Implementation of an Open-Source Software Suite for the Galileo High Accuracy Service

Oliver Horst

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 31.12.2021

Supervisor

Prof. D.Sc. Jaan Praks

Advisor

Lic.Sc. (Tech.) Tuomo Malkamäki

Aalto University
School of Electrical Engineering

Aalto University
School of Electrical Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Oliver Horst

Title Implementation of an Open-Source Software Suite for the Galileo High Accuracy Service

Degree programme ICT Innovation

Major Autonomous Systems **Code of major** ELEC3055

Supervisor Prof. D.Sc. Jaan Praks

Advisor Lic.Sc. (Tech.) Tuomo Malkamäki

Date 31.12.2021 **Number of pages** 49+4 **Language** English

Abstract

Global Navigation Satellite Systems (GNSS) have already become an integral part of today's technology-driven society. The emerging of new technologies, requiring highly accurate positioning services, has given rise to multiple providers offering augmentations to the freely available navigation solutions. With the conversion of the Galileo Commercial Service (CS) into the High Accuracy Service (HAS), the European Galileo programme is to offer an alternative to these costly providers, by making a decimeter-accurate positioning service for free to users worldwide. However, being a new service, the initial impact on the GNSS community can be limited.

This thesis concentrates on ways of promoting this service and increasing its accessibility. To do so, a standalone Python library is developed to bring the encoded transmission data into a format known to commonly used Precise Point Positioning (PPP) tools. In addition, such a PPP tool is adapted to process the data to offer an easy-to-use solution to assess the capabilities of the HAS. Both software solutions together are used to evaluate the HAS in its current development status and reveal a greatly improved performance over standard solutions. Lastly, a proposal for a new HAS message format is made with the aim of ensuring the quality of the service in the long run. Yet, as the evaluation revealed performance issues in a few aspects, some further investigation is advised.

Keywords GNSS, Galileo, HAS, High Accuracy Service, PPP, Precise Point Positioning, GNSS Augmentation

Preface

This thesis would not have been possible without the support received over the course of working on it.

First, I want to express my gratitude to my supervisor D.Sc. Jaan Praks for giving me the opportunity to work on this topic and for giving constructive feedback throughout the process.

Next, I would like to thank my advisor Tuomo Malkamäki for his guidance and open ear during the last months. However, I do not want to leave out my unofficial advisor and guide to the field of GNSS, D.Sc. Martti Kirkko-Jaakkola. His recommendations and advice proved vital more than a few times, helping me out to a great extent. I also want to extend this gratitude to the whole FGI Navigation group, who took me into their midst and assisted me not only in topical but also organisational matters.

In addition, I wish to thank my family for the continuous support before and during this period in my life and in the years to come. Thank you for making me who I am.

Last, but not least, I want to thank Emily Hien Vu. Your constant motivation and support made me go past what I imagined and provided me with a pillar I could count on.

Otaniemi, 31.12.2021

Contents

Symbols and Abbreviations

Symbols

γ	Parity vector
$\mathbf{\Gamma}$	Code vector
$\mathcal{T}$	Observed ranges between receiver and satellite
ρ	Frequency-independent terms of the ranging calculations
$\mathbf{c}$	Message vector
c	Speed of light, 299,792,458 m/s
f	Frequency
$\mathbf{G}$	Generator matrix
$\mathbf{I}$	Identity matrix
t	Time

Abbreviations

AR	Ambiguity Resolution
C/A	Coarse Acquisition (the legacy GPS signal)
CRC	Cyclic Redundancy Check
CS	Commercial Service (planned as part of the Galileo project)
CSSR	Compact SSR
DF	Dual Frequency
DGNSS	Differential Global Navigation Satellite System
DGPS	Differential GPS
GNSS	Global Navigation Satellite System
GSA	European GNSS Agency
HAS	High Accuracy Service (formerly CS)
IGS	International GNSS Society
MT	Message Type
NRTK	Network Real Time Kinematic
OSR	Observation Space Representation
PPP	Precise Point Positioning
PRN	Pseudorandom Noise (PRN) Code
RTCM	Radio Technical Commission for Maritime Services
RTK	Real Time Kinematic
S/A	Selective Availability
SF	Single Frequency
SIS	Signal-In-Space
SSR	State Space Representation
TEC	Total Electron Content
QZSS	Quasi-Zenith Satellite System

1 Introduction

For about 40 years, satellite navigation has been an integral part of navigation worldwide [1] and the upcoming of high accuracy services further increases its usage. The ability to accurately tell your position has opened the door to many new applications and immensely increased the potential of numerous others. Along with the user base, the accessible accuracy increased as much as the areas of usage. It has been a long time since the main concern was for airplanes to avoid potentially dangerous airspace by a matter of positioning within dozens of meters. Instead, the satellites of today enable precise, automated agriculture as well as an automated rendezvous of space vehicles in orbit [2], [3]. As such, the positioning market is undergoing a shift in recent years, with techniques of positioning within a decimeter or even centimeter range becoming more available. At the same time, areas of application requiring such an accuracy, like robotics or drones, are developing and growing.

In order to support this trend, the European Commission and the Galileo project have announced the launch of a High Accuracy Service (HAS). It will provide users with a positioning service in the decimeter range. While similar services have been commercially available before, it was decided to provide this service globally and for free [4]. This is intended to promote innovation in fields that require such data within the European Union. In its early stages, the HAS has been received well among the stakeholder community [5].

While it can be expected that the HAS will be a major asset to multiple fields, the adaptation of such a new service is often dependent on its accessibility and usability. This is in particular true for a service, which is intended not to be used by users from the same area of work, but by those who simply intend to use the data in their own domains. Especially due to other services being already established in the market, the HAS will need to be convincing in all aspects in order to reach the users. The goal of this thesis will thus be to *increase the HAS accessibility and make the service more easy to use from the moment it launches for the general public.*

To do this, the thesis will analyse existing conventions in the domain and compare them against the specifications of the HAS. It is planned that a decoding- and converting software will be implemented in order to transform HAS data into more common, already used data formats. Additionally, available software solutions capable of using the contained data to perform high accuracy positioning will be identified and adapted if necessary. Together, this work aims to provide users with a software suite enabling a simple use of the new service without substantial modification to existing architecture. Next to this, the thesis will identify potential improvements in the service and make recommendations for a successful market penetration. This work is being carried out as a part of the work of the Finnish Geospatial Research Institute within a project under the European Commission. The project's stated goal is to develop and improve the necessary infrastructure in advance to the start of operation of the Galileo HAS.

The remainder of this master's thesis proceeds as follows. First, the next chapter introduces the theoretical foundations for the work (Chapter 2). This includes an overview of the domain of satellite positioning and an analysis of relevant practices and techniques. Based on this and an analysis of the Galileo HAS, Chapter 3 outlines the design of the previously mentioned software suite with important constraints and identified requirements taking into account. In Chapter 4, the implementation work is described in detail. This is not limited to the actual practical work but also includes the complete work methodology. Afterwards, both the outcomes and the current state of the Galileo HAS will be evaluated based on relevant performance indicators in Chapter 5. Finally, Chapter 6 completes the thesis with a concluding summary and provides an outlook with opportunities and suggestions for future work.

2 Background

The following chapter will provide a broad overview of the key concepts, work and developments in the general field of Global Navigation Satellite System (GNSS). Since the Galileo HAS will be operating and competing in the high accuracy positioning field, a special focus will be put on tools and practices in this area in order to analyse how the goal of this work can be achieved.

2.1 Global Navigation Satellite Systems

Since the first Global Navigation satellite launched into orbit in 1960 [1] as part of the American Transit system, the field of satellite navigation has undergone a dramatic shift. This first system has since been replaced by the GPS. Next to it, new GNSS constellations have been developed and launched into space in many parts of the world. Originally a part of the military endeavours in the cold war between the USA and USSR, they are now used every day by billions of users on land, water and in the sky. The use is by no means limited to positioning, but GNSS clocks are widely spread in the business world of today [6].

Generally, the concept of classic, i.e. code-based, GNSS is based on simple triangulation, as depicted in Figure 1: If a receiver has at least three different satellites in view whose positions are accurately known and it can determine the distance to each of them, calculating its own position is very straightforward. Of course, GNSS systems have to cope with some more difficulties, such as the fact that the calculation of the distances is dependent on highly accurate clocks. However, while the clocks of satellites are kept extremely accurate, such as one nanosecond per 24 hours, as is the case for Galileo [7], receivers are typically a lot cheaper and inaccurate. Thus, taking a fourth satellite into the equation, receivers are able to solve for not only the three spatial dimensions but also the time one.

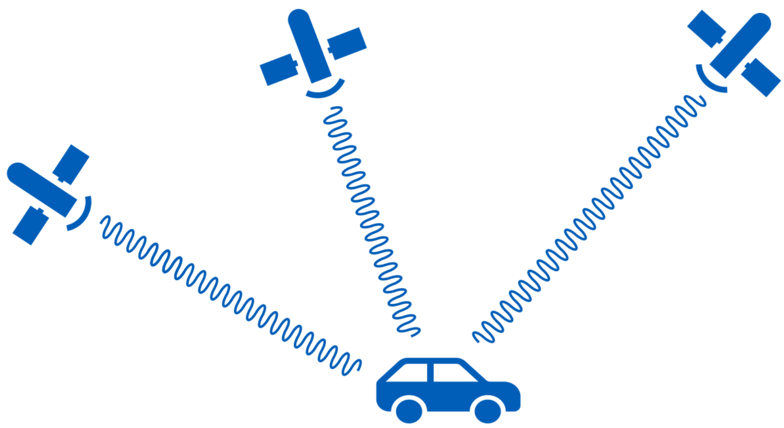


Figure 1: "Classic" code-based GNSS: Triangulation of a receiver in space and time through precise knowledge about orbital parameters of satellites

The history of GNSS can be seen as a continuous development of accuracy. Especially in the first days of public availability of GPS (1983 to 2000), the accuracy which was accessible by non-US-military-endorsed users was degraded by a pseudo-random noise introduced by the US military. This only allowed a vague positioning with an error of around 50 m vertically [8]. Only in 2000, the US president Bill Clinton announced the end of the "selective availability" (S/A). Today, the common error sources originate mostly from small errors in a few properties of satellites together with the ever-changing composition of the atmosphere. Namely, although the positions of GNSS satellites are very well known, they run on highly accurate clocks, and their software is very efficient, the speed of light is the biggest factor in the calculations. Furthermore, with a 3 ns clock error resulting in a 1 m difference in observation, small errors or delays in either of the satellites' models stand in the way of the sub-meter or even centimeter accuracy. This in turn is required by emerging applications such as autonomous driving.

2.2 Satellite Positioning Augmentation

Amongst others, these errors are responsible for a good portion of the inaccuracies we observe when using GNSS systems. For example, Diggelen and Enge determined in a worldwide experiment the mean accuracy of a GPS-enabled smartphone to be 4.9 m under open sky [9]. High-end receivers can improve this significantly, however, with emerging markets such as robotics and autonomous driving, the need for accessible high accuracy positioning becomes apparent. In order to improve the accuracy obtained using "classic" GNSS, a number of techniques has been implemented, which are able to significantly boost the performance. The following section will go into some detail on these concepts and their developments, which will help to understand the market in which the Galileo HAS will launch.

2.2.1 Differential Positioning

Differential GNSS (DGPS) was one of the first ways of augmenting the "classic" satellite positioning. The general idea of it came up in time of Transit, with developments being as old as the developments of the GPS system. The concept is depicted in Figure 2: If a base station with a known precise location is used, an error, or difference, between the true known location and the observed position can be calculated. This difference is usually comprised of different error sources combined, and it is rather similar for receivers in the same area. If this difference can then be broadcasted to users nearby, it can be applied to other observations increasing the accuracy.

As pointed out by Cardall and Cnossen [10]'s simulations, in simulations it was already shown in 1981 that the differential technique could decrease GPS errors by more than 60% for the main Coarse Acquisition (C/A) signal. As this resulted in an increased accuracy, namely from 28 meters of error to only 10, this meant the usability of the signal increased tremendously for applications in, for example, aviation. The ultimate boost for differential GPS, however, was the result of several civil agencies coping with the selective availability. As the accuracy provided by the new GPS satellites was too degraded to prove usable for applications in, for example, marine or aviation, differential GPS was found to address this issue. Multiple studies have discussed the effect of S/A on differential GPS, and, as stated by Parkinson and Spilker [11], while it still impacted the accuracy, the effect was considerably lower than on "classic" GPS. This was due to the S/A introduced error changing only slowly, and being rather constant over large areas. Thus, as long as base stations could see more or less the same satellites as a user, significant improvements can be achieved: In their study, they found to decrease the 1-sigma horizontal position error of code-based GPS from 41.1 meters to 2.2 meters using differential corrections.

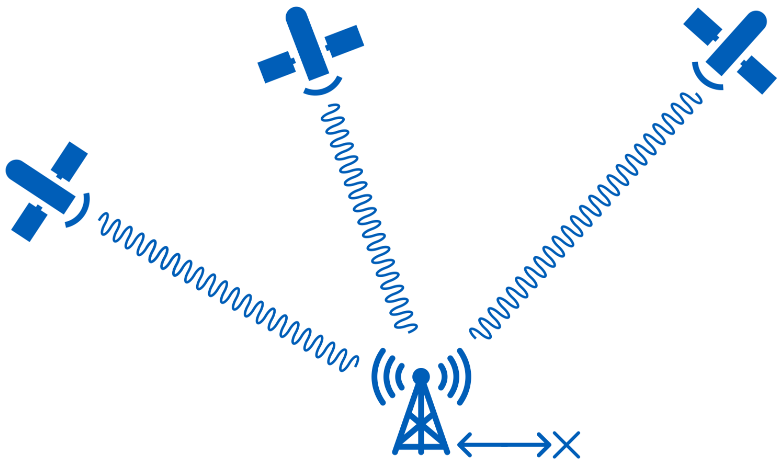


Figure 2: Differential GNSS: Using knowledge of a base station, differences between the true and observed position can be broadcasted to receivers in the area

2.2.2 Dual Frequency Positioning

Generally, the most relevant error sources impacting positioning solutions can be grouped into satellite internal and external ones. On the one side, internal errors dilute the knowledge about the states of the satellite, such as satellite clock and orbit parameters. The selective availability error can also be counted towards this category. On the other side, external factors come into play: Positioning algorithms heavily depend on the radio signals transmitted by satellites in order to travel with light speed. However, due to the refractive and diffractive properties of the atmosphere, especially the ionosphere, radio signals are temporarily slowed down or even slightly diverted in their course. Thus, the travel time of the signal is increased by these phenomena, diluting the precision of positioning. Another external factor, similar to the last point, is the multipathing error which comes into play when a receiver receives the same signal at different times, which can be the case if it was reflected by obstacles around, leading to more complications.

For the biggest error source, the ionospheric effect, in particular, dual frequency (DF) positioning offers an elegant, yet simple solution: Millman [12] analysed the ionospheric effect, and found the effect of the ionosphere on radio waves to be a function of the frequency of the signal. The induced time delay can be described by Equation 1. This frequency dependence offers an interesting possibility if two signals on two different frequencies can be processed. A linear combination of both signals' transmission durations can be constructed, which cancels out the ionospheric effect yielding an observation free of this error. The approach is depicted in Figure 3. The underlaying equations and constraints for the example of the linear combination of the GPS L1 and L2 signal are presented in Equations 2-5. Here, Equation 2 shows the general effect of the ionosphere on single frequency (SF) receivers: While ρ denotes the frequency-independent terms of the ranging calculations, the fraction is a slightly simplified version of Equation 1. Then, the iono-free combination can be written as in Equation 3. Using the constraints of Equation 4, fitting values for α and β can be deduced.

