Number of operations	N/A	Side peak effects canceled, but complexity analysis too restrained.
Split FFT	[52]	DSP block reduction	N/A	Complexity further reduced compared to DBZP, but limits the range of sampling frequencies.
PCS mSBZP PCM	[48]	MAC operations	$\mathcal{O}(6 \cdot m \cdot n \log(n/m))$	Complexity is higher than PCPS, but with an increase in sensitivity.

Additionally, differences in performance metrics (e.g., sensitivity) lead to more difficult comparisons when trying to understand the trade-off between complexity and performance of a particular algorithm.

### B. GNSS Signal Tracking

After successfully identifying a signal and estimating the Doppler/code phase parameters by the acquisition phase, the tracking phase refines these parameters to align the code replica precisely. Once this is achieved, the code can be removed from the signal only to be left with the data, and the decoding phase can be started. Contrary to the acquisition phase, the tracking phase is essentially impacted by the modulation used, either BPSK or BOC. Modernized signals using a BOC modulation cannot be properly tracked using algorithms designed for BPSK signals [11]. The BOC ACF is different from legacy BPSK ACF, with additional peaks located around a central one (see Sections IV-D and V-A). It can lead to ambiguous tracking, as a false lock on a side peak could occur, with dramatic consequences on the measurement accuracy. Thus, we differentiate BPSK and BOC tracking techniques and evaluate their respective complexity.

1) *Tracking BPSK Signals*: Looking further into the process, a tracking loop will try to align the frequency and the code of the replica. The receiver generates several versions of the replica, shifted by a fraction of a chip. These replicas are then correlated with the incoming signal, and the outputs are compared to understand if the replica is too early/late. These different versions of replicas are usually called *correlators*. The ACF of the legacy Binary Phase Shift Keying (BPSK) signal is a single-peak shape (i.e., triangular), given by the square shape of binary PRN code. Three correlators are usually used for successful tracking, as they are sufficient to fully capture and track the signal in an *unambiguous* manner. An

Early and Late correlator for tracking the code delay in the Delay Lock Loop (DLL), and a Prompt correlator for tracking the frequency in Phase Lock Loop (PLL) or Frequency Lock Loop (FLL). Once compared, the replica is corrected from its possible shifts in code and frequency for the next loop. If the signal is successfully aligned, which can take several tens of milliseconds depending on the signal quality, the code can be completely removed from the signal, and the data can be decoded.

While there can be different implementations of the DLL/PLL/FLL discriminator or filter order, the concept of tracking has stayed relatively the same over the years, unlike to the acquisition algorithm. Additional correlators have proven to lead to better resistance to multipath, as this phenomenon might create distortions and additional peaks in the ACF, leading to biases. For example, the Multipath Estimating Delay Lock Loop (MEDLL) algorithm presented by [55] uses 10 or more correlators. Consequently, such an algorithm is much more complex than the Early-Prompt-Late (EPL) method, requiring more correlators.

2) *Tracking BOC Signals*: The BOC modulation has a ACF which can lead to ambiguous tracking if BPSK techniques are used. Solutions for unambiguous tracking of modernized signal can be divided in four categories: (1) BPSK-like techniques; (2) Bump-Jumping (BJ) techniques; (3) side-peaks cancellation techniques; (4) sub-carrier tracking techniques. The first two groups BPSK-like and BJ have limited performances and rather high complexity. BPSK-like methods tracks each side lobe of the BOC signal independently [56]. As a result, the ACF is unambiguous, but the number of correlators greatly increases compared to BPSK tracking. Moreover, the sharp correlation peak of BOC is lost, and the measurement precision is reverted to BPSK level. BJ methods start by tracking a peak and identifying if this is the central peak. If not, the tracking “jump” towards the right peak. This was

TABLE V  
SUMMARY OF TRACKING ALGORITHMS FOR MODERNIZED SIGNAL. CORRELATORS ARE CONSIDER COMPLEX.  
CORRELATORS COUNT DOES NOT ACCOUNT FOR DOPPLER REMOVAL OPERATIONS

Technique	Source	Category	Correlators	Notable features
EPL	[18]	Traditional	3	Simplest implementation for code tracking. Ambiguous for BOC tracking.
MEDLL	[71]	Traditional	10+	Multipath resistant tracking but with increased computational complexity. Ambiguous for BOC tracking.
BJ	[57]	Bump-Jumping	5	EPL with VE and VL correlator for main peak identification. Can be ambiguous for BOC in low-SNR conditions.
BPSK-like	[56]	BPSK-like	6+	Unambiguous BOC tracking but reduced measurement precision.
ASPeCT	[59]	Side-peaks cancellation	5	First side-peak cancellation method. Unambiguous tracking for sine-BOC( $n, n$ ) only.
DE	[68]	Sub-carrier tracking	7	First sub-carrier tracking method. Unambiguous tracking for all BOC.
PUDLL	[61]	Side-peaks cancellation	4	Reduced complexity and similar tracking precision w.r.t. ASPeCT. Application extended to sine-BOC( $kn, n$ ).
GRASS	[60]	Side-peaks cancellation	5	Application extended to sine-BOC( $kn, n$ ). Lacking performance comparison with other methods.
SPAR	[58]	Side-peaks cancellation	3	Reduced computational complexity but lacking performance comparison with similar methods.
Astrium	[67]	Sub-carrier tracking	5	Reduced computational complexity and similar tracking precision w.r.t. DE.
DPE	[69]	Sub-carrier tracking	6	Reduced computational complexity but lacking performance comparison with similar methods.
IGRASS	[62]	Side-peaks cancellation	5	Application extended to all BOC, with similar performances compared with methods for sine-BOC( $kn, n$ ).
YT V1/V2	[63]	Side-peaks cancellation	4	Increased performances for high-order sine-BOC. Performances compared with PUDLL and SPAR methods.
DDPE	[70]	Sub-carrier tracking	8	Reduced computational complexity for high-order BOC, with relatively low degradation in tracking accuracy.
MCCF	[64]	Side-peaks cancellation	4	Increased performances compared to PUDLL, SPAR and YT V1/V2. Complexity is increased as it requires multi-level operations.
BUT-EAS	[65]	Side-peaks cancellation	4	Thorough comparison with other side-peaks cancellation methods. Relatively low computational complexity and good performances. Application extended to sine-BOC( $m, n$ ).
SRSA	[66]	Side-peaks cancellation	4	Reduced computational complexity and similar tracking precision w.r.t. ASPeCT. Application extended to all BOC.

first described in [57], where two extra correlators called Very-Early (VE) and Very-Late (VL) are added. The correlators are used to identify if the current tracked peak is the central peak. However, it is based on magnitude comparison, which can fail to identify the right peak in a low SNR environment [58].

Due to these limitations, most algorithms proposed over the last decade have looked into side-peaks cancellation and sub-carrier tracking techniques. Many methods have been presented over the years and reviewed by survey papers [11], [12]. This work is meant to look further into the complexity aspects of these methods, with a particular focus on low-complexity methods. Table V proposes a summary of the methods presented.

Side-peaks cancellation techniques focus on removing the additional peaks created in the ACF while keeping the narrow central peak. It is achieved by creating different modified signal replicas to be correlated with the incoming signal and combining the correlation results. The first method using this principle is called Auto-Correlation Side-Peak Cancellation Technique (ASPeCT) and was proposed by [59]. It is only applicable to sine-BOC( $n, n$ ) modulation, yet it requires minimal changes in receiver design. Consequently, many

algorithms have been proposed during the last decade, building upon the original idea and trying to generalize it to all BOC [58], [60], [61], [62], [63], [64], [65], [66]. Details of each implementation are tedious and require dedicated sections for a proper description. This is outside the scope of this paper and thus will not be reviewed. A comprehensive comparison of the algorithms' quality performances can be found in [65]. Note that the more recent methods have been looking into increased performance for high-order BOC signals, which are significantly more computationally expensive to track than low-order BOC when using legacy methods [11]. Side-peak cancellation methods showed relatively straightforward implementations, yet most of the methods presented are only applicable to sine-BOC signals. Moreover, only Galileo PRS signals use a cosine-BOC modulation, thus making these methods suitable for open and commercial signals across constellations.

The final group are sub-carrier tracking techniques, which process the sub-carrier component like the traditional carrier. Thus, a dedicated loop called Sub-carrier Lock Loop (SLL) is added, resulting in independent tracking of the code and the sub-carrier. Two delays are recovered: an ambiguous but



accurate delay from the SLL and an unambiguous but less accurate delay from the DLL [67]. The delays are joined to have an unambiguous and accurate delay estimation, fully exploiting the accuracy of the BOC modulation. The first method to introduce this concept is the Double Estimator (DE) by [68]. An EPL loop is implemented for the SLL before the DLL. In [67], the same principle is used for the “Astrium” method and compared against DE. By tying the sub-carrier and PRN code estimation together, the DLL can be removed to keep a PLL/FLL and SLL loop. This allows to reduce the number of complex correlations. In [69], another method called Double Phase Estimator (DPE) is introduced. The sub-carrier is approximated as pure sinusoid and the SLL is replaced by a PLL for the sub-carrier i.e., a Sub-carrier Phase Lock Loop (SPLL). The results are compared to DE, with close performances but a lower computational load. In [70], the high sampling rate required for DE and DPE is pointed out and new method called Decimation Double Phase Estimator (DDPE) is proposed. The tracking process is divided into two parts: (1) removal of the nominal carrier and sub-carrier using constant frequencies, lowering the sampling rate; (2) tracking of the residual carrier and sub-carrier Doppler frequencies and code at a decimated rate. Limited SNR degradation and similar measurement accuracy can be observed between the DPE and DDPE. Moreover, the paper details a computation cost reduction of 86% for a BOC(15, 2.5) (e.g., Galileo E1-A PRS) signal tracking using DDPE compared to DPE.