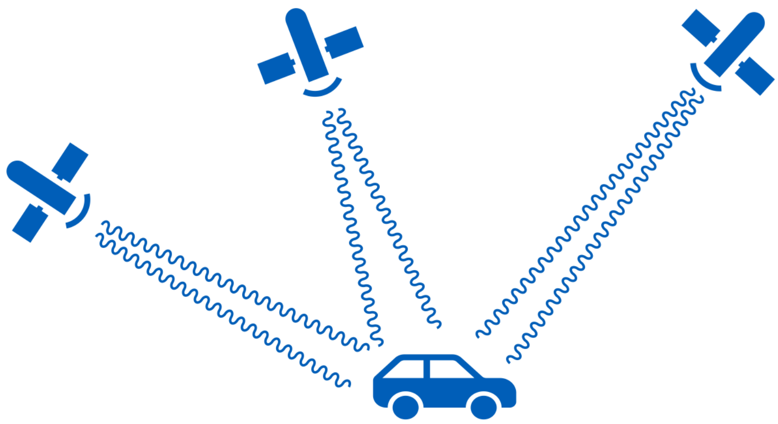


Figure 3: Dual Frequency Positioning: Utilising the frequency dependence of the atmospheric effect, two different signals together can form ionospheric-free observations

Generally, the dual frequency GNSS approach has several upsides when compared to single frequency positioning. An example of the performance of this technique is presented by the beginnings of dual frequency smartphone approaches. Robustelli *et al.* [13] analysed the possible accuracy achieved on a smartphone using dual frequency versus single frequency. In this study, they found that by using this technique, the position error could be decreased significantly from a horizontal RMS value of 4.57 m to almost 1 m. An additional effect of it is that combining different carrier frequencies enables an easier detection of multipathing errors, as has been claimed by Mubarak and Dempster [14]. The downside of it can be seen from the example equations mentioned. The two factors α and β (Equation 5), being greater than 1 in the absolute, also amplify any noise the signal was subject to. Another, rather significant shortcoming is apparent from the equations: In dual frequency, only the frequency-dependent ionospheric effects are addressed. As this leaves out other errors that can occur, especially satellite internal ones, dual frequency solutions typically are rather precise, but they include some bias which makes them less accurate.

$$\Delta t = \frac{40.3}{cf^2} * TEC \tag{1}$$

$$\mathcal{T}_{L1} = \rho + \frac{40.3TEC}{f_{L1}^2}\, meters \;\; and \;\; \mathcal{T}_{L2} = \rho + \frac{40.3TEC}{f_{L2}^2}\, meters \tag{2}$$

$$\mathcal{T}_{iono-free} = \alpha\mathcal{T}_{L1} + \beta\mathcal{T}_{L2} \tag{3}$$

$$where \;\; \alpha + \beta = 1 \;\; and \;\; \alpha\frac{40.3TEC}{f_{L1}^2} + \beta\frac{40.3TEC}{f_{L2}^2} = 0 \tag{4}$$

$$\alpha = 2.54573 \;\; and \;\; \beta = -1.54573 \tag{5}$$

2.2.3 Real Time Kinematic Positioning

Real time kinematic positioning (RTK) is an approach very similar to Differential Positioning, in which observations of a base station and a receiver are combined for high accuracy relative positioning. The general architecture is thus rather similar and depicted in Figure 4. It rose quickly after GNSS started to be well established in the public, with the first commercial solutions becoming available in 1992 [15].

The difference to Differential Positioning which makes the new level of accuracy possible, however, is that instead of using the data of the signals, the carrier frequency of the signal is used. Due to the shorter and usually consistent wavelength of these, circa 19 cm for GPS L1 for example, this opens up a new possibility: The *integer ambiguity resolution* (AR). This term describes the mathematical problem that, while the receiver can easily track the phase in combination with the satellites' pseudorandom noise (PRN) code, it can only track this fraction of a cycle. The integer number of full cycles between a satellite and the rover is not so easily obtained. However, if this problem can be solved, quasi-exact positioning becomes possible. To enable this, RTK makes use of the mentioned static receiver base station to combine measurements of both the rover and base station. When combining both measurements into double differences, clock errors on both ends are eliminated and the relative position of the rover to the base station can be determined [16].

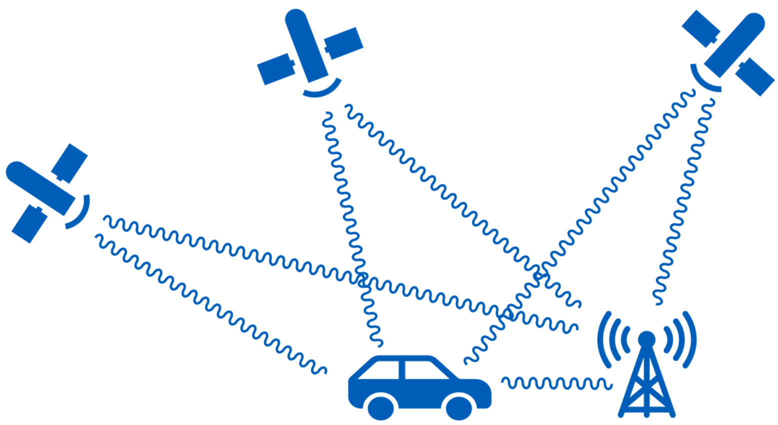


Figure 4: Real Time Kinematic positioning: Observations of two receivers are combined to achieve high accuracy relative positioning

This positioning processing is done by the base station, utilising the *Observation Space Representation* (OSR). For this, the rover sends data to the station which corrects the errors and transmits the position back to the rover. This correction process is comparatively fast and can give a position fix within a minute usually. Although this technique is highly accurate, down to errors of a few millimeters [16], the required base stations are a large constraint. As Langley describes, under optimal conditions, a typical base station's radius of operation can be 28 km at maximum. The newer network RTK (NRTK) solves this portion partially, using *virtual base stations*, but it also only increases this radius to 40 - 80km [17]. The additional need of a direct communication link between base station and rover poses an additional downside.

2.2.4 Precise Point Positioning

Precise point positioning (PPP), which came up shortly after RTK in the same decade, is taking an approach of pure broadcasting. This technique still uses base stations as computing agents. Unlike RTK, the stations calculate universally valid corrections which can be broadcasted to all users. The overall architecture is thus very similar to the one depicted in Figure 2, apart from their different internal calculations. Specifically, fixed stations with known positions calculate corrections for the most common, satellite-specific errors and biases. Just as for differential GNSS, these corrections can then be applied to observations made by the user. As mentioned previously, this mitigates errors in the satellite internal clocks and their position as well as biases in software timings. This also shows the core difference to the previous augmentation methods: These corrections are transmitted in a *State Space Representation* (SSR), describing the actual state rather than an observation. This state includes corrections for satellites rather than locations, which makes physical distance to the base stations unnecessary. As the errors are universal, a user is not dependent on observing the same satellites as the station. This format was established by Wübbena *et al.* [18] in 2005. When applying these corrections to a sufficient number of satellites, the user's position can be estimated down to a centimeter or decimeter level [15].

The advantages of this technique is obvious, only a few base stations worldwide are needed, while corrections are universally applicable. This makes a scaled broadcast solution possible, especially in combination with transmitting satellites, which eliminates the need for a cellular connection entirely. PPP is especially interesting for applications where an accuracy of centimeters is sufficient. However, PPP has a significant downside as well. Since it is only using a single receiver, and the ambiguity has still to be solved, a typical PPP convergence will take 20 - 40 minutes to reach a horizontal error of below 10 cm.

Both the commercial and research sector have been experimenting with combining the advantages of RTK and PPP, and a somewhat hybridisation of both called PPP-AR has since become established in the market. In this case, AR stands for the aforementioned ambiguity resolution. The European GNSS Agency (GSA) highlights that this technique is able to mitigate the main disadvantages of both whilst retaining their advantages [19]. Specifically, the technique compliments the SSR messages from PPP with additional atmospheric corrections, which are locally calculated by a denser network of base stations. Including this additional bit into the correction set enables receivers to perform ambiguity resolution (AR), making even more accurate and permanent position fixes possible. Whilst increasing the amount of stations needed in the network, corrections can still be broadcasted, although they stay only available in a certain area. This information has the additional effect of decreasing the convergence time to less than 10 minutes usually, and it sometimes can be achieved in seconds [19]. Another advantage is the versatility, i.e. when a user moves or starts outside of the atmospheric corrected area, they are still left with conventional PPP.

2.2.5 Augmentation Market

Some general conclusions on the landscape of the high accuracy market can be drawn from the main augmentation techniques presented in the previous sub sections. Here, it is also beneficial to note the different application areas. An important distinction to be drawn here is between high-end devices which are used, for example, in autonomous systems or for niche applications such as surveying and consumer devices, most notably smartphones. These former devices utilise to high quality antennas and processing modules which make some techniques like RTK or other ambiguity resolution ones more feasible. The latter, on the other hand, use lower quality receivers. These are enough for the classic, code-based positioning, but are more prone to errors, which make more sophisticated approaches harder to implement.

Overall, the different techniques have varying levels of spread in the market: Dual frequency GNSS is widely used and is starting to be even found in newer smartphones, as presented before [14]. For pure satellite-based positioning, this technique can be seen as a simple-to-use baseline, as it is independent of other information. As such, it also can be seen as a direct competitor to the precise point positioning technology. Real time kinematic positioning has also been widely adapted by the first movers in high accuracy positioning since it was first standardised by the Radio Technical Commission for Maritime Services (RTCM) in 1994 [20]. Since then, it became the de-facto standard in this market, with multiple providers in Europe and Northern America. Precise point positioning, on the other hand, seems to have been largely overlooked by the general user base, perhaps being overshadowed by RTK. New providers of PPP corrections were mostly seen to enter the market from around 2010 onward, with the RTCM standardization process still being unfinished in 2021.

It can be concluded that especially the two most promising GNSS augmentation methods meet the needs of different markets: In the case of RTK, the need for a dense network of base stations together with a high-bandwidth, bidirectional communication between rovers and stations, make it scale very badly to the mass-consumer market's requirements [19]. This claim is also further backed by the fact that in their report from 2005, Engfeldt [21] found that in Northern and Central Europe there were in total less than 5000 users of the seven most relevant RTK service providers. PPP, however, can be assumed very scalable due to the lower number of base stations needed and the ability to offer the service in a broadcast format. Although it offers a lower performance than RTK, the possible accuracy of it is sufficient for many consumer-grade applications. Especially considering the quasi-hybridisation of the two, PPP-AR, which eliminates the long convergence period of PPP, this technique is able to cater to the mass-market.

2.2.6 Galileo High Accuracy Service

In response to the need for high-accuracy positioning in these large emerging markets, the European Commission decided the implementation of the Galileo High Accuracy Service (HAS). This service is targeted to be the first, globally available PPP correction service utilising the Signal-In-Space (SIS) as a propagation method [5]. Initially planned as the *Commercial Service*, the project was later renamed and decided to be available free of charge. According to Fernandez-Hernandez *et al.*, this decision was made to "foster innovation in both consolidated and emerging markets, notably in key areas such as drones and autonomous cars" [22]. The release of this service is set to be done in three stages, with the first one, *HAS testing and experimentation* taking place at the time of writing. The first public availability of the service will come with phase 1, *Initial Service*, which will provide global PPP corrections for Galileo and GPS and aims at a convergence time below 5 minutes. The GSA expects this phase to start in 2022, followed by phase 2, *Full Service*, in 2024. With the full service, the PPP service is complimented by atmospheric corrections within the European Coverage Area, in order to enable PPP-AR and reduce the convergence time to below 100 seconds. The targeted accuracy in both phases is 20 cm horizontally and 40 cm vertically. At the moment, the service is in the final development stages, with the dissemination format of correction information being close to finalised. For this, a state space representation format was developed based on the Japanese Compact State Space Representation (CSSR).

As the goal of the provision of the service for free was to promote innovation, the impact of the Galileo HAS on the existing high-accuracy market has been considered. Next to stakeholders from the target markets, other high accuracy providers have been included in the shaping of the service. This has led to a slight shift and some clarifications in the final service specification. On the one side, the target accuracy has been reduced from the initial 10 cm to the current 20 cm goal. This was assumed to be sufficient for the target areas, as presented in the Commission Implementing Decision 2018/321 [4]. Additionally, the operational phase was pushed back from 2018 to at least 2020, as announced in 2018 [22]. However, seeing the current delays, an initial start of operation in 2018 is to be seen questionable already besides this. On the other side, the HAS has been clarified to provide neither any specialised service nor anything beyond the high-accuracy data itself. These changes are expected to leave the commercial market untouched as far as possible, while still providing the key areas identified with sufficient data to allow for new applications to be developed [22].

2.3 State Space Representation Formats

There exists a number of published open and proprietary formats created for the purpose of efficient dissemination of SSR information. The constraints of the individual application areas they are developed for play a large role in the design of them. However, data efficiency can generally be identified as a common design goal which plays an especially large role if satellites are used as a transmitting medium, due to a smaller bandwidth availability. In the following sub sections, the most established formats will be presented and their relevance discussed.

Vana *et al.* [23] and Hirokawa and Fernandez-Hernandez [24] identified the most relevant SSR formats to be:

- RTCM 3.0
- CSSR (Compact SSR)
- SPARTN (formerly SAPA)

3GPP LPP, which is aimed at the mobile network based dissemination of correction data via LTE or 5G, is omitted due to the very different application area. In turn, the open *IGS SSR* standard by the International GNSS Society (IGS) and the preliminary format of the Galileo HAS will be presented alongside.

2.3.1 Galileo High Accuracy Service

The format used by the Galileo HAS is still being finalised and may be changed in the future as stated by the GSA [25]. However, the internal documentation is in its final state and has reached the testing and evaluation phase. This thesis' work is part of this last phase. In general, the format is based on the open Compact SSR format, used by the Japanese Quasi-Zenith Satellite System (QZSS) programme. The undertaken changes enable for smaller messages and more reliable reception of corrections.

Generally, the Galileo HAS format only consists of one single message type (MT), namely MT1. To allow for varying update intervals, however, the message contains a content block which indicates the availability of different information blocks. The different blocks and their information content is presented in Table 1. The usage of masks reduces the amount of needed bandwidth, as satellite PRN codes do not need to be transmitted for each satellite. Furthermore, the possibility of a clock correction message for only a subset of satellites takes into account the different qualities of clocks on satellites, partly mitigating the absence of a timed component in the correction.

Block	Content
Mask	Indicating corrected satellites and signals per GNSS system. Optional: Signals per satellite (Cell mask)
Orbit Corrections	Radial, along- and cross-track deltas
Full Clock Corrections	Clock delta (C0) for all satellites in mask
Subset Clock Corrections	Clock delta (C0) for satellite subset. Can contain do-not-use value
Code Biases	Code bias(es) per satellite
Phase Biases	Per satellite: Biases; Per bias: Discontinuities, bias

Table 1: Galileo HAS Message contents

The dissemination scheme has been adapted in order to account for two main constraints of the system: Often, users have to cope with low satellite availability [26], but at the same time need access to messages significantly longer than the usual ones in order to correct satellites in view [27]. In their paper, Fernandez-Hernandez *et al.* describe a method in which a message M is divided into k pages, where k is determined by a fixed page length of 492 bits. The pages' bytes are then vertically encoded using a Reed-Solomon encoder, effectively creating 255 pages from the original maximum of 32 pages, with empty pages being filled with zeros. This allows for a decoding of the original message, as soon as k different pages of the 255 possible ones are received. Since the decoding of an uncorrupted Reed-Solomon message can be done with a single matrix inversion and multiplication, the decoding can usually be performed highly efficient. In tests under realistic conditions, the authors found a time-to-retrieve-data of around 9 s, reaching up to 45 s in worst-case scenarios. In general, the splitting of workload to multiple satellites is able to improve reception times a lot. While the fixed size of pages can generally lead to unnecessary bits being transmitted, the relatively small size of pages combined with the 1 s transmission time [25], the dissemination scheme is a step towards a reliable high accuracy service.

2.3.2 RTCM 3

The RTCM is an international standardization organization consisting of organizations from around the world with government, educational and professional members. These sometimes form working groups to publish open standards. The RTCM 3 message format is developed by the RTCM SC-104 working group, and their aim is to develop message formats for the Differential Global Navigation Satellite Systems (DGNSS). Despite the name, RTCM open standards are widely used outside of maritime applications, which started especially from the efforts of the SC-104 group and led to a continuous development and rising interest in it [16], [28].