3) *Complexity Analysis*: Complexity comparison of tracking algorithms is not trivial. As seen in the work from [11], often the comparison needs to be reduced to the relative grouping into classes (e.g., low/medium/high), left to the reviewer’s discretion. Research in acquisition presented in Section VII-A already showed flaws in algorithmic complexity analysis. Yet from the methods reviewed for tracking, the complexity analysis is often not expressed in mathematical terms. Instead, tracking algorithms are usually compared in complexity based on the number of correlators used. As correlations are computationally hungry tasks, it is a valid metric for complexity comparison. However, since algorithms also perform additional processing tasks around these operations, they might not be captured by only counting the number of correlators.

As tracking operation only requires a few correlation operations, it can be assumed to be less complex than the algorithm reviewed for the acquisition phase. Consequently, the acquisition is often highlighted as the most complex operation in the chain, which could explain why so much research focuses on reducing its complexity. Yet acquisition would only happen once in an ideal case scenario, whereas tracking is a continuous process after the signal has been acquired. It is similar to the fundamental difference between power and energy consumption pointed out in Section VI-A1. As the goal is to minimize energy consumption, it could be more beneficial in terms of absolute energy consumed to slightly reduce a tracking algorithm complexity compared to an acquisition complexity. However, such a hypothesis is difficult to verify from the results of the papers reviewed.

Table V summarizes the tracking algorithms reviewed, detailing the complexity analyses performed by the authors and the

number of correlators used. It can be seen that the number of correlators is an interesting metric to differentiate techniques and perhaps get a first idea of their complexities. Yet getting the correlator number is not always straightforward. One has to account for real and complex correlators, which can be mislabeled or unspecified depending on the studies. Moreover, some might or might not also include correlation operations for carrier Doppler removal and PLL/FLL tracking. Therefore for complexity comparison purposes, the method should be carefully reviewed to extract the correct number of correlators. In Table V, the number displayed account for complex correlations and does not account for Doppler removal operations since we are interested in the DLL tracking loop.

In [64], the authors also differentiate multi-level (e.g., bit quantization of two and more) from binary level (e.g., bit quantization equal to one) correlation operations. According to [54], a reduced-complexity correlation method can be applied if the reference code is kept at  $\pm 1$  level. In this case, complex multiplication can be replaced by additions and sign inversions in the correlation. For PUDLL, the replica signals are multi-level, thus leading to a higher computation complexity than MCCF and BUT-EAS [65] even though their total number of correlators are equal. Additional details are needed for better complexity comparison.

Therefore, while the number of correlators can provide an overview of complexity, it is insufficient for a comprehensive review. In [66], [70], the number of operations is reviewed. Yet, similarly to acquisition methods, the lack of homogeneity in complexity analysis makes comparison difficult.

Reducing the complexity of tracking algorithms is challenging, given that no significant breakthrough has been published over the last decades. In [13], the problem is attacked from a different angle: instead of reducing the complexity, a duty cycling strategy is implemented to allow intermittent tracking. Tracking requires continuous operation to align the code replica correctly. The replica alignment can degrade quite quickly depending on the receiver oscillator drift and the relative receiver movement. This intermittent tracking is usually not performed for more than a few milliseconds. The duty cycling strategy proposed by [13] is to extend this non-tracking time up to a few seconds, using the information from inertial sensors (i.e., IMU) to predict the alignment. The implementation provided good results, yet no in-depth complexity and power consumption analysis is presented. Moreover, this solution was only demonstrated in software on pre-recorded data and not implemented on hardware, which does not directly show the improvement in energy consumption and could lead to quite different results in real experimentation.

### C. Navigation

Once tracking of the signal is ongoing, the receiver can perform the positioning task by reconstructing the GNSS measurements (i.e., pseudoranges, phases, etc.). Navigation algorithms being so diverse, deriving complexity is even more dependent on the application case than on acquisition/tracking. Multiple techniques exist, such as Standard Point Positioning (SPP), Differential GNSS (DGNSS), Extended Kalman Filter

(EKF), and Precise Point Positioning (PPP), which do not require the same measurements, precision or external data. For example, a full PPP algorithm will require very precise orbital products (ephemeris and clocks) to solve the phase ambiguities. Moreover, the process of ambiguity searching is quite a computationally hungry task depending on the number of satellites tracked.

To simplify the analysis, we focus on the complexity of SPP, the conventional way of computing a positioning from satellite measurements. A Least Square Estimation (LSE) is used to minimize the residual errors and the coordinates and clock error of the receiver. It corresponds to a set of matrix multiplications and inversions. In [72], multiplication of matrix of size  $n \times n$  is proved to have a complexity of  $\mathcal{O}(n^3)$ , and matrix inversion has a similar complexity.

Yet, contrary to the DSP phase, the frequency of the computations is much lower (e.g., 1 to 10 Hz), and the size of the matrices  $n$  is relatively small. Compared to the complexity of acquisition/tracking, the navigation process can be almost discarded in the computations budget.

#### D. Summary and Lessons Learned

This section reviewed low-complexity techniques applicable to the acquisition and tracking of legacy and modern GNSS signals, focusing on their algorithm complexity estimation. These techniques can be further reviewed in the references [11], [12], [48], [73].

Acquisition and tracking have been the most reviewed segments of the GNSS processing chain over the last decade, as seen from this review. We showed that since navigation computations require a much smaller number of operations due to their lower computation frequency, they are not of interest when exploring the complexity factors in the processing chain. The complexity of tracking algorithms has been highlighted as much lower than acquisition algorithms but happening for a longer time and continuously, which corresponds to the energy analysis found in [15]. It is an important point for low-power GNSS, as the acquisition is often highlighted as the most computationally complex phase. Note that only non-assisted processes have been reviewed for the acquisition methods, meaning without any previous knowledge of the satellite data (e.g., satellite almanacs). Providing Assisted GNSS (A-GNSS) data would drastically reduce the search space and, therefore, the complexity of the computations, yet it is limited to a connected environment only.

The tracking phase showed to be the longest one, as it is the default mode of a channel after successful acquisition. Thus, even though the acquisition showed higher computation complexity, the energy would be mostly consumed during the tracking phase. This is crucial for an embedded device, which has finite energy resources. A small decrease in tracking complexity should, thus, lead to consequent energy reductions over time. For efficient decrease of energy consumption, future research should prioritize the reduction of computation complexity in the tracking phase over the acquisition phase.

Yet, the efficiency of the tracking algorithms for BPSK signals seems to have reached an optimum, as no significant

improvement has been demonstrated over the last decade besides severely lowering the tracking frequency. Instead, research has focused on tracking modernized signals with BOC modulation, which requires completely new tracking methods. Given the several algorithms reviewed in this section, they all showed a higher algorithmic complexity than legacy signal tracking, based on the number of correlation operations needed. It has been noted that other modernized signals, like the ones present on the L5 frequency, do not use the BOC modulation (recall Table II). While they can use the same algorithms as the legacy signals, their higher code rate requires a higher sampling rate for successful tracking and make them more computationally complex than legacy signals.

As modern signals come with additional functionalities (see Section III), the corresponding increase in complexity needs to be overcome. The prime challenge today for the new generation of low-power GNSS receivers is unequivocally the way to accommodate modern signals while keeping similar energy consumption as for legacy signals.

Finally, we observed that complexity-based estimation might be sufficient to review the differences between algorithms and their impact on energy consumption at a theoretical level. However, it is insufficient to clearly understand a GNSS receiver's power and energy consumption. While the surveyed papers provide a mathematical assessment of the algorithms, an energy-consumption estimations would be too dependent on hardware variations, user requirements, and signal environment. Recall that these were defined as the prime energy consumption factors in Section VI. We believe that a theoretical assessment of the whole processing chain would be time-consuming and inefficient. It would be based on many assumptions and could only provide a gross insight that might still be far from the actual energy consumption once implemented in a real-life scenario. The lack of common metrics and the heterogeneity of methods used for complexity estimation create ambiguities in comparing the algorithms and understand which is the most suitable for a specific application case. Further research on the algorithmic complexity is needed to successfully highlight the best methods for the best use cases. This topic is further reviewed in Section IX.

## VIII. OFFLOADED PROCESSING

In the previous section, we reviewed the case of performing all the signal processing tasks onboard the receiver, which has been the default way of processing for decades.

In offloaded GNSS, high-complexity tasks are outsourced to a remote hardware, external to the one receiving the signal. This external hardware can take multiple forms: from a server in a data centre (i.e., Cloud computing) to a closer device (i.e., Edge computing) [74]. This is represented in Figure 9. Most studies focus on a Cloud-based architecture, where a vast number of receivers could send their data for centralized processing. For this paper, the device receiving the RF signals will be named *receiver*, and the external hardware will be referred to as *server*.