In the past, this working group standardised a number of message formats already, such as the OSR messages used by RTK applications. Version 3.0 [29] was prompted by heavy concerns about the efficiency of the previous RTCM 2.0

Message	Content
Orbit Corrections	Radial, along- and cross-track deltas & velocities
Clock Corrections	2^{nd} degree polynomial (C0, C1, C2)
Highrate Clock Corrections	Clock delta (C0)
Orbit+Clock Corrections	Radial, along- and cross-track deltas & velocities, 2^{nd} degree clock polynomial (C0, C1, C2)
Code Biases	Code bias(es) per satellite
Phase Biases	Dispersive bias consistency indicator, MW consistency indicator, satellite corrections Per satellite: Yaw (angle & rate), biases; Per bias: integer indicator, WL-integer indicator, discontinuities, bias, standard deviation
URA	User Range Accuracy

Table 2: RTCM 3 SSR Message contents

messages. Additionally, since version 3, it also includes format specifications for the SSR messages which are needed for PPP. On a high level, the information content of these messages can be found in Table 2, displaying a high information count present.

In general, the format allows for messages of variable length which helps in reducing bandwidth usage. The information necessary for decoding can be read from the transport layer, adding to the versatility. The only limit for message sizes is the 10-bit length indicator, requiring messages longer than 1023 bytes to be split into multiple messages. Here, it differs from the Galileo HAS format which is dependent on fixed-size pages. Furthermore, Wübbena [30] states that RTCM 3 aims for its messages to be self-contained, providing all necessary information within themselves.

However, being the work of more than a singular entity, the RTCM 3 standard is showing relatively large shortcomings in terms of development. According to the GSA [19], the RTCM SC-104 working group has not made any progress in the standardization process since February 2011 due to too many private parties with vested interest. These parties often have already put their own solutions into place and are now reluctant to change these. It is therefore to be expected that standardization efforts will continue to be less fruitful in the future. However, as an answer to this, the RTCM SC-134 for safety critical applications has been formed [28]. It is understood as an evolution of the SC-104 group with fewer participants which seems to be progressing with new standardization efforts [19].

This problem also poses the most significant issue with this format, since it is still lacking not only final specifications for atmospheric correction messages but also official standards for GNSS constellations besides GPS and GLONASS. Galileo, amongst others, did not yet receive final message layouts, even for the basic PPP corrections. Both of these points create their own problems, as on the one side, the available correction messages only support "normal" PPP, instead of the more advanced PPP-AR, and on the other side, none of the draft messages available for

Galileo are final and may be changed in the future.

In conclusion, the ultimate acceptance of RTCM SSR message can be questioned, seeing how the development stagnated over the years. If the SC-104 committee fails to continue standardisation much longer, it can be expected that other open standards, developed by singular entities, will be too established to overcome. Yet, the great influence of the RTCM standards in general can not be overlooked, with some new standards even actively claiming RTCM-compatibility [23].

2.3.3 IGS State Space Representation

The IGS SSR standard, developed by the IGS, is an open standard [31] which aims at helping the dissemination of SSR as a technology with the focus on real-time products. Being part of the SC-104 working group, the standard can be understood as a RTCM-compatible standard, both in the way that IGS SSR messages can be interpreted using the RTCM message type 4076 and by closely following the format specifications of RTCM 3. Thus, Table 2 can be referred to for the content of IGS SSR messages, with the exception of an adapted phase bias message and the ionospheric message, presented in Table 3.

Message	Content
Phase Biases	Dispersive bias consistency indicator, MW consistency indicator, satellite corrections Per satellite: Yaw (angle & rate), biases; Per bias: integer indicator, WL-integer indicator, discontinuities, bias
Ionospheric correction	VTEC spherical harmonics model

Table 3: IGS SSR Message contents

The two main advantages of this format arise out of minor advancements in the standardization progress and its openness. For example, support for more constellations besides GPS and GLONASS and a format for ionospheric corrections have been added early on. The content of messages correcting the "new" GNSS systems is equivalent to the content of existing RTCM 3 messages. Additionally, the IGS published the documentation openly, not requiring purchasing an official guide, as it is the case for RTCM 3.

The IGS SSR format is already in use for the real-time products of the IGS, which is well established in the field, suggesting a broad user base. In general, due to the similarity to RTCM 3, the advantages seem minor. However, it is still more advanced than RTCM 3, and if the SC-104 does not advance quickly, a broader user base will be expected due to compatibility whilst giving the advantage of a "final" standard.

Message	Content
Mask	Indicating corrected satellites and signals per GNSS system. Optional: Signals per satellite (Cell mask)
Orbit Corrections	Radial, along- and cross-track deltas
Clock Corrections	Clock delta (C0) for all satellites in mask
Code Biases	Code bias(es) per satellite
Phase Biases	Biases, discontinuities
Code+Phase Biases	Code biases, phase biases, discontinuities
URA	User range accuracy for each satellite
STEC	STEC quality indicator, polynomial coefficients (1st-3rd degree polynomial possible)
Gridded correction	Per grid point: Hydrostatic&wet tropospheric vertical delay, STEC residuals for each satellite

Table 4: CSSR Message contents

2.3.4 Compact State Space Representation

CSSR has been developed in Japan by the Quasi-Zenith Satellite System Services Inc. as a SSR format for the operational QZSS, which provides an experimental PPP service for centimeter-level augmentation. Since the start of this service, it has been released as an open format for SSR Messages [32].

The CSSR standard is designed in such a way that it keeps RTCM-format conformity and is able to be interpreted as a proprietary message in RTCM interpreters, but reduces the bandwidth usage significantly. As the Galileo HAS format is based on CSSR, the properties are the same for large parts, for example the usage of a mask message and a cell mask for biases. However, being a finished standard, the format does have a URA message and both functional and gridded atmospheric correction messages. The information contained in each message are presented in Table 4. Again, one of the main saving factors is the omission of timed components in the corrections. On the other hand, CSSR messages are fixed to a size of 2000 bits, including a 256 bit Reed-Solomon code. This on the one hand helps with recovering from errors in transmission, but it usually leads to heaps of unnecessary information being transmitted, especially if not used appropriately.

Having been in use by the QZSS service for some time already, the CSSR format can be considered to be well-tested. At the same time, it offers the advantage of a very compact message format whilst containing all relevant information for high accuracy PPP.

2.3.5 SPARTN

SPARTN (formerly developed as SAPA) is an open industry standard developed by Sapcorda Services along with key players from both the automotive and the GNSS sector with the aim to meet the requirements needed by modern industry applications. Possibly also due to the involved parties, it claims to be focusing on

Message	Content
OCB	Per satellite: Do-not-use flag, continuity, orbits, clocks, biases
Orbits	Radial, along- and cross-track deltas, yaw (optional)
Clocks	Clock delta (C0), user range error
Biases	Phase bias mask, phase biases, code bias mask, code biases
Phase	Fixed/Floating ambiguity flag, continuity, bias
Code	Bias
HPAC	Gridded tropospheric correction, ionospheric correction per satellite
GAD	Geographic area definition (lat/long) per area
BPAC	Gridded VTEC residuals

Table 5: SPARTN Message contents

the automotive industry [33]. The format is partially still in development with new features releasing occasionally. The most recent update included support for three new constellations in June 2021, Galileo amongst others [34].

Generally, the SPARTN format is clearly aiming at the conservation of bandwidth, with multiple techniques being used to reduce message sizes. Similar to previously presented formats, a satellite mask is used reducing the bits needed per satellite. However, this approach is taken a step further, by giving the possibility to dynamically size the satellite mask making adaptations of smaller constellations more efficient. Furthermore, some data fields can be disabled using flags, while other data fields, similarly to the satellite mask, exist in various resolutions or ranges so that the messages can be tailored to the existing data conserving bandwidth [23].

The general information content of the SSR messages can be found in Table 5. It has to be noted, that although the basic SSR corrections are given in a single message, similar to the Galileo HAS format, the presence of blocks is indicated by flags for orbit, clock and bias corrections each. Thus, messages can be created based on data availability without transmitting redundant information. The choice of making the satellite mask a fixed part of every message whilst generally increasing the size of the messages, has the advantage of making messages independent leading to a faster availability of corrections on average. Furthermore, quality indicators and do-not-use flags in the messages can further help in increasing the safety of use.

In conclusion, the format does seem to be in a strong position with multiple relevant stakeholders involved in the development. The focus on the automotive industry can further help with the market penetration, because the needs of a single sector can be fully taken into account, such as safety features and good availability. Especially Vana *et al.* [23] presented good arguments in favour of the standard, and the recent increase in supported GNSS constellations shows active development and a continuous interest.

2.4 Precise Point Positioning Tools

In order to make use of SSR corrections in combination with GNSS receiver measurements, users can use various PPP software projects, both from the commercial and the Open-Source domain. Since the goal of this work is to make the Galileo HAS as accessible to everyone as possible, only the latter will be considered.

The most common Open-Source tool for high-precision GNSS processing is RTKLIB. The software, which has been available since 2006 in a first version, is distributed free of charge and has become quite versatile and commonly used since then [35]. Today, it supports most of the common GNSS constellations and most positioning modes, including dual-frequency, DGPS/DGNSS and PPP. It has to be noted that the original version is only supporting the finalised RTCM 3 messages (see 2.3.2), and it is thus missing support for not only constellations besides GPS and GLONASS but also phase biases in general.

There exists an Open-Source fork of RTKLIB [36], based on version 2.4.3, which adds support for the other constellations and phase bias messages. However, this one does not process phase biases, thus, PPP with ambiguity resolution is not an option in this version either. As the project stopped working on the Open-Source version, but decided to take it to a commercial level, only this outdated version is available.

Another Open-Source project called PPP Wizard has been developed by Laurichesse and Privat at the French CNES. It is based on RTKLIB 2.4.2 but adds functionality for full PPP with zero difference ambiguity resolution [37]. Support for Galileo and Beidou has been added, however, the amount of supported receivers is severely limited and some newer Galileo messages, such as the F/Nav message, are not being decoded.

Next to these RTKLIB-based projects, a standalone PPP library is offered by Chen and Chang [38]. The PPPLib offers efficient computation of PPP solutions, including all major GNSS constellations. Additional to atmospheric corrections, it even includes simple tools to visualise the data. However, the software is solely relying on IGS file formats whilst offering no support for SSR messages.

3 Design

Summarising the findings of Chapter 2, multiple existent formats for GNSS SSR corrections with various degrees of market penetration have been identified. The driving factor for most developments, next to slow standardisation efforts, seems to be the message size, with new formats utilising different techniques to reduce the needed bytes to convey necessary information. Additionally, the Open-Source solutions available for PPP solutions, even if mostly reliant on RTKLIB, are in different states of development. From all this, three action points have been identified as viable to the goal of this thesis:

1. Implementation of a HAS Decoder library
2. Updating of an existent Open-Source PPP software
3. Analysing the HAS message specifications

Especially (1) and (2) are deemed particularly vital, with (3) having the potential of improving the usability and competitiveness in the long run. In the following, these three actions are discussed and the design is worked out. This includes justifications for any design decisions made and the overall structural concept.

3.1 Decoding Library

Initially, only those parties involved in the development of the Galileo HAS will have the capability to utilise the Galileo HAS. However, as pointed out by Fernandez-Hernandez *et al.*, the goal of the freely available HAS is to increase the innovation output within the European Union, which requires an accessible service. Thus, it is beneficial to implement an Open-Source decoding software which is able to decode the HAS messages from the SIS and convert them into other widely spread formats, such that the service can easily be integrated into existing solutions. In doing so, potential users can start using the HAS from day one on, without having to wait for receiver manufacturers to roll out updates to their software and without having to adapt their own.

Several requirements can be derived from this: Firstly, the library has to be able to run efficiently in a real-time scenario, not taking up too many resources. This is to not impact the users' software significantly. Secondly, it should be easily usable, ideally platform-independent and integrate into existent routines. Lastly, it should be easy to adapt the library to the users' need, either by settings or by simple reprogramming. As this project is carried out within a project under the Finnish Geospatial Research Institute and the European Commission, the software design will partly be influenced by the expressed requirements and constraints.

3.1.1 Message Decoding

To ensure consistent availability, Galileo HAS messages are vertically encoded in a Reed-Solomon scheme, spreading a full 32-page message over 255 pages. In practice, for a message with k pages, this means that decoding of it possible as soon as k different uncorrupted pages have been received. Generally speaking, this encoding increases the pool of pages to transmit without increasing the information necessary for reading the full message. This in turn allows for an effective global dissemination scheme, utilising all satellites the same. While usually one of the advantages of Reed-Solomon codes is the error-correction, the incorporating of the message cyclic redundancy check (CRC) enables a faster decoding scheme. Since a Reed-Solomon encoding is a linear operation in some Galois field, both encoding and decoding is possible to be expressed as a matrix multiplication inside of this field. As is presented in the Galileo HAS interface documentation [25], Equation 6 presents the encoding scheme for the full message vector c, utilising a generator matrix G which results in the encoded code vector Γ.

$$\boldsymbol{\Gamma} = \begin{pmatrix} \Gamma_0 \\ \Gamma_1 \\ \vdots \\ \Gamma_{254} \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{31} \\ \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{222} \end{pmatrix} = \begin{pmatrix} \boldsymbol{c} \\ \boldsymbol{\gamma} \end{pmatrix} = \boldsymbol{G} \cdot \boldsymbol{c} = \begin{pmatrix} \boldsymbol{I} \\ \boldsymbol{P} \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{31} \end{pmatrix} \tag{6}$$

Galileo HAS messages can thus be efficiently be encoded and decoded via simple matrix operations: For this, the generator matrix G is a combination of two submatrices I and P. I is an identity matrix of size 32x32, meaning that the first 32 of the 255 pages are the unchanged HAS message pages. The second, P, is a dense submatrix of size 223x32, responsible for the other 223 parity pages. Due to this approach, decoding becomes equally simple, eliminating the need for a resource expensive Reed-Solomon decoding. Instead, incoming pages are identified by their page number, stored in the unencoded page header. Thus, once a sufficient number of k different pages is received, message bytes are vertically grouped into words, with their page ID resembling the index of each byte in the new words. Each of these words is then decoded through fast matrix multiplication within the respective Galois field. For this, the decoding matrix is constructed by inverting a submatrix of G which consists of those rows and columns of G which correspond to the received page IDs. Due to this decoding matrix being the same for each vertical word, only one inversion is required for the whole message. This further reduces the resources necessary for the decoding process. For a more extensive description of the encoding and decoding process, please refer to the interface documentation [25].

3.1.2 Format Choice

From the previously presented formats, the best fitting two are chosen to be possible output formats of the decoder. For this, the market penetration is to be considered along with general aspects of message efficiency and a fit-to-use value calculated using the consistency between HAS data compared to format data.

Generally, it can be said that the RTCM 3 standard has the advantage of coming from a de facto official standardisation committee. Although progress is slow, their influence is undeniable, and even competing formats like IGS or CSSR claim to be compatible with RTCM in general. The IGS SSR format, whilst practically finalising the RTCM 3, seems to be mostly used by IGS themselves, as a means to disseminate their correction products. The same goes for the SPARTN format which is used by the u-blox PointPerfect service. The CSSR format on the other hand, is currently only used by the Japanese QZSS for PPP services in Japan.

Regarding message efficiency, Vana *et al.* [23] analysed the previously presented formats for bandwidth requirements, revealing different results for different corrections. Generally, it can be said that SPARTN and CSSR messages are significantly more efficient than RTCM 3 and IGS messages, partly because these formats utilise satellite masks. This makes messages scale a lot better than transmitting PRN codes for each satellite. On average, SPARTN and CSSR are shown to transmit up to 61% and 80% less data, respectively, for a message containing corrections for 20 satellites, which is a rather usual use case. Here, the difference for clock corrections is significantly larger than for orbit corrections.

However, the lower use of bandwidth is in large parts explained by the lower information content in the different formats. RTCM 3 and IGS messages contain not only offset values but also the first and second order derivatives of those, which leads to more stable solutions over time. Of these, SPARTN and CSSR only include the former. This ultimately leads not only to less stable solutions over time, but also potentially to a higher required update rate of the messages. Table 6 summarises the information content of each format. In this, an "x" indicates the presence of a particular feature. Additionally, an "x" in brackets indicates the theoretical presence of a feature, be it plans to include it, optional features, or similar.

This comparison is used to calculate a fit-to-use value, utilising a weighting system. For this, individual weights are based on the importance on the one, and on the non-necessity of data not available in the HAS on the other side. It is important to note that at the time of designing, the most recent SPARTN ICD was v1.8 [39]. Since then, the format included three additional constellation in its ICD 2.0 from June 2021 [34], which could not be considered in this work. Additionally, also draft messages have been considered for the capability assessment of RTCM 3. It can be noted that generally, the differences between the formats in terms of information content are minuscule. All formats fulfill the basic requirements of information content needed to be transmitted, with the exception of SPARTN, which, as pointed out earlier, did not have support for the Galileo constellation. Generally speaking, the advantages of the formats lay mostly in the saving of bandwidth, which can be considered nearly irrelevant in the use case of data processing within one computer.

Taking into account all presented elements, it was advised to proceed with RTCM 3 and IGS SSR as initial target formats. Although CSSR was showing a better fit value, which is no surprise as the HAS format is based on it, the limited international use does not make it a viable option. Conversely, it was recommended to add support for SPARTN when support for Galileo was added, as the do-not-use feature is not existent in the other formats. Additionally, the already existent use of the format by u-blox users makes it a relevant format.

Properties	HAS	RTCM 3	IGS	SPARTN	CSSR
Constellation					
GPS	x	x	x	x	x
GLONASS		x	x	x	x
Galileo	x	x	x		x
BeiDou		x	x		x
QZSS		x	x		x
Information					
Mask	x			x	x
Cell mask	x				x
Orbit corr.	x	x	x	x	x
1st/2nd order		x	x		
Clock corr.	x	x	x	x	x
1st/2nd order		x	x		
Factor (*x)	x				
HR clock corr.	(x)	x	x		
Code Bias	x	x	x	x	x
Phase Bias	x	x	x	x	x
Yaw		x	x	(x)	
Integer		x	x		
Wide lane		x	x		
URA		x	x	x	x
Ionosphere	(x)		x	x	x
Troposphere				x	x
Do-not-use	x			x	
RTCM-compatible		x	x		x
Individual messages	(x)	x	x	(x)	x
Message/page size	492bit	<1024b	<1024b	<1024b	2000bit
Fit-to-use	9.7	8.4	8.6	6.2	9.6

Table 6: SSR Format Comparison and fit-to-use value: Combined with market spread, it was advised to proceed with RTCM 3 and IGS SSR

3.2 Positioning Software

The PPP Wizard has been chosen as the Open-Source PPP processing library to be adapted to work with the HAS Decoder. Compared to the standard RTKLIB, the PPP Wizard has several advantages, although some issues have to be solved still. Firstly, RTKLIB does not support many SSR message types yet, and only the RTCM 3 finalised ones are included, which limits the supported constellations to GPS and GLONASS, whereas PPP Wizard added draft support for the other major constellations. Additionally, the PPP Wizard was built on top of RTKLIB, extending its functionality using additional files on top which makes the adaptation of it relatively straightforward. This is a necessary step, as the available Open-Source version of it is using an outdated version of RTKLIB, with some newer GALILEO messages not being processed. Lastly, as the goal of the PPP Wizard was to be a precise point positioning demonstrator "with integer and zero-difference ambiguity resolution" [37], the advantage of it is the processing of phase biases, which RTKLIB is not yet capable of. For this, the supported RTCM 3 messages have been extended to mirror not only the finalised but also the proposed message types besides the missing constellation-specific ones.

However, the Open-Source development of the PPP Wizard was discontinued, which poses some problems that need to be resolved in order to make it fully usable. Firstly, the supported receivers are currently limited to the u-blox LEA-*T series and the NV08C-RTK receiver. As RTKLIB generally has support for more receivers, this will have to be enabled. Additionally, the RTKLIB version used by the PPP Wizard will need to be updated and support for newer Galileo messages such as the F/Nav has to be taken care of. Finally, an effortless interfacing between decoding library and PPP Wizard needs to be ensured.

3.3 Full Software Architecture

The Galileo HAS Decoder library is designed with the Open-Source context in mind, offering a mostly modular class structure which can easily be extended. The most important types of classes and their functionalities are presented in the following:

- Reader
  The reader classes are responsible for handling the input streams into the library and parsing them into the processing routines. Data recordings can be read from Septentrio .sbf and from BINEX files. For BINEX, both forward and backwards readable records are permitted. Additionally, live or pre-processed data in either sbf or BINEX format can be read from Serial ports and TCP servers.

- HAS
  Incoming, encoded HAS message pages are stored in a HAS message collection class, responsible for sorting and handling individual HAS messages, storing received masks and decoding messages with enough pages received.

- SSR
  Messages are read into a quasi-universal SSR class structure which can easily be used to extend the library's supported formats to other incoming or outgoing SSR formats. In its current state, the structure stores all information evident in the HAS ICD 1.4 [25]: Orbit and clock corrections, and signal biases for the satellites and other structural information like masks.

- Converter
  From the previous SSR classes, the converter class can read relevant data and convert it into other SSR message formats as defined earlier.

- Output
  Converted messages can be output in different ways. In order to make the output as accessible and as extendable as possible, different classes exist for them. The most straightforward one creates a file and write messages to it. However, if desired, data can also be written to a TCP server for further live processing. Lastly, a specified output format is available which is expected by the processing part of the PPP Wizard software. This is possible to be output both as a file or into the standard output stream.

- HAS Converter
  The HAS Converter class is the main interface class of the library which ties together all of the other functionalities in a simple-to-use way. For all of the subroutines, this class offers access to most of the parameters that enable even more sophisticated use cases.

The modular structure of the code makes it simple to not only extend but also only use exactly those parts of the software suite that are needed for other projects in which the library may be useful. Furthermore, the input and output modes of the library have been chosen to fit most use cases and make the usage as accessible as possible. For this, usual modes have been analysed and were combined with knowledge of existing software solutions in place at the partner's facility.

Next to this, the PPP Wizard comes with four main executables, out of which two are relevant to this project in a typical use case. Both are usable standalone and described below:

- getStream
  The getStream programme compiles data from multiple input channels into one datastream which can be output into a file or fed into the processStream executable directly. The configuration of getStream is done with the conf_get.txt, with one line per input channel, following default RTKLIB conventions. Important to note is the order, as rover/receiver data should be put first, with other streams (like the correction stream) after.

- processStream
  This part of the software provides the PPP computational part of the suite, utilising the integer and zero-difference ambiguity resolution algorithm presented by D. Laurichesse and A. Privat [37]. The configuration of this is done through the conf_process.txt file.

The full software structure and flow of information is depicted in Figure 5. As can be seen, the reader classes are the central part to the software, handling the optional conversion and outputting steps. The interfacing of the decoder library with the PPP Wizard is the next step which has to be taken care of. As discussed, this can be done using a TCP server and the getStream executable to connect via a TCP client. Alternatively, if a single feed with all GNSS information is available, the library can output the data in a PPP Wizard-compatible format. This means that the acquisition executable, getStream, can be skipped and the decoder output is directly processable via a file or the standard stream.

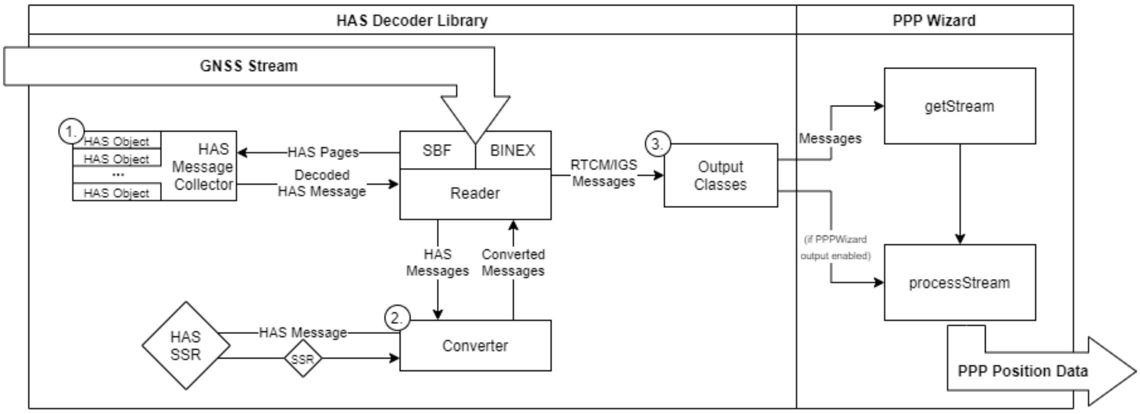


Figure 5: Software Suite Structure: Data flow within subroutines

3.4 Message Analysis

As discussed, message size efficiency is currently one of the main drivers in the development of new formats. Due to the relatively low data rate of the HAS dissemination, 448 bits per second per satellite, this is of particular concern for the Galileo HAS. In the current pilot projects, HAS messages have between 2 and 15 pages, depending on the information content. However, being still in development, the HAS can be expected to see increases in capabilities. On the one side, this would move in directions already planned, like adding additional code biases for constellations. On the other side, in the longer run, the addition of, e.g., the GLONASS constellation would be a natural next step. Having this in mind, the sizes for full messages are only expected to increase in the future. With a single page being transmitted per second by one satellite, this could lead to problems in the service. Table 7 showcases three scenarios of the HAS, together with the resulting requirement for HAS capable and transmitting Galileo satellites in view. This requirement can easily be calculated from the corrections provided: While clock corrections need to be updated every second, other corrections including orbit corrections and biases are only provided over a time frame of 7 s, as they are significantly bigger in size. However, in a study performed by Zhang *et al.* [40], they usually had from 6 to 8 Galileo satellites in view. Along with the danger of messages to be corrupted, or the low chance of a user receiving double pages, this could potentially lead to insufficient data for a continuous use. Thus, it will be beneficial to users if the page count can be be reduced. Only like this, a sufficient update rate of parameters can be ensured. To this purpose, this thesis aims at proposing an alternative message structure, reducing required bandwidth under real use conditions.

Case	Required Pages	Required Galileo satellites in view
1: Status Quo	2p/s clk correction 15p/7s other corrections	4.14 → 5 satellites
2: More Content	2p/s clk correction 17p/7s other corrections	4.42 → 5 satellites
3: 3rd Constellation	3p/s clk correction 25p/7s other corrections	6.57 → 7 satellites

Table 7: Galileo HAS Satellite Requirement, based on 1 page/second/satellite dissemination scheme

For this, typical relevant SSR data for various constellations has been analysed, revealing a tendency to cluster around values within a constellation. Thus, a new message type is proposed, utilising this clustering to its advantage. During the initial analysis phase, this effect was most present in clock and orbit corrections. Thus, the message type will include corrections for these parameters only with biases and, naturally, masks being only present in the available MT1.

The full message specification for this proposed MT2 is presented in Appendix A. Where possible, this specification is re-using existent HAS data fields as defined by the GSA [25] to ensure maximal compatibility. For all other fields, special care has been taken to exactly mirror the information content of the original message type MT1. Table 8 presents a subset of this proposed message type, namely, the clock corrections for one single constellation. The main advantage of this message type is its flexibility: Depending on how clustered or spread the corrections within a system are, fewer or more bits are used for the individual satellite's modifier which is added to the system-specific offset. This message, as an alternative to the original static message type, is expected to scale more efficiently with more satellites and offers large potential bandwidth savings.

CLOCK CORRECTIONS					
Delta Clock C0 Offset (GNSS ID 1)	±10.2375	13*	0.0025	m	Delta Clock offset value for all satellites of GNSS ID 1.
Delta Clock C0 Individual Correction Bits (GNSS ID 1)	1-16	4	1	bits	How many bits are used for individual satellite clock corrections of GNSS ID 1.
Delta Clock C0 Multiplier (GNSS ID 1)	1-4	2	-	-	Multiplier for all Delta Clock C0 corrections of GNSS ID 1 (offset **and** modifier). "b00": x1 "b01": x2 "b10": x3 "b11": x4
...					
Delta Clock C0 Offset (GNSS ID N_{sys})	±10.2375	13*	0.0025	m	See GNSS ID 1 description.
Delta Clock C0 Individual Correction Bits (GNSS ID N_{sys})	1-16	4	1	bits	
Delta Clock C0 Multiplier (GNSS ID 1)	1-4	2	-	-	
Delta Clock C0 Mod. (SV 1)	var.	var.	0.0025	m	Delta Clock C0 modifier for SV 1. "b10..0" indicates data not available. "b01..1" indicates the satellite shall not be used.
...					
Delta Clock C0 Mod. (SV$_{nsat}$)	var.	var.	0.0025	m	See SV 1 description.

Table 8: Proposed MT2 Body

4 Implementation

In the following, the implementation phase of the project will be described with particular focus on identified risks, actions taken, and other challenges faced during development.

4.1 Work Methodology

As the work was carried out within a larger project, involving multiple companies and following a strict timeline, the structure of the project was mostly given. This timeline was slightly adapted and its work packages were extended for this thesis project. It is depicted in Figure 6. In this, the most relevant work packages and milestones are presented. Officially, the project started with the kickoff in January 2021. The first phase consisted mostly of the definition of work packages and general project management tasks. This thesis' work began later in March, at T0+3.

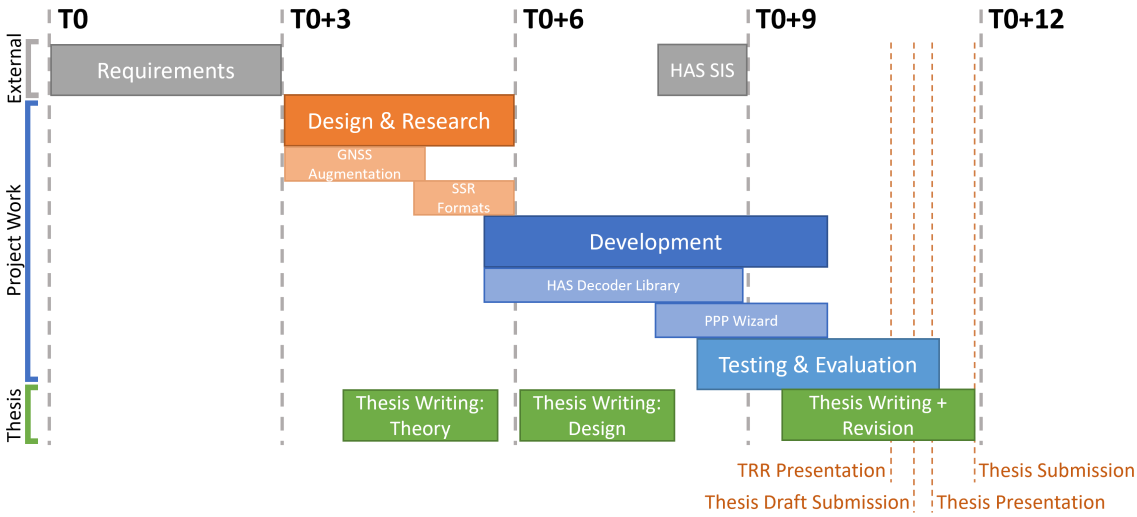


Figure 6: Project management plan: Internal and external phases and events

Generally, biweekly status meetings were carried out to update about developments, risks, and issues. Apart from this, close contact with the industry advisor was kept. This helped to keep the overall development and evaluation phase on track and within the specified timeline. Alongside with this, regular meetings and presentations with an industry expert were held to reveal and address any shortcomings or issues with the project. These meetings ensured a continuous improvement of the software and especially supported the evaluation phase.