As reviewed in Section VII, the tasks with a high computation complexity are acquisition and tracking. If those tasks are

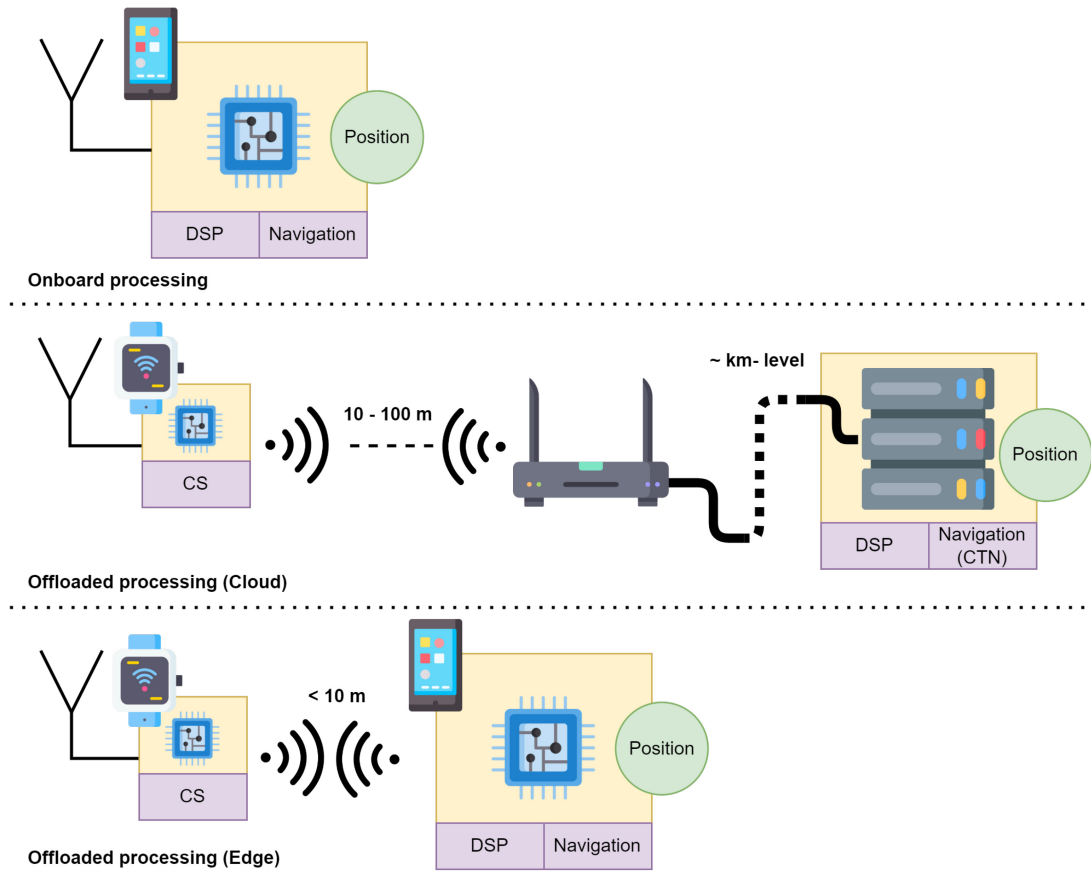


Fig. 9. Comparison between onboard (top), Cloud offloaded (middle), and Edge offloaded processing (bottom).

outsourced, the GNSS receiver simply becomes an RF logger. Saving raw RF signals for GNSS post-processing is not a new concept and is used for any software-defined receiver [75]. Recently, studies have demonstrated the applicability of offline loggers (i.e., with no communication module) for certain applications like lightweight tags for wildlife monitoring [76], [77]. However, such implementations require the tag to be physically retrieved for processing the signals, as the data is saved on a Secure Digital (SD) card. The transmission of the signals to a server can solve this issue, yet it presents challenges and is unsuitable for every application. This section focuses on GNSS offloading with a communication link to a remote server for processing. We identify the necessary techniques for performing an energy-efficient offloading and review the challenges and application cases of the current state-of-the-art research.

#### A. Processing Strategy

The major problem with offloading GNSS computations is the amount of data to be transmitted. Focusing on a mono-frequency dataset, this amount depends on the sampling frequency, the quantization, and the recording length. Taking the simple case of GPS L1 C/A, the main lobe bandwidth is 2.046 MHz [21], leading to a minimum sampling frequency of 4.092 MHz to fulfil the Nyquist rate condition. Bit quantization can be reduced to 1-bit, with a loss of 1.96 dB in an ideal environment case [18]. Finally, in a standard positioning

scenario, 6 to 30 seconds of signals should be recorded, depending on the start status of the receiver (cold/warm/hot), to finally reconstruct the pseudoranges (see Section VIII-A1). Taking the case where ephemeris is known by the receiver and six seconds of data are recorded, then  $6 \cdot 1 \cdot 4.092e6 = 24.552$  Mbits (3.3 MBytes) needs to be transmitted to the server. Additionally, overheads due to the communication link protocols should be accounted for in the total number of bits to transmit. For an IoT device, this is a substantial amount of data to send, limiting the number of possible wireless communication links for energy-efficient data retrieval. For example, the IEEE 802.15.4 technical standard for Low-Rate Wireless Personal Area Network (LR-WPAN) [78] is the basis for several low-power communication protocols used in IoT. In its definition, transfer rates range from 20 to 250 kbit/s. Thus, in the best case, it would require more than 100 seconds of data transmission to retrieve just six seconds of data necessary for the standard positioning approach. Without going into the budget of such a link, it is clear that the energy required for such an architecture renders it completely out of the question.

Notably, two major concepts have to be introduced to diminish the communication link requirements, namely, Coarse-Time Navigation (CTN) and Compressive Sensing (CS).

1) *Coarse-Time Navigation*: In a standard positioning scenario, the receiver goes into the tracking process to align its code replica with the incoming signal to extract the data within the signal [18]. While all the navigation data might not be required, depending on the receiver state, it is mandatory to

find the TOW to build the pseudorange measurements (see Sections III and IV). The TOW is always present in every sub-frame of the navigation message, which has a length of six seconds. Therefore, theoretically, if a receiver wants to compute the pseudoranges, it needs a minimum of six seconds of data to be certain the TOW is present and retrieve it. Of course, if the receiver has some previous knowledge about the TOW position in the message, this time can technically be reduced but may require more complex hardware and/or communication link.

In [43], another method to compute the pseudoranges is presented, named Coarse-Time Navigation (CTN), which provides a way of computing the pseudorange measurements without decoding the TOW and without having a precise knowledge of the absolute time (i.e., time uncertainty in the receiver  $> 2$  s). To build the pseudoranges, the time of transmission is needed. Yet, in CTN, both transmitted and received times are not precisely known, leading to a scale issue. Since the resolution of a pseudorange measurement is based on the C/A code length (i.e., 1 ms or 300 km), the receiver does not know how many codes preceded its reception. It is known as the “code ambiguity” (a.k.a. “millisecond ambiguity”). In a standard computation, if the receiver already accurately estimates its time and position, this ambiguity is solved instantly. If the receiver does not have this knowledge, it can be estimated using the transmitted time from the navigation message, and the receiver clock error is computed in the navigation least-square minimization process. In the CTN process [43], the additional unknown  $\delta_{tc}$  is introduced to represent the transmit time error. The measurement model of a pseudorange  $P$  can be reviewed as follows

$$P = \rho + \delta_b + \dot{\rho} \cdot \delta_{tc} + \epsilon, \quad (1)$$

where  $\rho$  is the geometric range,  $\dot{\rho}$  is the geometric range rate (including the satellite clock error),  $\delta_b$  is the receiver clock bias, and  $\epsilon$  the unmodeled errors. While  $\delta_{tc}$  might seem very similar to the receiver clock error  $\delta_b$ , it will affect each satellite’s measurements differently, as the time of transmission is also used to compute the satellite’s positions and clock corrections in the measurement model. In the end, we obtain a model with five unknowns instead of four, namely, the position of the receiver ( $x, y, z$ ), the receiver clock bias, and the transmit time error. A minimum of five satellites is therefore needed to solve this problem. It is referred to as the *5-state equation*.

At this point, the code ambiguity is still present, and minimization over this model might result in large positioning errors. To solve this, the authors of [43] proposed a two-stage algorithm, where the code ambiguity integer is solved first to obtain the pseudoranges. In the second part, the 5-state equation is used to obtain an estimation of the unknowns. The procedure is given in detail in [43].

This technique allows pseudorange reconstruction with less than 6 seconds of data. Several studies [79], [80] have demonstrated that it is possible to reduce the recording time to a minimum of 1 ms, i.e., one complete C/A code. Yet, for a better result, it is advised to record a minimum of 20ms. The GNSS logger only needs to transmit 20ms of the signal instead

of six seconds. Returning to the previous computations for GPS L1 C/A signal, this corresponds to  $0.020 \cdot 1 \cdot 4.092e6 = 81.84$  kbits (10.23 kBytes). This number is much more in line with the capacity of an IoT communication network. Yet the IEEE 802.15.4 technical standard highlights data rates as low as 20 kbps, which would still require several seconds to extract fully the data from the device without accounting for the communication link overheads. Data compression is needed to reduce further the amount to transmit.

2) *Compressive Sensing*: Quantization and sample time length have already been reduced to their minimum possible value, respectively 1-bit quantization and 1 millisecond sample. To further reduce the size of the data, the only factor left is the sampling frequency. Over the last two decades, significant progress in signal reconstruction has led to a breakthrough in signal compression, pushing sampling frequency to a level lower than the Nyquist rate. This technique is called Compressive Sensing (CS) and has been first described in [81] and explained quite extensively in [82]. The main idea behind CS is that any “natural” signal is compressible, meaning that it can be represented by a *sparse vector* on a certain transform *basis*. The design of the basis is an important part of the CS theory. This basis can be viewed as a *dictionary*, containing simple pieces that, added together, can represent a complex signal. For example, the generic bases commonly used are Fourier or Wavelet bases. Based on the CS, if a signal is represented on such a basis, only a limited number of measurements is required to represent this signal, leading to a sub-sampling opportunity. This sub-sampling can be much lower than the theoretical Nyquist rate. A summary of the mathematical definition of CS is provided in the Appendix. An extensive explanation can be found in [82].

In summary, the only computation requirement on the logger side is sub-sampling the received data. On the remote server side, complex processing is needed to recover the original signal (see the Appendix). However, such processing is not an issue since the remote server is not limited in energy or processing resources in this architecture.

## B. Current Literature Landscape on Offloaded Processing

This section reviews the research status of GNSS offloading in the literature, limiting the research scope over the last decade. Several architecture types have been identified that off-load processing at different measurement stages. We review these architectures, highlighting their advantages and disadvantages, and discuss the limits of the presented analysis.

1) *Application With Coarse-Time Navigation Only*: In [83], the authors present the Low Energy-Assisted Positioning (LEAP) algorithm. The presented architecture is based on CTN, which only requires the code phase estimate from the tracking process. The receiver logs a few milliseconds of data and performs the acquisition/tracking process. The code phase estimates for each satellite found are offloaded to the remote computer. For 12 satellites tracked, 40 bytes (320 bits) are assumed to be sent through the cellular network. CTN computations are then performed on the remote computer, assuming access to satellite ephemeris (broadcast or precise). To further



reduce energy consumption on the receiver, a duty-cycling scheme is enforced, leading to the shutdown of the receiver from 10 to 60 seconds, depending on the application parameters. At each wake-up, the receiver needs to re-perform the acquisition and tracking process, thus, limiting shutdown periods to less than 60 seconds still allows for reducing the search space of the acquisition process based on the user and satellite theoretical dynamics. A position accuracy analysis has been performed using a GPS hand-held commercial device as a reference. Over multiple datasets, it showed an average positioning performance from 200 m to 10 m, impacted by the number of satellites retrieved (5 to 8), the tracking length (10 to 500 ms), and the distance to the nearest “cell tower” (0.2 to 3 km).

The advantage of such an architecture is the small amount of data that needs to be transmitted to the remote computer while minimizing the receiver ON time. The disadvantage is that the receiver still has many computations to perform, requiring significant processing and memory capacities. Moreover, while the duty cycling strategy drastically reduces the tracking time, the acquisition has to be performed frequently, although on a reduced search space. A comparison between LEAP and a “typical” GPS receiver is provided and shows energy savings ranging from 60% and 80%. It assumes an average power consumption of 400 mW for the receiver, which is a rather conservative assumption from the state-of-the-art receivers of that time. The paper thus assumes an energy consumption based on this power and an ON time derived from their analysis. It might provide a coarse ballpark value, but it still fails to include the energy used for communication with the mobile network. This energy, most probably significant in the budget, is needed to fairly compare onboard and offloaded processing.

In [84], the authors present a service called Cloud-Offloaded GPS (CO-GPS). Similarly to [83], the navigation process is based on CTN, but all the processing is offloaded from the receiver. In their strategy, the receiver acts solely as a logger, receiving 2 to 10 ms of data and storing it for later processing. On the remote computer side, the processing starts with the acquisition. The difference with [83] is that no tracking is performed, and the coarse code phase estimation is used directly in the CTN computations. On average, it produces a location accuracy of 35 m using 10 ms of data.

The major advantage of such architecture is the number of computations on the receiver, which is reduced to its minimum since it only logs RF signals and sends them to the remote computer. However, compared to [83], the amount of data transmitted is increased by three orders of magnitude (40 B vs. 40 kB). Yet this aspect and impact on the transmission link are not reviewed in the paper. For a comparative analysis with onboard consumption, the communication link to offload this amount of data should be considered, as it will require a significant amount of energy. The energy consumption analysis is derived from the experiment because a specific electronic board was designed for the signal reception. Yet, only the signal reception and the memory writing phase are captured, as no communication transceiver is implemented on the board.

In a white paper recently published [14], low-cost GNSS positioning techniques have been reviewed. Among other

techniques for power consumption reduction, the snapshot positioning technique is shown as viable for low-rate positioning. For applications requiring only a few position updates per day and at a maximum of one per 2 hours, the communication to cellular connection Long Term Evolution (4G), category M1 (LTE-M) is still energy efficient. It also introduces a new service called “CloudLocate” [85]. This service claims “up to 10 times energy savings compared to standalone GNSS power savings approaches”, with an accuracy “under 5 meters in most cases”. The technology is directly available on multiple System-on-Chip (SoC) receivers from the company and can be used on a wider range of connectivity technology. No details regarding the processing architecture are given on either the user side or the server side. A little information is provided in the CloudLocate datasheet [85]: only three seconds of up time for the GNSS receiver is needed; the data packet size sent by the receiver ranges from 12 to 50 bytes. Given these, the architecture can be assumed to be similar to [83], meaning acquisition and tracking are performed on the user side, and only the code phase information is uploaded to the server using a cellular LTE-M connection, where CTN is performed. Most probably, no raw RF signal or CS compression technique is used, as the packet sizes are too small.

Unfortunately, no precise metrics, experimental results, or datasets were provided in [14] to further assess the service’s quality or positioning. Yet, it is interesting to see an industrial implementation of offloaded GNSS, as it might be a significant proof of its energy efficiency and economical viability.

2) *Application With Coarse-Time Navigation and Compressive Sensing*: In [86], the authors present the Sparse-GPS (S-GPS) algorithm. Contrary to the previous sections, the research focuses primarily on reducing the amount of data sent to the remote computer. S-GPS uses the principles of CS while associating it with the acquisition process. As discussed previously, the dictionary definition in CS is crucial. The strategy adopted by the author is to consider a dictionary made from a different version of the PRN code, each possible code phase, and the frequency shift combination for each satellite listed. The dictionary is composed of each *bins* from the conventional acquisition process. The reconstruction and acquisition process is performed at the same step, which skips the need first to reconstruct the signal and then perform the lengthy acquisition process over the reconstructed signal. The CS process of S-GPS is thoroughly described as the dictionary creation process.

No tracking algorithm is reviewed by the paper, although a resulting position accuracy is proposed in the result section. It is assumed that CTN was computed based on the acquisition results directly, similarly to [84]. The research team tested several configurations for the compression ratio of the measurement vector, from 70% to 90% compression, and analyzed the acquisition and positioning results. The number of milliseconds of data to be processed (from 2 to 10 ms) is also reviewed, as these factors influence the amount of data required and the accuracy of the results. As expected, more data and lower compression rates provide the best results, yet there is room to optimize both factors for optimal accuracy and data size. The average position accuracy is around 50m and is

TABLE VI  
EXAMPLES OF DATA REQUIRED TO PRODUCE POSITION THROUGH OFF-LOADING, USING THE CTN AND CS TECHNIQUES

System		GPS			Galileo		BeiDou	
<b>Signal</b>		L1 C/A	L1C	L5	E1	E5a	B1C	B2a
<b>Code length [ms]</b>		1	10	1	4	1	10	10
<b>Min. sampling frequency [MHz]</b>		4.092	8.184	40.920	8.184	40.920	8.184	40.920
<b>No compression</b>	<b>Data size [kbit]</b>	81.840	1636.800	818.400	654.720	818.400	1636.800	8184.000
	<b>Transfer time (20 kbps) [s]</b>	4.092	81.840	40.920	32.736	40.920	81.840	409.200
	<b>Transfer time (250 kbps) [s]</b>	0.327	6.547	3.274	2.619	3.274	6.547	32.736
<b>CS at 70%</b>	<b>Data size [kbit]</b>	24.552	491.040	245.520	196.416	245.520	491.040	2455.200
	<b>Transfer time (20 kbps) [s]</b>	1.228	24.552	12.276	9.821	12.276	24.552	122.760
	<b>Transfer time (250 kbps) [s]</b>	0.098	1.964	0.982	0.786	0.982	1.964	9.821

also influenced by the compression factor and the data length. Energy analysis shows that S-GPS can be ten times better in terms of energy consumption than a standalone GPS. The paper also compared themselves to the CO-GPS solution [84] and showed it could be twice more energy efficient.

The work [86] also provided a power consumption estimation for the communication link based on a state-of-the-art transceiver and considers the overall energy budget. The details on the data savings made by the compression are not provided by the authors, thus we have decided to provide some numbers based on the parameters from the experiments. In Table VI, we detail several signals and evaluate the amount of data required for each of them, comparing with and without compression. Based on the experimentation from [86], we assume 20 codes repetition to be offloaded, 1-bit quantization and real data (i.e., not complex I/Q). The data size  $D$  is computed as

$$D = C_n \cdot C_l \cdot f_s \cdot q_b, \tag{2}$$

where  $C_n$  is the number of codes recorded (i.e., 20),  $C_l$  is the code length (extracted from Table II,  $f_s$  is the sampling frequency,  $q_b$  is the bit quantization (i.e., 1 bit).

From the IEEE communication standards [78], we compare the time required to offload using different transfer rates. Transfer time  $T$  are computed as

$$T = \frac{D}{t_r}, \tag{3}$$

where  $t_r$  is the transfer rate. In this comparison, we have used 20 kbps and 250 kbps, as defined in [78].

The transfer time is significantly reduced by using CS, and it is clear that GPS L1 C/A is the one requiring the least data. Note that these numbers are provided as a broad comparison and should be refined. Further research is needed to understand the feasibility of offloading modern signals since this has not yet been reviewed in research to the author’s knowledge. It is foreseen that similarly to data bits in GPS L1 C/A signals, secondary codes should be accounted for, depending on the signals. Likewise, multi-level signals like E1 might require more than 1-bit quantization. Overheads from communication links and error correction also need to be considered.

These numbers can even be further reduced by reducing the length of the dataset or using an even larger compression rate.