4.2 Risks

The major relevant risks identified for this work were concerning both the decoder and the PPP processing part: On the one hand, the continuous development status of the Galileo HAS posed the danger of releasing an already outdated version, or having to adapt to newly released documentation. For this, it has been decided to concentrate on the then-current version 1.4 of the ICD [25], and evaluate necessary steps upon finishing of the project. The most recent version 1.5, which will bring the HAS development close to completion, has been analysed and was judged to have only minor impacts. On the other hand, the choice of the PPP Wizard, without in-depth analysis of the code, is a potential pitfall. As it is no longer developed in Open-Source, and support is not offered for the last open version, not much could be said about the quality of the software. However, due to the constraints of the project, namely the processing of all SSR corrections contained in the HAS, the PPP Wizard was still the most viable choice. No particular action, besides allocating a larger time frame to the work on this software, was taken.

4.3 Challenges

Apart from the general risk assessments, minor challenges were posed by the difference between SSR formats. Especially due to varying information content, decisions on the implementation of the converter had to be taken.

Firstly, the difference and compatibility between the HAS validity interval and the RTCM/IGS update interval had to be assessed. While the validity interval of the HAS is straightforward in its meaning, the update interval is more related to the expected update rate of corrections. However, it also expresses the minimum validity of the corrections. Generally, testing has shown no particular effect on changes of the update interval variable, as long as corrections are still provided regularly. Due to this and the safety reason of not increasing the set validity interval of HAS corrections, it was decided to always set an update interval lower or equal to the HAS validity interval. It is however not always possible to choose the same value, since both intervals are based on a keyed table which does not align in all possible values. However, using the *lowerUDI* setting, the converter also offers the possibility of using the next higher one in case of ambiguity. In this case the user can opt for a potentially less accurate, but more available service.

A similar issue was encountered for the IGS SSR format: While both RTCM 3 and HAS allow for a single bias being linked to two signals of the same band, e.g. E1-B and E1-C, the IGS format strictly applies a bias to a single signal. However, this was easily overcome, as individual biases can be sent for each signal. As the corrections from the converter are likely to be used in the same machine, saving bandwidth is much less an issue than for the SIS.

Another problem was found in the differences in information content between the formats. Notably on the HAS side was the do-not-use indicator, which can be useful if a satellite is found to act anomalous. On RTCM and IGS SSR side, especially the timed components of orbit and clock corrections were non-retrievable from HAS messages. But also the HAS phase bias messages lack multiple pieces of information present in those other formats, for example satellite yaw parameters and various signal properties such as the integer and wide lane integer indicator. The use of each parameter has been studied from both the format guides and the PPP Wizard code. It was found, that for the HAS do-not-use value, on the one side, not transmitting any corrections for such a satellite, excludes it from the positioning calculations. Furthermore, setting the timed components to zero has the same effect as not using them. Lastly, as no information about the phase bias properties or yaw parameters was present in neither the messages nor the HAS ICD, these have been set to the conservative values indicating no special usability. For one last, however unlikely, special case, in which only parts of a satellite's orbit corrections are available, it has been decided to exclude this satellite's corrections entirely.

One more challenge, which was foreseen, was the updating of the RTKLIB version used by PPP Wizard. As some changes were done to the version used, these had to be made out and either included in the new version of PPP Wizard which especially concerned the reading of RTCM SSR messages. But some additional changes were set by the PPP Wizard resulting in more extensive debugging as some flags were used in different ways than intended by RTKLIB.

4.4 Outcome

Finally, the following will present the full software suite as is. The general structure from Chapter 3 still holds, but the exact interfacing and usage is presented as well. For this, Figure 7 illustrates the different levels on which the library can be utilised. There exist multiple options, again in order to allow for an easy adaptation for every use case. The most simple one is the use of the library using the separate *HAS_Decoder.py* file. Doing this, most settings can be accessed with ease via the command line. Below, an example command is shown.

```
> python3 HAS_Converter.py -s data.sbf -f RTCM -t console |
./processStream -conf conf_process.txt -rover rover.txt
```

In this case, the HAS converter reads out the SBF file *data.sbf*, and converts decoded HAS messages into the RTCM 3 format. These messages are printed to the console, which only is possible for the PPP Wizard conform textual output. Thus, the output can be directly read by the PPP Wizard's *processStream* which processes this data as defined in the *conf_process.txt* configuration file. The *rover.txt* file can optionally be used to indicate an approximate starting position, improving convergence time. Calculated PPP positions are being put out into the same console window. For this, on default, an RTKLIB-compatible format is used, making further processing of the data and high-level integration an easy task. For a more extensive documentation of the PPP Wizard, please refer to the PPP Wizard documentation included in the software package. It is to be mentioned at this point that the PPP Wizard tends to be rather sensitive to settings indicated in the configuration file. Notably is especially the model position noise which, if over-constrained, can lead to rapid changes in observed positioning errors.

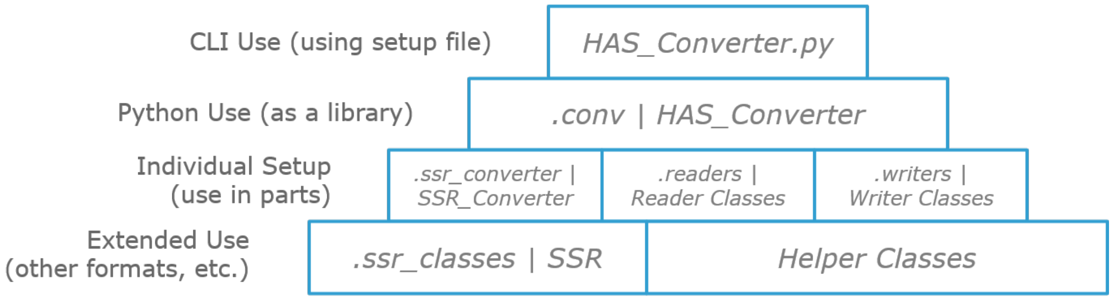


Figure 7: Decoding library: Different complexity levels of usage

The next level is the actual use of the Galileo HAS Decoder as a library that delivers all necessary functionality inside of the *HAS_Decoder* class within the *.conv* module which provides all functionality within a single class. Once imported, all settings are possible to access as parameters at the setup or the reading step. Additionally, it is to note that this also offers the possibility to process only a set number of messages or, if using live input, only to run for a set time. The documentation of the HAS Decoder Library can be found in the software package as well.

The additional levels mostly become relevant when thinking of further development or integrating parts of the library into existing software projects. Notably here are the *.readers*, *.ssr_converter* and *.writers* modules which can be used to access each functionality individually. Additionally, the *SSR* structure from the *.ssr_classes* module is a core piece, as this provides HAS correction information in the previously mentioned quasi-universal structure. This makes the addition of alternative output modes or the direct access of correction information possible.

5 Evaluation

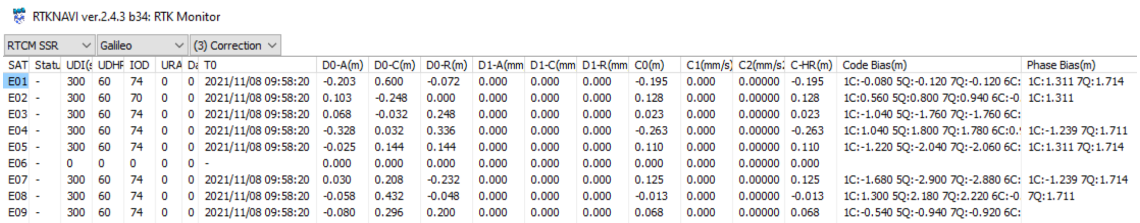
This chapter will give an evaluation of various things. Firstly, the implemented software suite will be assessed with regards to previously defined requirements. Additionally, officially stated goals will be used to analyse both the software's and the HAS's performance. Lastly, the proposed message type will be evaluated under real-use conditions, utilizing recorded correction data.

5.1 High Accuracy Service Software Suite

In the following, the Galileo HAS PPP suite's completeness and usability will be evaluated with respect to the requirements as defined in the Galileo HAS Info Note [5], and constraints of the software are described. The key requirement of the software was aiming at the real-time usability of the software. Additionally, the accuracy, convergence, and stability of the PPP library will be analysed. Alongside with this, the current state of the Galileo HAS is investigated and evaluated with respect to the same requirements.

5.1.1 Decoding and Converting

Firstly, the performance of the HAS Decoder library will be examined individually. In Figure 8, a screenshot of the RTKNavi RTCM SSR message overview is shown after reading a file of converted HAS messages. This tool is able to read RTCM 3 messages from a data stream and is used to test the overall functionality of the decoder. As can be seen, all available information is filled accordingly, proving the correct construction of messages. Phase bias data has been filled from self-constructed sample HAS messages, since in the current development state HAS messages do not contain correct phase biases yet. This step is helpful in order to be able to know causes of penitential inaccuracies encountered later on. As messages are constructed correctly, this means that the performance of the software suite is mostly determined by the Galileo HAS correction quality and the PPP Wizard's performance. As the former cannot be assessed easily, the latter is needed to be evaluated independently.



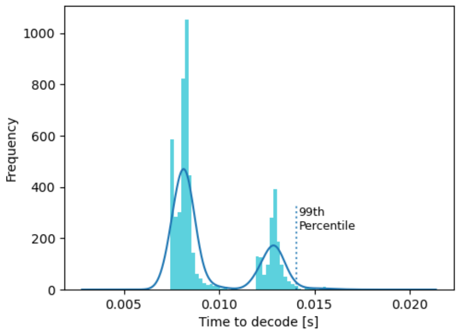
RTKNAVI ver.2.4.3 b34: RTK Monitor

RTCM SSR | Galileo | (3) Correction

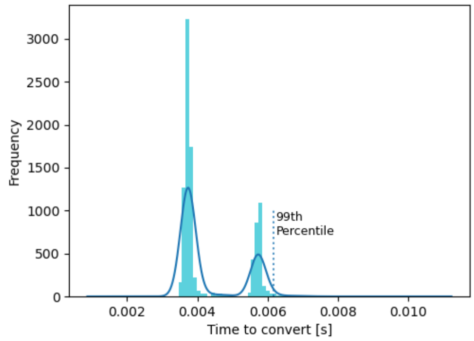
SAT	Statu	UDI(s	UDHF	IOD	URA	Da	T0	D0-A(m)	D0-C(m)	D0-R(m)	D1-A(mm	D1-C(mm	D1-R(mm	C0(m)	C1(mm/s)	C2(mm/s:	C-HR(m)	Code Bias(m)	Phase Bias(m)
E01	-	300	60	74	0	0	2021/11/08 09:58:20	-0.203	0.600	-0.072	0.000	0.000	0.000	-0.195	0.000	0.00000	-0.195	1C:-0.080 5Q:-0.120 7Q:-0.120 6C:	1C:1.311 7Q:1.714
E02	-	300	60	70	0	0	2021/11/08 09:58:20	0.103	-0.248	0.000	0.000	0.000	0.000	0.128	0.000	0.00000	0.128	1C:0.560 5Q:0.800 7Q:0.940 6C:-0.	1C:1.311
E03	-	300	60	74	0	0	2021/11/08 09:58:20	0.068	-0.032	0.248	0.000	0.000	0.000	0.023	0.000	0.00000	0.023	1C:-1.040 5Q:-1.760 7Q:-1.760 6C:	
E04	-	300	60	74	0	0	2021/11/08 09:58:20	-0.328	0.032	0.336	0.000	0.000	0.000	-0.263	0.000	0.00000	-0.263	1C:1.040 5Q:1.800 7Q:1.780 6C:0.	1C:-1.239 7Q:1.711
E05	-	300	60	74	0	0	2021/11/08 09:58:20	-0.025	0.144	0.144	0.000	0.000	0.000	0.110	0.000	0.00000	0.110	1C:-1.220 5Q:-2.040 7Q:-2.060 6C:	1C:1.311 7Q:1.714
E06	-	0	0	0	0	0	-	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00000	0.000		
E07	-	300	60	74	0	0	2021/11/08 09:58:20	0.030	0.208	-0.232	0.000	0.000	0.000	0.125	0.000	0.00000	0.125	1C:-1.680 5Q:-2.900 7Q:-2.880 6C:	1C:-1.239 7Q:1.714
E08	-	300	60	74	0	0	2021/11/08 09:58:20	-0.058	0.432	-0.048	0.000	0.000	0.000	-0.013	0.000	0.00000	-0.013	1C:1.300 5Q:2.180 7Q:2.220 6C:-0.	7Q:1.711
E09	-	300	60	74	0	0	2021/11/08 09:58:20	-0.080	0.296	0.200	0.000	0.000	0.000	0.068	0.000	0.00000	0.068	1C:-0.540 5Q:-0.940 7Q:-0.920 6C:	

Figure 8: Message conversion: RTKNAVI RTCM SSR message overview

Within the Galileo HAS decoder, the main processes are the decoding of a message and the subsequent conversion. The other parts of the software, namely reading and writing of the messages, are performed at usual rates. In Figure 9a and 9b, the processing times for these two blocks during testing on real data are plotted. It is apparent that both decoding and conversion times vary depending on the block lengths. This was expected because both the inverting of a larger matrix and converting longer message blocks should require higher processing power. In the messages analysed in the test run, two block types are notably: The shorter clock correction blocks (13 bits per satellite) and the longer orbit correction blocks (circa 46 bits per satellite).



(a) Message decoding times (per HAS message)



(b) Message conversion times (per HAS block)

Figure 9: Decoding library subroutines: Processing times

Taking into account both together with all other intermediate steps, in the processing times per full HAS message, as seen in Figure 10, the same two clusters are visible. The difference in proportions is accounted to the short and frequent clock correction messages, while the other longer blocks are combined into a more infrequent larger message. Still, as 99% of received HAS messages are processed in less than 31 ms with new messages becoming available each second, the real-time usability of the Galileo HAS decoding library is assumed to be given. Especially the more frequent and time-sensitive clock correction messages are usually being processed within less than 20 ms.

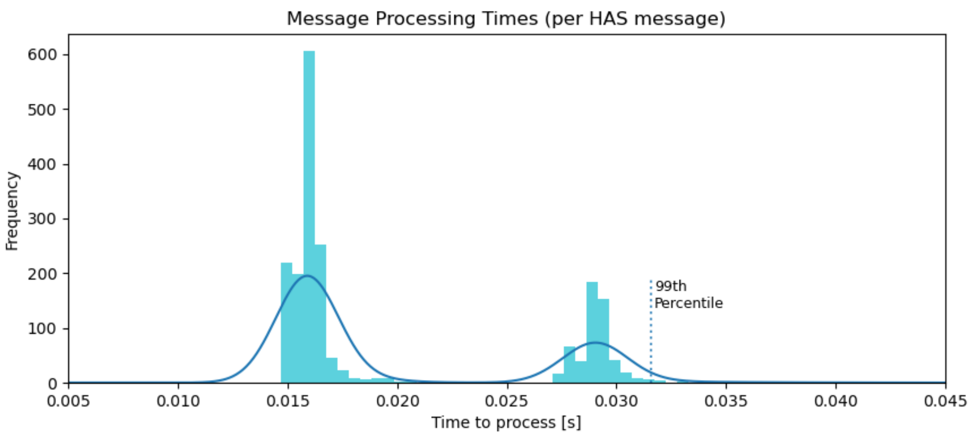
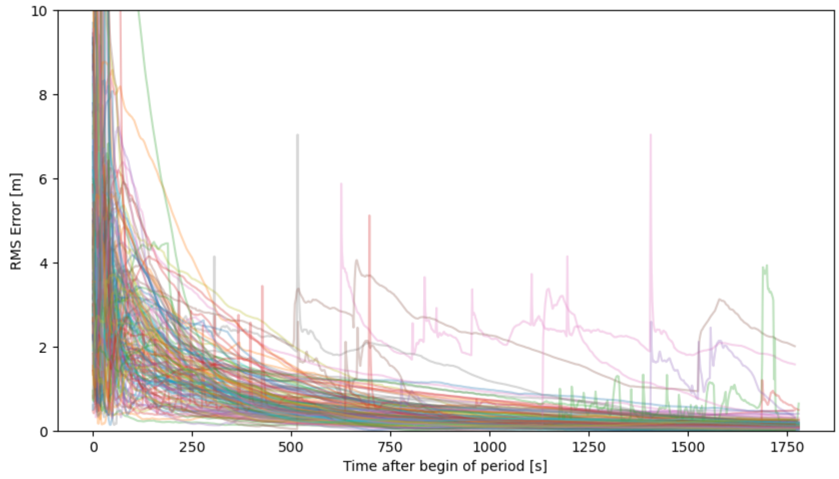


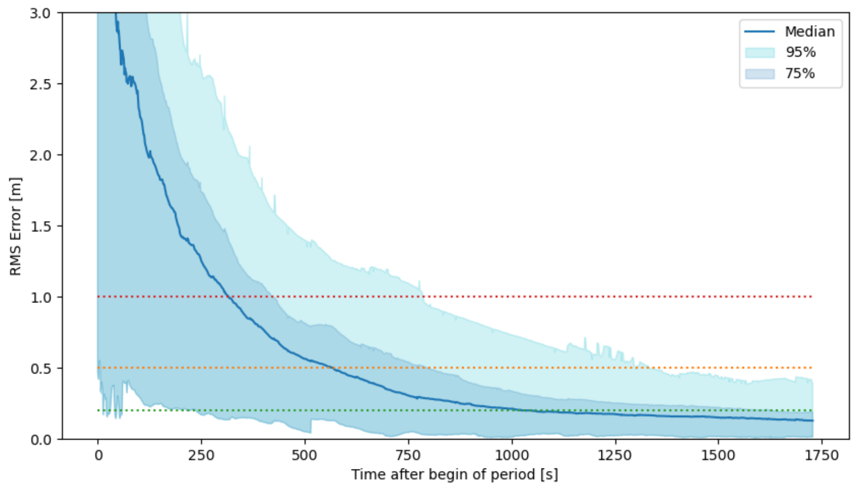
Figure 10: Full decoding library: Time-to-process per HAS message

5.1.2 High Accuracy Positioning: PPP Wizard

As the PPP Wizard software has been left largely untouched except for the replacement of the RTKLIB version, which is also an established software in the GNSS society, the real-time usability of it will be assumed to be out of question. Instead, the accuracy, convergence, and stability of calculated positions are analysed with regards to the Galileo HAS goals as presented in the HAS info note from 2020 [5], where the ultimate accuracy goal was set to 20 cm horizontally and 40 cm vertically. Due to the limited availability of Galileo HAS SIS transmissions and the development status of it, the following sub section will first decouple the testing of the PPP Wizard from the Galileo HAS, using external data provided by the IGS. On the one hand, this is to evaluate each subroutine on its own. On the other hand, this helps to further narrow down potential error sources.



(a) Individual convergence runs: Despite some noise, runs tend to converge within a few minutes



(b) Statistical analysis of convergence runs: Typically, convergence to decimeter accuracy is achieved within less than 30min

Figure 11: PPP Wizard convergence analysis using additional PPP correction data

The first test will utilise high quality GNSS data from a ground antenna in Toulouse in combination with PPP corrections calculated by the CNES in France and distributed via NTRIP over the IGS network [41]. In total, 37.5 hours have been recorded and split into observation frames of 30 minute length each with a 15 minute overlap. Figure 11a and 11b display the result of these 150 time frames of the PPP Wizard. The PPP corrections provided are known to lead to stable solutions so this integrity is given.

In combination with the PPP Wizard, it is noticeable from Figure 11a that in a few runs, some events greatly disturb the accuracy of the solutions. These events sometimes are seen more frequently due to the overlap of observation frames. However, Figure 11b shows the solutions to converge overall to an RMS error of below 50 cm within 22 min in 95% of the cases, with 75% reaching an RMS error of less than 20 cm at the end of the period.

The trend can be observed to be stable over longer periods as well, as Figure 12 shows. In this case, the observation period has been split into 26 subsets of 2 hours each with an overlap of 30 minutes. Due to some observation noise and the low sample size, the solutions within a certain time frame can be seen to be rather unstable, but in more than 75% of the trials the RMS error stayed well below 20 cm once it reached this mark.

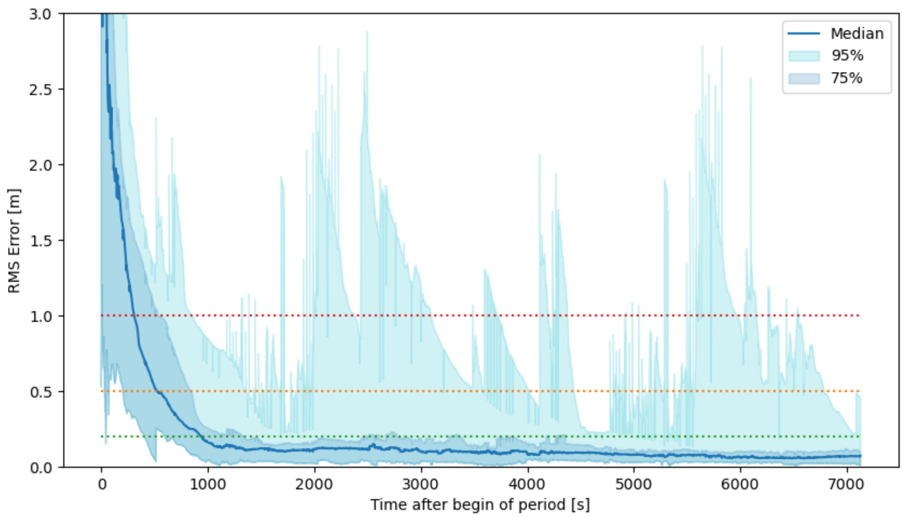


Figure 12: PPP Wizard stability analysis of 26x2h observation windows: Some interferences, but decimeter to sub-decimeter accuracy remains mostly stable

Figure 13 shows the comparison of the PPP solutions over 30 min with the same observations processed in Dual Frequency mode. These observations come from the same observation time frame. It is clear that the PPP solutions converge significantly better and more stable than dual frequency ones. Especially, it is visible that PPP converges nicely towards the true position, while the dual frequency solutions converge to some offset. This phenomena was described earlier and is due to the fact that dual frequency solutions only correct the ionospheric error. These tests have proven the PPP capability of the PPP Wizard in general, if provided with proper SSR corrections from any source. Another effect of using PPP over DF is the significant increase of computational performance, which sped up positioning greatly.

Additionally, it became apparent, that generally, the PPP Wizard performs exceptionally well when calculating dual frequency solutions. To investigate this, solutions by both the PPP Wizard and RTKLIB were analysed. Given the same observation data, PPP Wizard solutions were found to be of greatly improved stability and often showed a lower error offset than RTKLIB solutions. Parts of this improvement were accounted for by Kalman filtering done by the PPP Wizard, however, some parts of it remain unaccounted for.

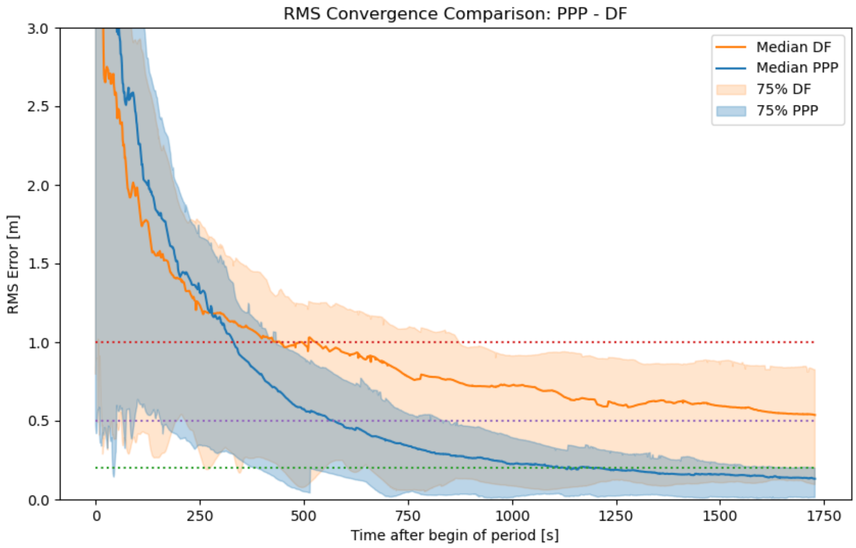


Figure 13: PPP vs DF Convergence: Accuracy improves from sub-meter to low-decimeter when using PPP corrections

5.1.3 High Accuracy Service

In the next step, the decoder and PPP Wizard are combined to evaluate both the full software suite and the Galileo HAS: On the one side, it is tested whether the PPP Wizard is generally processing converted HAS messages properly and is able to calculate PPP corrections from this. On the other side, the observed performance of the Galileo HAS is checked against the formal target accuracy stated by the program. As presented earlier, in the first phase of the HAS this is targeted to be 20cm/40cm horizontally and vertically within 300 seconds.

For this evaluation, the Signal-In-Space of HAS transmitting Galileo satellites is used together with navigation information provided by both HAS supported constellations, GPS and Galileo. The signals are recorded with an E6-band-capable antenna and receiver. The live data is recorded and split into subsets of equal length. These are fed into the Galileo HAS Decoder library, with the PPP Wizard format output directly input into the PPP processing unit. It is to mention that in this testing phase, the transmitted HAS corrections were slightly limited. They contained clock and orbit corrections for both Galileo and GPS satellites together with code biases for at least two signals per constellation. Especially the absence of phase biases is expected to impact solution accuracies. Data from three different days was used, including two different locations. The available observation windows are presented in Table 9. Recording on multiple different days is expected to reveal potential differences in performance and catch any false conclusions. These might occur due to the development status of the HAS, since different days might show differences in accuracy of the transmitted corrections. Additionally, the latitude difference between the two locations might impact the achievable accuracy of solutions in more northern areas, although the Galileo constellation has been found to perform rather well in high latitudes [42]. All recordings are, as mentioned, split into subsets of 20 or 30 minutes length with a maximum overlap of 5 minutes. This is done to increase the sample size while retaining a sufficient level of independence between each recording.

Obs. ID	Location	Date	Duration
M.1	Masala, FI	22/09/21	7.5 h
M.2	Masala, FI	29/09/21	14 h
I.1	Ispra, IT	22/09/21	24 h
I.2	Ispra, IT	27/09/21	24 h
I.3	Ispra, IT	29/09/21	24 h

Table 9: Analysed Galileo HAS recordings

Loc. ID	Location	Latitude	Longitude	Height
L.1	Masala, FI	60.16109°	24.54545°	54.216 m
L.2	Ispra, IT	45.81036°	8.62995°	278.873 m

Table 10: Antenna locations of observations

In both locations, the antenna positions are fixed and have been determined using highly accurate RTK. These coordinates are listed in Table 10 in Latitude/Longitude/Height format. This enables an easy calculation of XYZ, or ENU (East, North, Up), errors, allowing for easy comparison between data collected in different locations. To normalise time, the seconds since the beginning of a respective subset window is used.

Figure 14 displays the typical accuracy observed during the trials. This particular trial is the observation M.1, which consists of 20 minute subsets. Two relevant points are noticeable: The main convergence period of the PPP solutions is over within approximately 4 minutes with many solutions stabilizing in the first 3 minutes already. Also, it is observable that the convergence is not nearly as good as the one with full RTCM SSR data like the one provided by the IGS. This is of course expected, as the lack of timed components in the HAS reduces the accuracy between parameter updates. Additionally, the lack of phase biases can decrease the accuracy further. However, while the accuracy encountered stays behind the official HAS goal, convergence times seem to be well within the targeted performance.

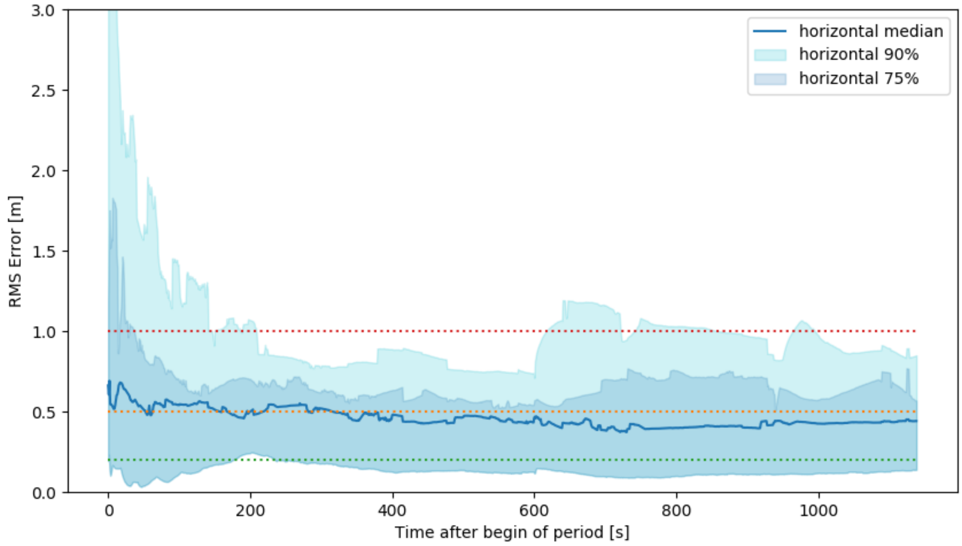


Figure 14: HAS Performance: Horizontal RMS error for observation M.1

Observations from the same day, I.1, show a similar trend. Here, the recording has been split into 60 minute subsets to analyse the performance on a longer time scale. Figure 15 shows that the accuracies of individual runs slowly converge closer to the true location. Also, it is evident that the vertical error, although being larger than the horizontal one, reaches nearly target levels after around 50 minutes.

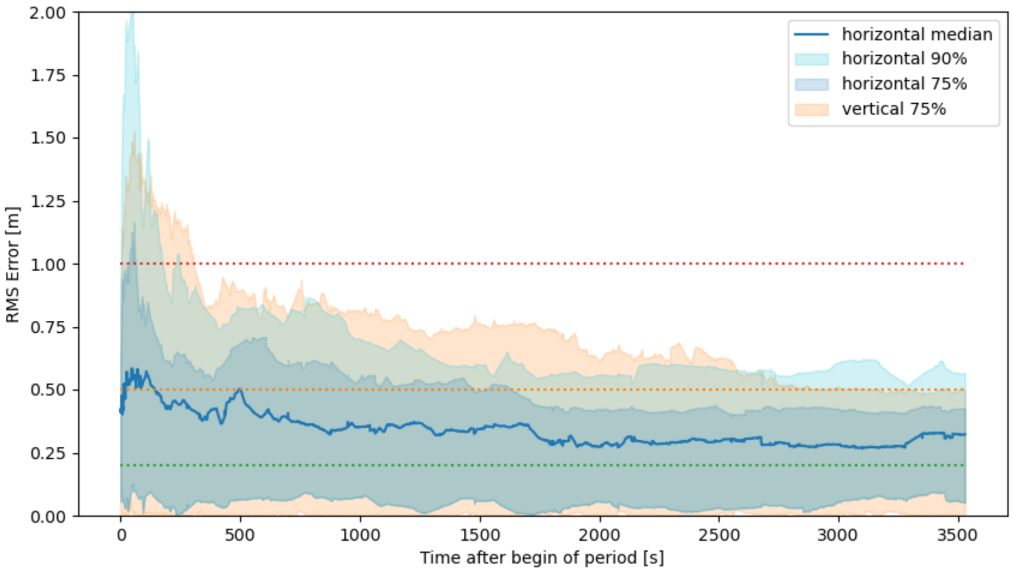


Figure 15: HAS Performance: Horizontal and vertical RMS error for observation I.1

Additionally, observation I.1 displayed a small but stable improvement from classic Dual Frequency solutions when incorporating Galileo HAS corrections. For this, the same observation data has been processed both by the PPP Wizard alone in Dual Frequency mode and by the full software suite in PPP Dual Frequency (PPP-DF) mode. As can be seen in Figure 16, solutions converge closer to the true position in the precise point positioning mode. However, the error of solutions is still close to Dual Frequency performance, which could be explained if HAS corrections are not yet on fully operational level. However, the ground-plot in Figure 17a, based on observation data from M.1, nicely visualises the effect of the HAS on positioning solutions: Compared to 17b which displays DF results for the same observations, solutions are significantly more clustered around the true position of the antenna. Additionally, convergences seem to be a lot more targeted towards it. While these tests already show the advantage of the Galileo HAS, the possible accuracy is expected to increase with growing information content of the messages.