Finally, [86] compares compression between S-GPS and the popular LZ77 algorithm used in ‘gzip’. As expected, compression performances for random binary streams are rather limited, leading to low performances for LZ77. S-GPS allows far greater compression capabilities. In Table VI, we also applied a 70% CS on the data size to provide a theoretical comparison with the no-compression case.

### C. Summary and Lessons Learned

Processing offloaded data fundamentally differs from what is performed onboard a “traditional” GNSS receiver. Instead of performing the signal processing stages locally, DSP stages are shifted to a remote server. We have introduced two concepts that are the enablers of such architecture, namely, the CTN, proposing a different approach to pseudorange computation; and the CS, for a drastic reduction of data to be uploaded. Computation complexity on the receiver side appears to be greatly reduced, leading theoretically to a reduction in energy consumption. Different implementations have been reviewed, resulting in data being offloaded at various stages before [84] or after [83] DSP operations. The main difference is the amount of data that needs to be uploaded to the server, as unprocessed raw data will take longer to upload but will require less processing on the receiver side.

Data transmission to the processing server remains the major challenge in this architecture. In this section, we have shown that the amount of data to be uploaded to a remote server is considerable and will require a proper wireless communication link with sufficient bandwidth to be uploaded in a reasonable time. A detailed review of the required bandwidth, transmission distance limit, or impact on the energy consumption of the wireless communication link could not be retrieved in any studies. Future research should concentrate on defining the requirements and limits for energy-efficient usage of the offloaded architecture, as it would prove important to decide in which scenarios such architecture can be applied.

Nevertheless, the offloaded architecture provides an interesting way of finding a receiver position while drastically reducing the onboard algorithmic complexity. Compared to the onboard processing algorithms reviewed in Section VII, the techniques used for off-loading are still quite novel, especially

the CS concept. Future research in this domain would benefit the offloaded architecture, which could be a game-changer for low-power GNSS processing, assuming that the challenges related to wireless communication links can be answered.

## IX. ON THE FUTURE OF LOW-POWER GNSS

During this work, we have dived into the topic of “low-power GNSS”. Our goal was to understand the factor that influences the energy consumption of a GNSS receiver, as the limiting factor in embedded devices today is the finite amount of energy available. To decide if a GNSS receiver should be embedded in a device, it needs to answer the positioning requirements of the application, but also consume an affordable amount of energy per position.

In this section, we recall the research questions (Q1-Q4) detailed in Section I, and link them to our review.

### A. The Factors Driving the Energy-Consumption (Q1)

In Section VI, we surveyed several papers to find out the factors influencing the energy consumption of a receiver. Some factors are unalterable, they do not come from the receiver design, but they will need to be accounted for. Those factors are the user’s requirements and the operating environment. Stringent positioning requirements and a difficult wireless environment will increase the difficulty to recover a signal, thus increasing the TTFF and the algorithm’s complexity. The rest of the factors are decided during the receiver design, again based on the user’s precision and energy consumption requirements. The hardware complexity, algorithm complexity, and activity time will be the final drivers of the energy consumption of a receiver.

Our analysis has compared different industrial receivers designed for high-precision or low-power applications. Power consumption and TTFF provided by manufacturers can provide an overview of how functionalities might influence energy consumption (e.g., number of channels, constellations, TTFF, etc.). However, to understand the influence of the factors mentioned above, we need more control over the processing chain. To the author’s knowledge, there is no open-source tool providing a controlled environment to explore the energy consumption of a receiver based on its hardware or software implementation. Future research aiming at reducing a receiver’s power/energy consumption should focus on this domain.

### B. Algorithms With the Lowest Complexity (Q2)

Our review mostly focused on the acquisition and tracking phases of the receiver. They are the first phases of the processing chain, dealing with a large amount of data and operating at high frequency, i.e., every 1-10 ms). They are expected to be the most energy-consuming phases of the software side. For onboard processing, many algorithms have been proposed over the years. In Section VII, the complexity of several methods has been reviewed, along with their applicability to modernized signals.

During the acquisition phase, FFT-based methods outperformed the search-based methods in terms of algorithmic time complexity. The new methods presented kept the same

principle and tried to reduce the complexity further. The QuickSync [44] and Modified XFAST [45] algorithms showed the best results in terms of time complexity reduction for GPS signal acquisition. Yet the Modified XFAST has significant degradation in SNR and could prove insufficient in a challenging wireless environment. The following methods concentrated on the modernized GNSS signals but failed to provide a time complexity analysis. Nevertheless, the Split FFT method proposes a reduction in operations number compared to PCPS.

During the tracking phase, BPSK modulated signal tracking seems to have reached its optimum efficiency, as no significant improvements could be found in recent papers. The most commonly used EPL technique appears to have the lowest complexity. Lowering the tracking frequency while keeping successful tracking seems to be the next challenge for lowering the tracking complexity of BPSK signals, as demonstrated theoretically in [13]. Alternatively, the research trend of the last decade has focused on improving BOC modulated signal tracking. This will be further reviewed in Section IX-C.

As exploring energy consumption at the algorithm level is not possible, we have decided to focus this review on algorithms’ complexity and find what is the lowest complexity algorithms at a theoretical level. Future research is needed to see how these algorithms would behave in real scenarios. A possible solution is by implementing a fully digital GNSS receiver to be used as a controlled environment for algorithm benchmarking [87], [88], [89].

### C. The Cost of Tracking Modern Signals (Q3)

As expected, tracking modern signals has proven to be more costly, in both resources and energy consumption. While they offer many benefits (detailed in Section V), these benefits come at the price of increased signal complexity. Quantifying the cost in algorithms complexity is not trivial and a relative comparison to legacy signal tracking is needed. In [21], the acquisition of modernized signals showed an increase in computation complexity compared to the GPS L1 C/A signal. The required sampling frequency and bit quantization, influenced by the signal modulation, directly affect the computations.

For the tracking phase, the difference is made between BPSK and BOC signals. Recalling Table II, many modern signals used the modulation BPSK, yet with higher code rates and secondary codes present in the signal. Compare to legacy BPSK signals, the tracking complexity is thus increased. Yet, legacy tracking techniques such as EPL can still be used.

With the introduction of the BOC modulation, the surveyed papers focused on developing efficient and unambiguous tracking algorithms for these signals. All the techniques presented required a higher number of correlators than the EPL technique, which was shown to be a good metric to assess the tracking complexity. The side-peaks cancellation techniques showed reduced complexity compared to sub-carrier tracking techniques when comparing the number of correlators needed. The BUT-EAS [65] and SRSA [66] shows relatively low complexity and good performances.

It should be noted that only a relatively small number of modern signals use BOC (i.e., L1C, E1 OS, E5, B1C), and

many opted for BPSK (see Table II). Further research focusing on improving BOC tracking for low-power receivers might be done for L1-only receivers. An alternative to this would be to focus on L5-only receivers, as all signals across constellations on this frequency share the structure and use BPSK modulation. Examples of such implementation recently started to appear at the industry level [90].

#### D. Offloaded or Onboard Processing? (Q4)

In offloaded processing, transforming a GNSS receiver into an RF logger ensures the lowest consumption a GNSS receiver can ever get. The “atomic” level of TOA positioning is reached with the CTN technique. Delay-tolerant applications, with severe energy constraints and lower requirements over positioning quality, are the targeted applications. This is the case for many IoT devices. Yet, it quickly becomes inefficient for embedded devices with more constraints on their positioning quality.

The communication link requirements limit its application to connected environments only. In most of the studies found, the consequences in terms of energy consumption have not been theoretically nor experimentally reviewed. Data transmission will necessarily take a significant amount of energy and will be impacted by the application environment. Modelling of the communication links is lacking in the current state-of-the-art research in GNSS offloaded processing. Future research should define the impacts of scenarios and user constraints on the communication link, as well as assess its energy consumption. This will be crucial in understanding if the trade-off between onboard and offloaded architectures will result in energy savings for the receiver.

Offloading computations align with the concepts of Edge computing, a trending research topic in recent years for IoT applications [91]. In an Edge architecture, data is not transferred to the Cloud but processed at an intermediary node. It saves energy by reducing data movement while making the processing closer to the user for increased privacy and security. To the author’s knowledge, the derivation of offloaded GNSS processing into an Edge scenario has not been reviewed yet. Edge computing has also been reviewed in the scope of Approximate Computing (AxC) techniques [92], which could also prove applicable in the offloaded GNSS processing case.

## X. CONCLUSION

This survey reviewed state-of-the-art low-power GNSS techniques. The driving factors in power and energy consumption have been listed and detailed. We also reviewed modernized GNSS signals and their benefits for low-power positioning. While their increased signal complexity makes them counter-intuitive for low-power applications, several benefits of interference and multipath mitigation would make them suitable candidates for harsh signal environments. Yet a concrete assessment of their impacts on the energy budget over the whole positioning chain has not been found in any paper reviewed.

The two most promising architectures for low-power positioning have been reviewed: one where the positioning is

onboard, fully performed locally; the second one where the processing is outsourced to a server and the receiver becomes an RF logger only. The advantages and disadvantages of both systems have been reviewed, and the gaps in the current state-of-the-art pointed out. Current research in onboard processing showed keen interest in modern GNSS signals. We summarised methods with computation complexity estimation and tried identifying the least complex algorithms for each processing stage. However, the vast amount of proposed methods and very heterogeneous analysis prevent direct comparison of the performance of the methods. Offloaded processing showed a completely different positioning architecture that could enable lower energy consumption in embedded devices. The main challenge resides in communication link consumption, which effects should be further reviewed in the energy consumption budget. Developments in communication links dedicated to IoT will probably offer low-energy solutions for the transmission of the data to the remote server, yet given the large bandwidth required, their applicability is yet to be assessed.

A lack of a shared benchmark environment has been identified in both architectures. In-house development and data acquisition prevent direct comparison of the performance of the methods since the code and datasets are not shared with the article. The development of an open-source benchmarking platform would be a suitable solution, enabling the comparability of algorithms and identifying high-complexity tasks in the processing chain.

#### LIST OF ACRONYMS

A-GNSS	Assisted GNSS
ACF	Auto-Correlation Function
ALM	Adaptive Logic Module
ASPeCT	Auto-Correlation Side-Peak Cancellation Technique
AxC	Approximate Computing
BeiDou	GNSS from China
BJ	Bump-Jumping
BOC	Binary Offset Carrier
BPSK	Binary Phase Shift Keying
BUT-EAS	BOC Unambiguous Tracking using Extended Auxiliary Symbols
C/A	Coarse/Acquisition Code
CAS	Commercial Authentication Service
CCF	Cross-Correlation Function
CDMA	Code Division Multiple Access
CHIMERA	Chip-Message Robust Authentication
CMOS	Complementary Metal-Oxide Semiconductor
CRC	Cyclic Redundancy Check
CS	Commercial Service
CS	Compressive Sensing
CTN	Coarse-Time Navigation
DBZP	Double Block Zero-Padding
DBZPTI	Double Block Zero-Padding Transition Insensitive
DDPE	Decimation Double Phase Estimator
DE	Double Estimator
DGNSS	Differential GNSS



DLL	Delay Lock Loop	SPAR	Symmetrical Pulse Ambiguity Removing
DNF	Data Not Found	SPLL	Sub-carrier Phase Lock Loop
DPE	Double Phase Estimator	SPP	Standard Point Positioning
DSP	Digital Signal Processing	SRSA	Sub-function Reconstruction Synchronization Algorithm
DSSS	Direct Sequence Spread Spectrum	SS	Serial Search
EKF	Extended Kalman Filter	TOA	Time of Arrival
EPL	Early-Prompt-Late	TOW	Time of Week
EU	European Union	TTF	Time To First Fix
FDMA	Frequency Division Multiple Access	USA	United States of America
FEC	Forward Error Correction	USSR	Union of Soviet Socialist Republics
FFT	Fast Fourier Transform	VE	Very-Early
FLL	Frequency Lock Loop	VL	Very-Late
FOC	Full Operational Capacity	XFAST	Extended Replica Folding Acquisition Search Technique
FPGA	Field-Programmable Gate Array		
Galileo	GNSS from European Union		
GLONASS	Globalnaya Navigazionnaya Sputnikovaya Sistema (GNSS from Russia)		
GNSS	Global Navigation Satellite Systems		
GPS	Global Positioning System from USA		
GRASS	General Removing Ambiguity via Sidepeak Suppression		
HAS	High Accuracy Service		
ICD	Interface Control Document		
IEEE	Institute of Electrical and Electronics Engineers		
IF	Intermediate Frequency		
IGRASS	Improved General Removing Ambiguity via Sidepeak Suppression		
IMU	Inertial Measurement Unit		
IoT	Internet of Things		
LEAP	Low Energy-Assisted Positioning		
LR-WPAN	Low-Rate Wireless Personal Area Network		
LTE	Long-Term Evolution		
LTE-M	Long Term Evolution (4G), category M1		
MCCF	Multipled Cross-Correlation Function		
MEDLL	Multipath Estimating Delay Lock Loop		
NMA	Navigation Message Authentication		
OS	Open Service		
OSNMA	Open Service Navigation Message Authentication		
PCM	Partial Correlation Method		
PCPS	Parallel Code Phase Search		
PLL	Phase Lock Loop		
PPP	Precise Point Positioning		
PRN	Pseudo Random Noise		
PRS	Public Regulated Service		
PSD	Power Spectral Density		
PSK	Phase Shift Keying		
PUDLL	Pseudo Correlation Function-based Unambiguous Delay Lock Loop		
RADAR	Radio Detection and Ranging		
RF	Radio Frequency		
RFFE	RF Front-End		
SBZP	Single Block Zero-Padding		
SCA	Spreading Code Authentication		
SD	Secure Digital		
SLL	Sub-carrier Lock Loop		
SNR	Signal-to-Noise Ratio		
SoC	System-on-Chip		

## APPENDIX

## MATHEMATICAL DEFINITION OF COMPRESSIVE SENSING

As quantization and time length are already at their minimum possible value, the only factor still left is the sampling frequency. Over the last two decades, significant progress in signal reconstruction has led to breakthrough signal compression, pushing sampling frequency to a level lower than the Nyquist rate.

Compressive Sensing (CS) is a signal processing technique allowing signal reconstruction based on very few measurement samples, much fewer than required by the Nyquist rate. It has been first described in [81] and explained quite extensively in [82]. The following summary is a quick mathematical introduction to the technique, the reader is invited to review these references for more details.

According to CS theory, any “natural” signal is compressible, meaning that it can be represented by a sparse vector on a certain transform basis. Such a vector contains mostly zeros and is called  $K$ -sparse on this basis if there are exactly  $K$  nonzero elements. Let  $x \in \mathbb{R}^n$  be the original signal received, and  $s \in \mathbb{R}^n$  be its  $K$ -sparse representation in the basis  $\Psi$ . Given that “ $\cdot$ ” represent a matrix multiplication, their relation is represented by

$$s = \Psi \cdot x. \quad (4)$$

The design of  $\Psi$  is at the core of the CS theory. This basis can be viewed as a *dictionary*, containing simple pieces that, added together, can represent a complex signal. Fourier or Wavelet bases are generic  $\Psi$  bases used in many applications (e.g., image reconstruction).

Now going into the compression stage, if  $x$  can be represented as  $K$ -sparse in  $\Psi$ , then only a limited number of measurements is required to represent this signal, leading to a sub-sampling opportunity. Let  $y \in \mathbb{R}^m$  be the compressed (sub-sampled) version of  $x$ , and  $C \in \mathbb{R}^{n \times m}$  be the *measurement* matrix (a.k.a. sensing matrix).  $x$  and  $y$  are related to each other via  $C$  as expressed as

$$y = C \cdot x. \quad (5)$$

The  $C$  matrix needs to answer two conditions to be considered valid: (1)  $C$  must be incoherent to the *sparsifying*

basis  $\Psi$ ; (2) the number of measurements should be sufficiently large. The first condition can be met by using a random matrix as a measurement matrix, for example, a Gaussian or Bernoulli distributed random matrix. The second condition is met using the equation described in [82], which evaluates the number of samples required. It can also be assessed empirically. Further details on these conditions and their importance can be found in [82].

Given a sparse vector randomly sampled and transmitted to the remote computer, the original signal should now be reconstructed. The complexity of the problem resides in this part of the algorithm since, given a compressed measurement vector  $y$ , an infinite number of sparse vectors  $s$  can be constructed. CS theory actually tells us to find the *sparsest* vector  $s$  given the measurements  $y$ , to successfully reconstruct the signal  $x$ . The following equations define the optimization problem as

$$y = C \cdot \Psi \cdot s = \Theta \cdot s, \quad (6)$$

$$\hat{s} = \underset{s}{\operatorname{argmin}} \|s\|_1 \text{ subject to } y = C \cdot \Psi \cdot s. \quad (7)$$

We remark that this is a convex  $l_1$ -norm minimization problem, given  $\|\cdot\|_1$ . Originally the minimization should be over the  $l_0$  pseudo-norm, leading to a non-convex optimization, thus only solvable by combination test. It can be simplified to  $l_1$  thanks to the conditions set on the measurement matrix  $C$ .

In summary, the only computation requirement on the logger side is sub-sampling the received data with a random measurement matrix  $C$ . On the remote computer side, it should proceed with the minimization solving, which requires the measurement matrix  $C$  as well, given equation 6. Therefore, on top of the compressed measurement  $y$ , the logger also needs to send  $C$  or the seed to generate it on the remote side.