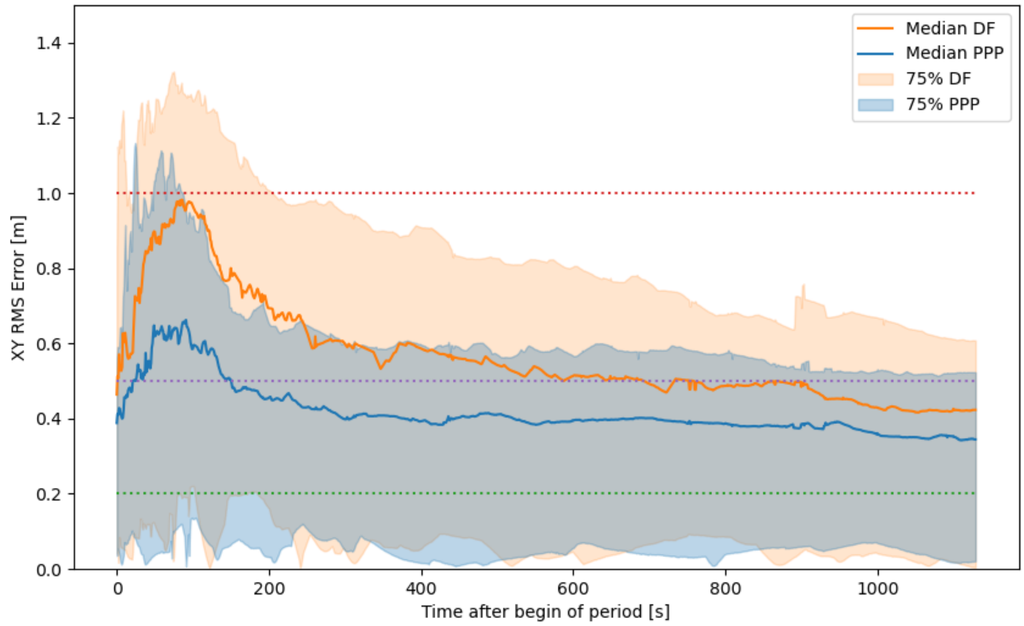
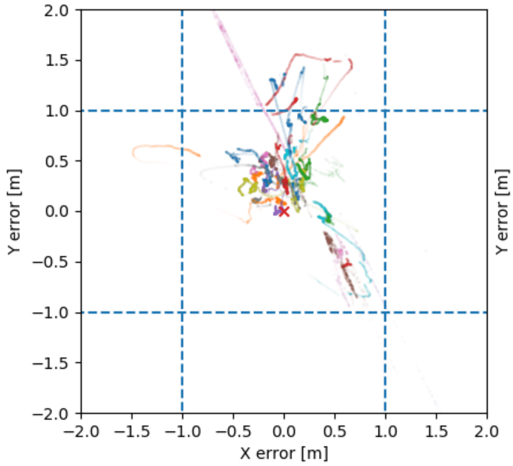


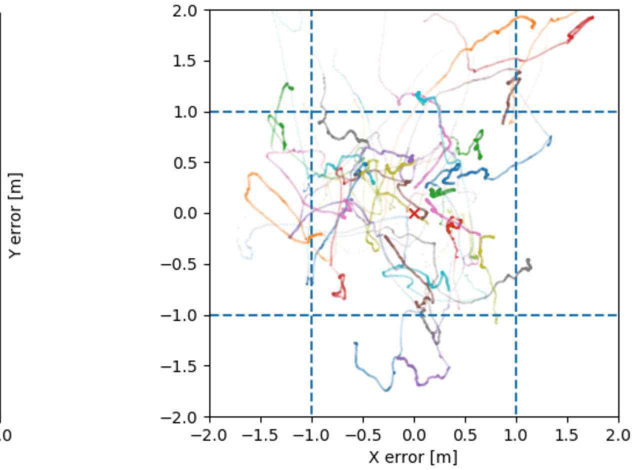
Figure 16: HAS corrected precise point positioning solutions compared to "classic" Dual Frequency for observation I.1

As expected, the performance does not vary greatly between the two locations, as is evident from Figure 18. Although it seems as if the location L.2 tends to produce slightly more accurate results, this difference is not significant. It has been hypothesised that some difference may also originate from differences in the quality of the antennae or potential obstructions in view. Generally however, some differences in performance between high-latitude and low-latitude locations are also to be expected due to the architecture of the GNSS systems.

It was possible, however, to observe a stark difference in accuracy between some days: While performance between September 22^{nd} (I.1) and September 29^{th} (I.3) was statistically not different, the observation on September 27^{th} (I.2) proved to be an anomaly of sorts: On this date, the achievable accuracy of HAS corrected positioning solutions decreased drastically, as can be seen in Figure 19. Without direct communication with the Galileo HAS responsibles, this has to be hypothesised to be due to experimental changes to correction calculations. Investigating the correction content at least showed no significant difference in the information which



(a) Galileo HAS enabled PPP



(b) Dual Frequency positioning

Figure 17: XY Groundplots of positioning solutions in Masala (L.1), "x" marking the true position

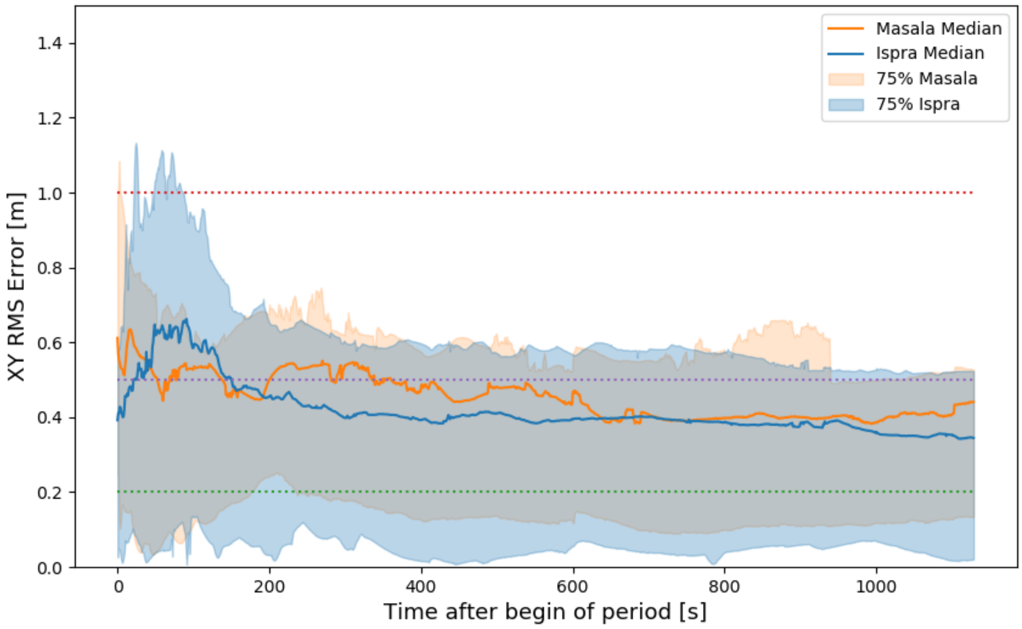


Figure 18: Latitude influence: Comparison between locations L.1 and L.2

would suggest that pieces are missing or similar. To this end, it was analysed whether the cause could be related to system-specific information differences. In particular, on days not included in this analysis, some problems with GPS correction data have occurred, with sometimes entire sets of corrections not being included. To some extent, this theory is backed by Figure 20. Here, PPP solutions for Galileo only have been compared to PPP solutions utilizing both Galileo and GPS constellations. However, this is especially true for the first 20 minutes only, with the accuracies of both converging.

In conclusion, it can be said that both the implemented software suite and the Galileo HAS seem to be operational. While the performance stays behind officially stated goals, multiple reasons for this are possible and were investigated. The most likely amongst those were, firstly, the reduced content of HAS messages currently, namely the exclusion of phase biases. Secondly, it is possible that accuracies were impacted by experimental changes to the correction calculation. Neither of these two reasons was fully verified, and it is advised to go into detail on these in the next available HAS SIS transmission session. For this, corrections provided by the service can be compared to and complimented by other correction providers like the IGS.

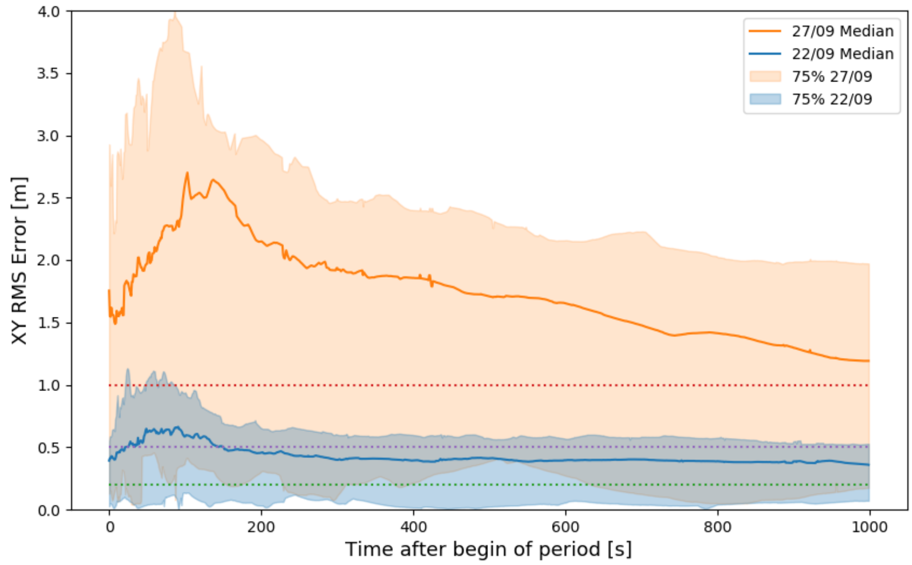


Figure 19: HAS performance compared between 22/09 (I.1) and 27/09 (I.2)

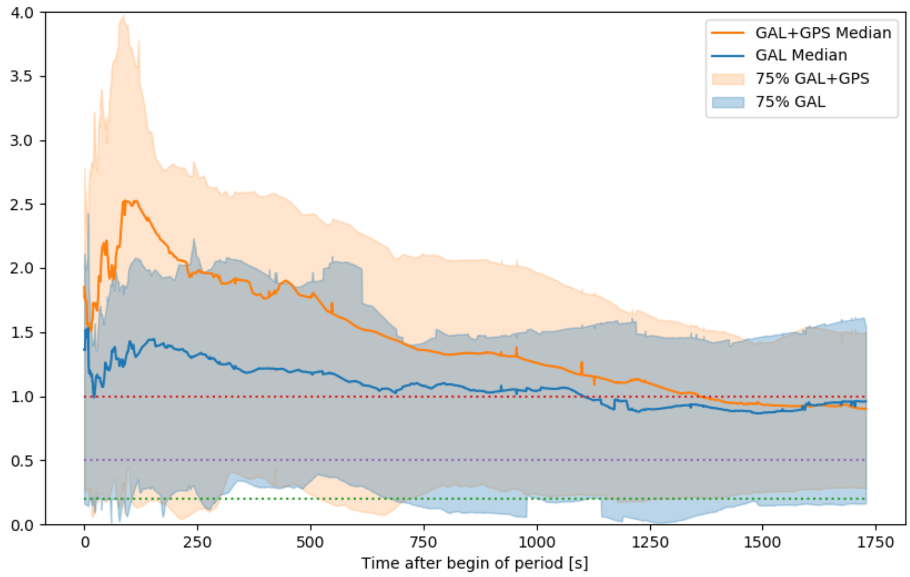


Figure 20: HAS performance 27/09 (I.2) Galileo only and GAL+GPS combined

5.1.4 Constraints

In this sub section, the known constraints of the software suite and overall evaluation phase are presented. Especially the PPP Wizard has to be discussed here. Since it was built as a demonstrator, as evident from the underlaying paper by Laurichesse and Privat [37] and Open-Source development was discontinued, the software is in parts outdated and, according to in-code documentation, not-final and convoluted. Parts of the code have been updated, enabling a PPP usage. However, because a more extensive updating of the library was out of the scope of this project, for continuous development it would be advisable to switch to another solution which is still receiving updates. RTKLIB is expected to offer such support, once the RTCM committee finalises more SSR messages.

Additionally, as explained earlier, the Galileo HAS Decoder library was built with the Open-Source context in mind. Thus, it must be said that, although real-time usability is given, this also means that the software is not tuned towards the most efficient and fastest run time possible.

Lastly, as no valid phase biases messages were transmitted during the Galileo HAS Public Observation phase, this part of the software was only tested with self-constructed data and the full HAS could not be tested with the PPP Wizard yet. Thus, the full capabilities of the HAS, also in combination with the software, could not be tested. Specifically speaking, this means that all HAS corrected results could only be produced in PPP-DF mode, and not using the significantly more accurate PPP-AR. In the future, this is expected to further increase accuracy and stability.

5.2 Proposed Message Format

In Section 3.4, a new message type for the HAS was proposed, essentially carrying the same information as the initial one. However, this new MT2 was designed with future extensions to the HAS in mind, which implies a need to compress the information accordingly. In the following, this potential will be evaluated and discussed.

For testing the proposal, HAS messages from three different days will be analysed. The days chosen for this are July 12^{th}, September 22^{th} and September 29^{th}. The recorded time frames are 1 h, 6 h, and 6 h, respectively. Both September recordings correspond to subsets of the previously defined M.1 and I.3 observations. This is done in order to cancel out any atypical data which might be observed if only one day is taken into account. Each transmitted message will be read out and the size of a message of the proposed type is calculated. This message is to mirror the information content exactly, and, thus, can be understood to be the result of a lossless conversion. As the goal of the proposed message type is to save bandwidth when transmitting larger amounts of correction data, the clock subset correction block is disregarded here.

Below, Figure 21 depicts the results of the aforementioned tests. The graph contains the size reduction of both clock and orbit correction blocks when transmitted in the proposed MT2 message structure instead of the original MT1 one. It can easily be seen that the proposed message, under typical conditions, is often able to reduce message sizes by 20% or more. Especially for clock corrections, the saved bandwidth usually reaches 28%. However, the potential for orbit corrections is not to be overlooked, as these are usually comprised of much more information. For example, while the mean clock correction block encountered covered a mere 697 bits, the mean orbit correction block contained over 2450 bits. Thus, an average MT2 encoded clock correction message will only use just over 500 bits to transmit the same information. For orbit corrections, this number of bits will be around 2000 bits.

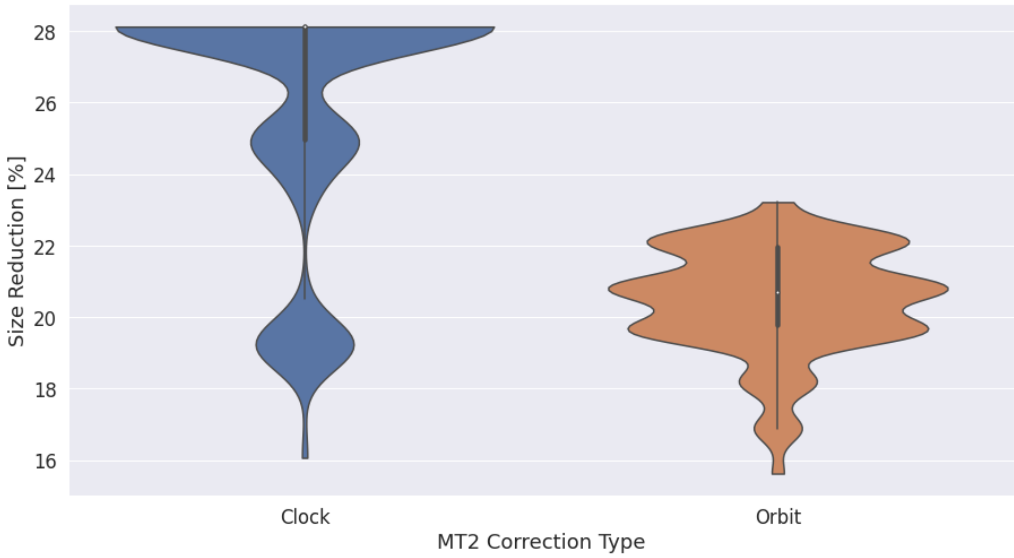


Figure 21: MT2 Bandwidth Saving Potential: Saved portion using the proposed message type 2 compared to the classic message type 1

The most important aspect in terms of size reduction of these messages, however, is the message size in pages. Especially clock corrections, requiring a higher update rate than orbit corrections, are transmitted alone and more frequent. As a single HAS page is 424 bits long, this means that the typical clock correction message is transmitted in two pages, which can also not be reduced using the proposed MT2. However, one can extrapolate the current size of the messages to include a third constellation: Then, a clock correction message of MT1 would require three pages instead, whereas two pages still suffice if transmitted in the proposed MT2 structure.