## REFERENCES

- [1] N. Spens, D.-K. Lee, F. Nedelkov, and D. Akos, "Detecting GNSS jamming and spoofing on Android devices," *NAVIGATION J. Inst. Navig.*, vol. 69, no. 3, p. 537, 2022. [Online]. Available: <https://navi.ion.org/content/69/3/navi.537>
- [2] L. Huang and Q. Yang, "Low-cost GPS simulator—GPS spoofing by SDR," in *Proc. DEFCON 23*, Aug. 2015, p. 109.
- [3] J. Song et al., "GNSS spoofing identification and smoothing localization method for GNSS/visual SLAM system," *Appl. Sci.*, vol. 12, no. 3, p. 1386, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/3/1386>
- [4] B. Aebischer and L. M. Hilty, *The Energy Demand of ICT: A Historical Perspective and Current Methodological Challenges*, vol. 310. Cham, Switzerland: Springer, Aug. 2015, pp. 71–103.
- [5] "Power-efficient positioning for the Internet of Things," Eur. GNSS Agency (GSA), Prague, Czechia, White Paper, 2020. [Online]. Available: <https://www.euspa.europa.eu/newsroom/news/power-efficient-positioning-iot>
- [6] S. Narayana, R. V. Prasad, V. Rao, L. Mottola, and T. V. Prabhakar, "Hummingbird: Energy efficient GPS receiver for small satellites," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, 2020, pp. 1–13. [Online]. Available: <https://doi.org/10.1145/3372224.3380886>
- [7] G. Rivela et al., "A low power RF front-end for L1/E1 GPS/Galileo and GLONASS signals in CMOS 65nm technology," in *Proc. Int. Conf. Localization GNSS (ICL-GNSS)*, 2011, pp. 7–12.
- [8] C. G. Tan et al., "A universal GNSS (GPS/Galileo/Glonass/Beidou) SoC with a 0.25mm<sup>2</sup> radio in 40nm CMOS," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2013, pp. 334–335.
- [9] C.-W. Cheng and Y.-J. E. Chen, "A 90-nm CMOS multi-standard GNSS receiver front-end," in *Proc. IEEE 14th Topical Meeting Silicon Monolithic Integr. Circuits Rf Syst.*, 2014, pp. 53–55.
- [10] T. Borejko et al., "NaviSoC: High-accuracy low-power GNSS SoC with an integrated application processor," *Sensors*, vol. 20, no. 4, p. 1069, 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/4/1069>
- [11] E. S. Lohan, D. A. de Diego, J. A. López-Salcedo, G. Seco-Granados, P. Boto, and P. Fernandes, "Unambiguous techniques modernized GNSS signals: Surveying the solutions," *IEEE Signal Process. Mag.*, vol. 34, no. 5, pp. 38–52, Sep. 2017.
- [12] W. Feneniche, K. Rouabah, M. Flissi, S. Atia, S. E. Mezaache, and S. Meguellati, "Unambiguous method for DLL BOC signals tracking," *Int. J. Commun. Syst.*, vol. 33, no. 9, p. e4374, 2020.
- [13] V. Bellad, "Intermittent GNSS signal tracking for improved receiver power performance," Ph.D. dissertation, Dept. Doctor Philos., Univ. Calgary, Calgary, AB, Canada, 2015. [Online]. Available: <https://prism.ualgary.ca/handle/11023/2667>
- [14] "Low-power GNSS for tracking applications." Ublox. 2021. [Online]. Available: <https://www.u-blox.com/en/publication/white-paper/low-power-gps-tracking-applications>
- [15] K. Chen, G. Tan, and M. Lu, "Improving the energy performance of GPS receivers for location tracking applications," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2017, pp. 85–90.
- [16] J. Subirana, J. Zornoza, and M. Hernández-Pajares, *GNSS Data Processing Volume I: Fundamentals and Algorithms*. ESA Commun., Paris, France, 2013.
- [17] J. Nurmi, E.-S. Lohan, S. Sand, and H. Hurskainen, Eds., *GALILEO Positioning Technology* (Signals and Communication Technology). Dordrecht, The Netherlands: Springer, 2014.
- [18] E. D. Kaplan and C. J. Hegarty, *Understanding GPS: Principles and Applications*. 3rd ed. Norwood, MA, USA: Artech House, 2017.
- [19] "Russia adds to GLONASS constellation with latest launch." GPS World. 2020. Accessed: Jun. 14, 2022. [Online]. Available: <https://www.gpsworld.com/russia-adds-to-glonass-constellation-with-latest-launch/>
- [20] M.-A. Fortin, F. Bourdeau, and R. J. Landry, "Implementation strategies for a software-compensated FFT-based generic acquisition architecture with minimal FPGA resources," *NAVIGATION*, vol. 62, no. 3, pp. 171–188, 2015. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/navi.110>
- [21] J. Leclère, R. Landry, and C. Botteron, "Comparison of L1 and L5 bands GNSS signals acquisition," *Sensors*, vol. 18, p. 2779, Aug. 2018.
- [22] "IS-GPS-200M / 705H / 800H." GPS. Accessed: Apr. 6, 2023. [Online]. Available: <https://www.gps.gov/technical/icwg/>
- [23] "Open service signal-in-space interface control document." Galileo. Accessed: Apr. 6, 2023. [Online]. Available: <https://www.gsc-europa.eu/electronic-library/programme-reference-documents>
- [24] "BeiDou signal-in-space interface control document open service." BeiDou. Accessed: Apr. 6, 2023. [Online]. Available: <http://en.beidou.gov.cn/SYSTEMS/ICD/>
- [25] National Academies of Sciences Engineering and Medicine, *Handbook of Frequency Allocations and Spectrum Protection for Scientific Uses*, 2nd ed. Washington, DC, USA: Nat. Acad. Press, 2015.
- [26] "New civil signals." GPS. Accessed: Apr. 6, 2023. [Online]. Available: <https://www.gps.gov/systems/gps/modernization/civilsignals/>
- [27] "China launches final satellite in GPS-like BeiDou system." PhysOrg. 2020. Accessed: Jun. 14, 2022. [Online]. Available: <https://phys.org/news/2020-06-china-satellite-gps-like-beidou.html>
- [28] "Galileo constellation grows with launch of 2 new satellites." GPS World. 2021. Accessed: Jun. 14, 2022. [Online]. Available: <https://www.gpsworld.com/galileo-constellation-grows-with-the-launch-of-two-new-satellites/>
- [29] J. W. Betz, "Binary offset carrier modulations for radionavigation," *NAVIGATION*, vol. 48, no. 4, pp. 227–246, 2001. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2161-4296.2001.tb00247.x>
- [30] F. Macchi, "Development and testing of an L1 combined GPS-Galileo software receiver," Ph.D. dissertation, Dept. Doctor Philos., Univ. Calgary, Calgary, AB, USA, 2010.
- [31] D. Schmidt, K. Radke, S. Camtepe, E. Foo, and M. Ren, "A survey and analysis of the GNSS spoofing threat and countermeasures," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 1–31, May 2016. [Online]. Available: <https://doi.org/10.1145/2897166>
- [32] A. Rustamov, N. Gogoi, A. Minetto, and F. Dovis, "Assessment of the vulnerability to spoofing attacks of GNSS receivers integrated in consumer devices," in *Proc. Int. Conf. Localization GNSS (ICL-GNSS)*, 2020, pp. 1–6.
- [33] B. Motella, M. Nicola, and S. Damy, "Enhanced GNSS authentication based on the joint CHIMERA/OSNMA scheme," *IEEE Access*, vol. 9, pp. 121570–121582, 2021.

- [34] I. Fernandez-Hernandez et al., "Galileo high accuracy service: Initial definition and performance," *GPS Solut.*, vol. 26, p. 65, Jul. 2022.
- [35] F.-A. Fortin, S. Grenier, and M. Parizeau, "Generalizing the improved run-time complexity algorithm for non-dominated sorting," in *Proc. 15th Annu. Conf. Genet. Evol. Comput.*, 2013, pp. 615–622. [Online]. Available: <https://doi.org/10.1145/2463372.2463454>
- [36] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.
- [37] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*. Cambridge, U.K.: Cambridge Univ. Press, 2009. [Online]. Available: <https://books.google.fi/books?id=8WJqvsoob48MC>
- [38] "Novatel PIM222A product sheet." Novatel. 2021. [Online]. Available: <https://novatel.com/products/receivers/gnss-gps-receiver-boards/pim222a>
- [39] "ZED-F9H module product summary." Ublox. 2022. [Online]. Available: <https://www.u-blox.com/en/docs/UBX-19026707>
- [40] "ZOE-M8B module product summary." Ublox. 2022. [Online]. Available: <https://www.u-blox.com/en/docs/UBX-17012173>
- [41] "Third-generation L1/L5 chip from broadcom for mobiles, wearables has multipath mitigation." InsideGNSS. 2021. Accessed: Apr. 6, 2023. [Online]. Available: <https://insidengss.com/third-generation-l1-l5-chip-from-broadcom-for-mobiles-wearables-has-multipath-mitigation/>
- [42] "UFirebird-UC6226 datasheet," Data Sheet, Unicorecomm, Beijing, China, 2022. [Online]. Available: [https://en.unicorecomm.com/assets/upload/file/UC6226\\_Datasheet\\_EN\\_R3\\_7.pdf](https://en.unicorecomm.com/assets/upload/file/UC6226_Datasheet_EN_R3_7.pdf)
- [43] F. Van Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS*. Boston, MA, USA: Artech House, 2009.
- [44] H. Hassanieh, F. Adib, D. Katabi, and P. Indyk, "Faster GPS via the sparse Fourier transform," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 353–364. [Online]. Available: <https://doi.org/10.1145/2348543.2348587>
- [45] M. Tahir, L. Lo Presti, and M. Fantino, "Folding techniques for low complexity acquisition of modern GNSS signals," in *Proc. Inst. Navig. Int. Tech. Meeting (ITM)*, vol. 1, Jan. 2012, pp. 269–310.
- [46] F. Benedetto, G. Giunta, E. S. Lohan, and M. Renfors, "A fast unambiguous acquisition algorithm for BOC-modulated signals," *IEEE Trans. Veh. Technol.*, vol. 62, no. 3, pp. 1350–1355, Mar. 2013.
- [47] J. Leclère, C. Botteron, and P.-A. Farine, "Acquisition of modern GNSS signals using a modified parallel code-phase search architecture," *Signal Process.*, vol. 95, pp. 177–191, Feb. 2014.
- [48] J. Svatoň, F. Vejražka, P. Kubalík, J. Schmidt, and J. Borecký, "Novel partial correlation method algorithm for acquisition of GNSS tiered signals," *NAVIGATION*, vol. 67, no. 4, pp. 745–762, 2020. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/navi.390>
- [49] D. J. R. van Nee and A. J. R. M. Coenen, "New fast GPS code-acquisition technique using FFT," *Electron. Lett.*, vol. 27, pp. 158–160, Jan. 1991.
- [50] Y. Zhang, M. Wang, and Y. Li, "Low computational signal acquisition for GNSS receivers using a resampling strategy and variable circular correlation time," *Sensors*, vol. 18, no. 2, p. 678, 2018. [Online]. Available: <https://www.mdpi.com/1424-8220/18/2/678>
- [51] M. Foucras, O. Julien, C. Macabiau, and B. Ekambi, "A novel computationally efficient Galileo E1 OS acquisition method for GNSS software receiver," in *Proc. 25th Int. Tech. Meeting Satell. Division Inst. Navig. (ION GNSS)*, vol. 1, Sep. 2012, pp. 357–364.
- [52] J. Leclère, C. Botteron, R. J. Landry, and P.-A. Farine, "FFT splitting for improved FPGA-based acquisition of GNSS signals," *Int. J. Navig. Observ.*, vol. 2015, p. 12, Dec. 2015.
- [53] C. Yang, J. Chaffee, J. Abel, and J. Vasquez, "Extended replica folding for direct acquisition of GPS P-code and its performance analysis," in *Proc. 13th Int. Tech. Meeting Satell. Division Inst. Navig. (ION GPS)*, Jan. 2000, pp. 2070–2078.
- [54] E. S. Lohan, A. Burian, and M. Renfors, "Low-complexity unambiguous acquisition methods for BOC-modulated CDMA signals," *Int. J. Satell. Commun. Netw.*, vol. 26, no. 6, pp. 503–522, 2008. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.922>
- [55] R. D. J. van Nee, J. Sierveld, P. C. Fenton, and B. R. Townsend, "The multipath estimating delay lock loop: Approaching theoretical accuracy limits," in *Proc. IEEE Position Location Navig. Symp. (PLANS)*, 1994, pp. 246–251.
- [56] N. Martin, V. Leblond, G. Guillotel, and V. Heiries, "BOC<sub>(x,y)</sub> signal acquisition techniques and performances," in *Proc. ION GPS/GNSS*, Jan. 2003, pp. 188–198.
- [57] P. Fine and W. Wilson, "Tracking algorithm for GPS offset carrier signals," in *Proc. Nat. Tech. Meeting Inst. Navig.*, 1999, pp. 671–676.
- [58] J. Qi, J. Chen, Z. Li, and D. Zhang, "Unambiguous BOC modulated signals synchronization technique," *IEEE Commun. Lett.*, vol. 16, no. 7, pp. 986–989, Jul. 2012.
- [59] O. Julien, C. Macabiau, M. E. Cannon, and G. Lachapelle, "ASPeCT: Unambiguous sine-BOC<sub>(n,n)</sub> acquisition/tracking technique for navigation applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 1, pp. 150–162, Jan. 2007.
- [60] Z. Yao, M. Lu, and Z. Feng, "Unambiguous sine-phased binary offset carrier modulated signal acquisition technique," *IEEE Trans. Wireless Commun.*, vol. 9, no. 2, pp. 577–580, Feb. 2010.
- [61] Z. Yao, X. Cui, M. Lu, Z. Feng, and J. Yang, "Pseudo-correlation-function-based unambiguous tracking technique for sine-BOC signals," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 4, pp. 1782–1796, Oct. 2010.
- [62] F. Shen, G. Xu, J. W. Cheong, and H. Feng, "Unambiguous acquisition and tracking technique for general BOC signals," *Radioengineering*, vol. 24, pp. 840–849, Sep. 2015.
- [63] T. Yan, J. Wei, Z. Tang, B. Qu, and Z. Zhou, "Unambiguous acquisition/tracking technique for high-order sine-phased binary offset carrier modulated signal," *Wireless Pers. Commun.*, vol. 84, pp. 2835–2857, May 2015.
- [64] T. Li, Z. Tang, J. Wei, Z. Zhou, and B. Wang, "An unambiguous tracking technique for cosine-phased BOC signals with low complexity," *Radio Eng.*, vol. 27, pp. 1191–1198, Sep. 2018.
- [65] B. Wang, T. Li, J. Wei, and Z. Tang, "A new unambiguous tracking algorithm for sine-BOC<sub>(m,n)</sub> signals," *GPS Solut.*, vol. 23, Apr. 2019, pp. 1–20.
- [66] F. Hao et al., "Unambiguous acquisition/tracking technique based on sub-correlation functions for GNSS sine-BOC signals," *Sensors*, vol. 20, no. 2, p. 485, 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/2/485>
- [67] F. M. Schubert, J. Wendel, M. Söllner, M. Kaindl, and R. Kohl, "The astrium correlation: Unambiguous tracking of high-rate BOC signals," in *Proc. IEEE/ION Position Location Navig. Symp. (PLANS)*, 2014, pp. 589–601.
- [68] S. Hodgart and P. Blunt, "Dual estimate receiver of binary offset carrier modulated signals for global navigation satellite systems," *Electron. Lett.*, vol. 43, pp. 877–878, Feb. 2007.
- [69] D. Borio, "Double phase estimator: New unambiguous binary offset carrier tracking algorithm," *IET Radar Sonar Navig.*, vol. 8, no. 7, pp. 729–741, 2014. [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-rsn.2013.0306>
- [70] T. Feng, "Decimation double-phase estimator: An efficient and unambiguous high-order binary offset carrier tracking algorithm," *IEEE Signal Process. Lett.*, vol. 23, no. 7, pp. 905–909, Jul. 2016.
- [71] B. R. Townsend, P. C. Fenton, K. J. Van Dierendonck, and D. J. R. Van Nee, "Performance evaluation of the multipath estimating delay lock loop," *NAVIGATION*, vol. 42, no. 3, pp. 502–514, 1995. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2161-4296.1995.tb01903.x>
- [72] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, vol. 42. Cambridge, MA, USA: MIT Press, Jan. 2001.
- [73] R. Morales-Ferre, G. Seco-Granados, and E. S. Lohan, "Energy efficiency considerations for GNSS signal acquisition," in *Proc. InsideGNSS*, Nov. 2021, pp. 32–39.
- [74] M. Asim, Y. Wang, K. Wang, and P.-Q. Huang, "A review on computational intelligence techniques in cloud and edge computing," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 6, pp. 742–763, Dec. 2020.
- [75] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, *A Software-Defined GPS and Galileo Receiver, a Single Frequency Approach*. Boston, MA, USA: Birkhäuser, 2007.
- [76] K. W. Payne, "Development of two new GPS avian tracking systems," Ph.D. dissertation, Dept. Doctor Philos., Univ. Otago, Dunedin, New Zealand, 2015. [Online]. Available: <http://hdl.handle.net/10523/5904>
- [77] T. C. A. Molteno, "Estimating position from millisecond samples of GPS signals (the 'FastFix' algorithm)," *Sensors*, vol. 20, no. 22, p. 6480, 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/22/6480>

- [78] C. Powell, *IEEE Standard for Low-Rate Wireless Networks*, IEEE Standard 802.15.4-2020, 2020. [Online]. Available: <https://standards.ieee.org/ieee/802.15.4/7029/>
- [79] V. Lucas-Sabola, G. Seco-Granados, J. López-Salcedo, and J. A. García-Molina, "GNSS IoT positioning: From conventional sensors to a cloud-based solution," in *Proc. InsideGNSS*, 2018, pp. 1–10.
- [80] M. Wang, H. Qin, and T. Jin, "Massive terminal positioning system with snapshot positioning technique," *GPS Solut.*, vol. 23, no. 2, p. 31, Jan. 2019. [Online]. Available: <https://doi.org/10.1007/s10291-018-0821-z>
- [81] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
- [82] S. L. Brunton and J. N. Kutz, *Data Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
- [83] H. S. Ramos, T. Zhang, J. Liu, N. B. Priyantha, and A. Kansal, "LEAP: A low energy assisted GPS for trajectory-based services," in *Proc. ACM Conf. Ubiquitous Comput.*, Sep. 2011, pp. 335–344.
- [84] J. Liu, B. Priyantha, T. Hart, H. S. Ramos, A. A. F. Loureiro, and Q. Wang, "Energy efficient GPS sensing with cloud offloading," in *Proc. 10th ACM Conf. Embedded Netw. Sens. Syst.*, 2012, pp. 85–98. [Online]. Available: <https://doi.org/10.1145/2426656.2426666>
- [85] "Product summary: CloudLocate." Ublox. 2021. [Online]. Available: <https://www.u-blox.com/en/product/cloudlocate>
- [86] P. Misra et al., "Energy efficient GPS acquisition with sparse-GPS," in *Proc. 13th Int. Symp. Inf. Process. Sens. Netw. (IPSN)*, 2014, pp. 155–166.
- [87] C. Fernández-Prades, J. Arribas, P. Closas, C. Avilés, and L. Esteve, "GNSS-SDR: An open source tool for researchers and developers," in *Proc. 24th Int. Tech. Meeting Sat. Div. Inst. Navig.*, Portland, OR, USA, Sep. 2011, pp. 780–794.
- [88] "The FGI-GSRx software defined GNSS receiver goes open source." Finnish Geodetic Institute. Accessed: Apr. 6, 2023. [Online]. Available: [https://www.maanmittauslaitos.fi/en/topical\\_issues/fgi-gsrx-software-defined-gnss-receiver-goes-open-source](https://www.maanmittauslaitos.fi/en/topical_issues/fgi-gsrx-software-defined-gnss-receiver-goes-open-source)
- [89] A. Grenier, E. S. Lohan, A. Ometov, and J. Nurmi, "An open-source software-defined receiver for GNSS algorithms benchmarking," in *Proc. 14th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, 2022, pp. 31–38.
- [90] "A pure L5 mobile receiver." InsideGNSS. 2020. Accessed: Apr. 6, 2023. [Online]. Available: <https://insidegnss.com/a-pure-l5-mobile-receiver/>
- [91] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in industrial Internet of Things: Architecture, advances and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2462–2488, 4th Quart., 2020.
- [92] H. J. Damsgaard, A. Ometov, and J. Nurmi, "Approximation opportunities in edge computing hardware: A systematic literature review," *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–49, 2022.