On the other side, the orbit corrections, as discussed, are transmitted using many more bits than clock corrections. Thus, the proposed structure can already bring down the required pages for Galileo and GPS corrections by at least a full page. This also reveals the potential if the correction block is paired with other traditional blocks, as the full compression potential could be utilised. Hence, it would be advised to combine the proposed message blocks with the traditional blocks to complete the MT2. Table A1 in Appendix A has been adapted to reflect this. However, as the structure of the other blocks has been left untouched, the message specification for these blocks is presented in short form only. For the exact layout of these, one should refer to the Galileo HAS ICD [25]. Table 11 presents the same three cases as earlier in Table 7, together with the required Galileo satellites in-view. This time, however, the adapted MT2 structure is used, reducing the message sizes of clock and orbit corrections. As the compression rate is not always consistent, in some cases realistic page ranges are used.

Case	Required Pages	Required Galileo satellites in view
1: Status Quo	2 p/s clk correction 13-14 p/ 7 s other corrections	4 satellites
2: More Content	2 p/s clk correction 15-16 p/ 7 s other corrections	5 satellites
3: 3rd Constellation	2 p/s clk correction 23-24 p / 7 s other corrections	6 satellites

Table 11: Galileo HAS Satellite Requirement: Proposed message type 2 used

From this, the advantage of the proposed message structure is clear: In both the current case and the prospected case, including an additional constellation like GLONASS, the required amount of Galileo satellites in-view can be reduced. And even generally, reducing the amount of required pages helps in mitigating the risk posed by corrupt messages.

6 Conclusion

This last chapter will first draw a conclusion on the thesis work with respect to the stated goal. After this, a reflection will point out relevant shortcomings and open questions and give advice for future work.

6.1 Summary

While the main activity within the project was the implementation of a software capable of decoding and converting Galileo High Accuracy Service messages, the underlying goal was to increase its accessibility and usability from day one. To this end, this thesis presents and offers two main aids:

First, the aforementioned software suite has been implemented, taking into account the main design requirements of the GNSS community. This will enable potential users to easily access the Galileo HAS as soon as it launches in 2022. Once released, the software will enable potential users to easily integrate it into existing solutions and at the very least evaluate the benefits of the HAS. Due to the tool being designed for an Open-Source release, continuous development and adaptation will be made easy and possible for anyone interested in the Galileo HAS. A first presentation with stakeholders has received positive feedback regarding the impact of this tool. Additionally, the availability of the tool will help the GSA to reach a broader user base, which ultimately will also help with establishing credibility.

Secondly, an addition to the set of messages was proposed which makes use of the structure of correction data more efficiently. Whilst already now offering an improvement to the availability of the service, in the long run it could help the Galileo HAS to ensure consistent availability when scaling up its capabilities.

6.2 Future Work

However, some issues remain to be addressed in future work. The following paragraph first addresses ideas to be performed in research and development on the software. Continuing, advice for official groups such as the working group behind the Galileo HAS is given.

Firstly, both the in-development status of the service and the raised insufficiencies of the PPP Wizard should be noted as potential room for future work. It is generally advised to remodel the software suite to use another PPP tool, as the work performed could not fully eliminate all issues regarding the PPP Wizard. Nonetheless, if due to the quality of solutions delivered by it or the lack of alternatives, it could otherwise be sufficient to establish a working relationship with the PPP Wizard developers or at the very least perform an extensive code analysis to address open questions. Also, updates to the HAS message structure are possible at this stage and will require updating of the decoder library.

Additionally, extending the variety of output formats for the converter could improve the impact of the software even further. It has been discussed that the Sapcordia SPARTN format could be a valuable addition, since the major reason of its exclusion, namely the unavailability of Galileo messages, has already been solved over the course of this work.

As discussed, it can be beneficial in the next HAS SIS transmission session to analyse HAS corrections and compare them to other service providers like the IGS. This will provide further insight into the causes of the raised performance issues of the HAS.

At the same time, the HAS group is advised to work closely together with other evaluation project groups, as the targeted accuracy could not be reached in this phase of the service. It will be beneficial to analyse as early as possible whether the cause for this was mainly the development status of the HAS or whether some changes are needed. At the same time, an in-depth analysis of the proposed message type might reveal further possibilities for an improved set of messages.

Lastly, more efforts in the direction of a finalisation of RTCM 3 SSR are required. Perhaps, pressure by organisations such as the Galileo programme could lead to developments in this area. This would be a large step towards the general usability of precise point positioning algorithms. Especially current Open-Source projects such as RTKLIB are, understandably, yet to implement full support, which can only be expected once the standard is final. Then, due to the growing demand for high accuracy positioning solutions combined with the increasing accessibility through projects such as the HAS, general PPP usage could see a surge.

References

[1] The Aerospace Corporation, *Brief History of GPS*, en. [Online]. Available: https://aerospace.org/article/brief-history-gps (visited on 10/27/2021).

[2] D. Gebre-Egziabher and S. Gleason, *GNSS Applications and Methods*, en. Artech House, 2009, Google-Books-ID: juXAE3SHQroC, ISBN: 978-1-59693-330-9.

[3] Inside GNSS, *SpaceX ISS Mission Relies on GPS Technology*, en-US, May 2012. [Online]. Available: https://insidegnss.com/spacex-iss-mission-relies-on-gps-technology/ (visited on 12/31/2021).

[4] European Commission, *COMMISSION IMPLEMENTING DECISION (EU) 2018/ 321 - of 2 March 2018 - amending Implementing Decision (EU) 2017/ 224 setting out the technical and operational specifications allowing the commercial service offered by the system established under the Galileo programme to fulfil the function referred to in Article 2 (4)(c) of Regulation (EU) No 1285 / 2013 of the European Parliament and of the Council*, en, Mar. 2018. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32018D0321&from=EN.

[5] European GNSS Agency, *Galileo HAS Info Note*, 2020. [Online]. Available: https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_HAS_Info_Note.pdf (visited on 03/23/2021).

[6] National Coordination Office for Space-Based Positioning, Navigation, and Timing, *GPS.gov: Timing Applications*. [Online]. Available: https://www.gps.gov/applications/timing/ (visited on 10/27/2021).

[7] European Space Agency, *Hyper-accurate clocks – the beating heart of Galileo*, en. [Online]. Available: https://www.esa.int/Applications/Navigation/Hyper-accurate_clocks_the_beating_heart_of_Galileo (visited on 10/28/2021).

[8] J. M. Cunha and K. LoPiccalo, "Case: The Global Positioning System (GPS)," en, NAVAL POSTGRADUATE SCHOOL MONTEREY CA GRADUATE SCHOOL OF BUSINESS and PUBLIC POLICY, Tech. Rep., May 2014, Section: Technical Reports. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA603841 (visited on 10/27/2021).

[9] F. van Diggelen and P. Enge, "The World's first GPS MOOC and Worldwide Laboratory using Smartphones," en, ISSN: 2331-5954, Sep. 2015, pp. 361–369. [Online]. Available: http://www.ion.org/publications/abstract.cfm?jp=p&articleID=13079 (visited on 10/28/2021).

[10] J. D. Cardall and R. S. Cnossen, "ITC 81," San Diego, California, Oct. 1981, pp. 1593–1605.

[11] B. W. Parkinson and J. J. Spilker, "Differential GPS," en, in *Progress In Astronautics and Aeronautics: Global Positioning System: Theory and Applications*, Google-Books-ID: t0eGFpSwN0wC, AIAA, 1996, pp. 3–50, ISBN: 978-1-60086-420-9.

[12] G. H. Millman, "A Study of Ionospheric Winds and Turbulence Utilizing Long Radio Waves," de, Ph.D. dissertation, Pennsylvania State College, Jun. 1952. [Online]. Available: https://www.proquest.com/openview/c8e9f9f993888b930d3ca11c235cfae7/1?pq-origsite=gscholar&cbl=18750&diss=y (visited on 12/24/2021).

[13] U. Robustelli, V. Baiocchi, and G. Pugliano, "Assessment of Dual Frequency GNSS Observations from a Xiaomi Mi 8 Android Smartphone and Positioning Performance Analysis," en, *Electronics*, vol. 8, no. 1, p. 91, Jan. 2019, Number: 1 Publisher: Multidisciplinary Digital Publishing Institute. DOI: 10.3390/electronics8010091. [Online]. Available: https://www.mdpi.com/2079-9292/8/1/91 (visited on 12/24/2021).

[14] O. Mubarak and A. Dempster, "Analysis of early late phase in single-and dual-frequency GPS receivers for multipath detection," *GPS Solutions*, vol. 14, pp. 381–388, Sep. 2010. DOI: 10.1007/s10291-010-0162-z.

[15] D. S. Choy, *GNSS Precise Point Positioning*, en, Kuala Lumpur, 2014. [Online]. Available: http://www.fig.net/resources/proceedings/fig_proceedings/fig2014/ppt/TS05A/Choy.pdf.

[16] R. B. Langley, "Gpsworld.september98.pdf," English, *GPS World*, vol. September 1998, p. 5, Sep. 1998. [Online]. Available: http://www2.unb.ca/gge/Resources/gpsworld.september98.pdf (visited on 10/28/2021).

[17] A. Cina, P. Dabove, A. M. Manzino, and M. Piras, *Network Real Time Kinematic (NRTK) Positioning – Description, Architectures and Performances*, en. IntechOpen, Mar. 2015, Publication Title: Satellite Positioning - Methods, Models and Applications, ISBN: 978-953-51-1738-4. DOI: 10.5772/59083. [Online]. Available: https://www.intechopen.com/chapters/47449 (visited on 10/28/2021).

[18] G. Wübbena, M. Schmitz, and A. Bagge, "PPP-RTK: Precise Point Positioning Using State-Space Representation in RTK Networks," Sep. 2005.

[19] European GNSS Agency, *PPP-RTK Market and Technology Report*, 2019. [Online]. Available: https://www.gsa.europa.eu/sites/default/files/calls_for_proposals/rd.03_-_ppp-rtk_market_and_technology_report.pdf (visited on 03/31/2021).

[20] Y. Heo, T. Yan, S. Lim, and C. Rizos, *International Standard GNSS Real-Time Data Formats and Protocols*, 2009.

[21] A. Engfeldt, "Network RTK in Northern and Central Europe," Tech. Rep., Mar. 2005. [Online]. Available: https://www.lantmateriet.se/globalassets/kartor-och-geografisk-information/gps-och-geodetisk-matning/rapporter/lmv-rapport_2005_05.pdf (visited on 12/26/2021).

[22] I. Fernandez-Hernandez, G. Vecchione, F. Díaz-Pulido, *et al.*, "Galileo High Accuracy: A Programme and Policy Perspective," Oct. 2018.

[23] S. Vana, J. Aggrey, S. Bisnath, R. Leandro, L. Urquhart, and P. Gonzalez, "Analysis of GNSS correction data standards for the automotive market," en, *Navigation*, vol. 66, no. 3, pp. 577–592, Sep. 2019, ISSN: 0028-1522, 2161-4296. DOI: 10.1002/navi.323. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/navi.323 (visited on 03/24/2021).

[24] R. Hirokawa and I. Fernandez-Hernandez, "Open Format Specifications for PPP/PPP-RTK Services: Overview and Interoperability Assessment," Sep. 2020. DOI: 10.33012/2020.17620.

[25] European GNSS Agency, *Galileo HAS SIS ICD 1.4*, 2021.

[26] D. Borio, T. Senni, and I. Fernandez-Hernandez, "Experimental Analysis of a Candidate Galileo E6-B Data Dissemination Scheme," Feb. 2020, pp. 509–520. DOI: 10.33012/2020.17159.

[27] I. Fernandez-Hernandez, T. Senni, D. Borio, and G. Vecchione, "High-parity vertical Reed-Solomon codes for long GNSS high-accuracy messages," *Navigation*, vol. 67, Mar. 2020. DOI: 10.1002/navi.357.

[28] K. Ferguson, "RTCM Background and SC-134 Evolution," en, *RTCM Paper 086-2018-SC134-005*, p. 10, Jun. 2018. [Online]. Available: http://rtcm.info/sc134/086-2018-SC134-005.pdf (visited on 10/25/2021).

[29] RTCM SC-104, *RTCM 10403.3, Differential GNSS (Global Navigation Satellite Systems) Services - Version 3 + Amendment 1 (April 28, 2020)*, Apr. 2020.

[30] G. Wübbena, "RTCM State Space Representation (SSR) Overall Concepts Towards PPP-RTK," en, *PPP-RTK & Open Standards Symposium*, vol. March 12-13, 2012, p. 41, 2012.

[31] International GNSS Service, *IGS SSR Message Format*, Oct. 2020. [Online]. Available: https://files.igs.org/pub/data/format/igs_ssr_v1.pdf (visited on 03/23/2021).

[32] Quasi Zenith Satellite System Services Inc., *QZSS / CSSR Interface Specification*, Nov. 2018. [Online]. Available: https://qzss.go.jp/en/technical/download/pdf/ps-is-qzss/is-qzss-l6-001.pdf (visited on 03/26/2021).

[33] *SPARTN: Secure Position Augmentation for Real Time Navigation*, en-US. [Online]. Available: https://www.spartnformat.org/ (visited on 10/27/2021).

[34] Sapcorda Services GmbH, *SPARTN Interface Control Document 2.0.1*, Sep. 2021. [Online]. Available: https://www.spartnformat.org/wp-content/uploads/210928_SPARTN_v2.0.1.pdf (visited on 10/27/2021).

[35] T. Takasu, *RTKLIB: An Open Source Program Package for GNSS Positioning.* [Online]. Available: http://www.rtklib.com/ (visited on 10/29/2021).

[36] rtklibexplorer, *Rtklibexplorer/RTKLIB*, original-date: 2016-03-11T01:48:08Z, Oct. 2021. [Online]. Available: https://github.com/rtklibexplorer/RTKLIB (visited on 10/29/2021).

[37] D. Laurichesse and A. Privat, "An Open-source PPP Client Implementation for the CNES PPP-WIZARD Demonstrator," en, p. 10, 2015.

[38] C. Chen and G. Chang, "PPPLib: An open-source software for precise point positioning using GPS, BeiDou, Galileo, GLONASS, and QZSS with multi-frequency observations," en, *GPS Solutions*, vol. 25, no. 1, p. 18, Nov. 2020, ISSN: 1521-1886. DOI: 10.1007/s10291-020-01052-4. [Online]. Available: https://doi.org/10.1007/s10291-020-01052-4 (visited on 11/16/2021).

[39] Sapcorda Services GmbH, *SPARTN Interface Control Document v1.8.0*, Jan. 2020. [Online]. Available: https://www.spartnformat.org/wp-content/uploads/200124_SPARTN_v1.8.0.pdf (visited on 03/24/2021).

[40] W. Zhang, E. Cannon, O. Julien, and P. Alves, "Investigation of Combined GPS/GALILEO Cascading Ambiguity Resolution Schemes," *Proceedings of ION GPS/GNSS*, Jan. 2003.

[41] G. Weber, D. Dettmering, and H. Gebhard, "Networked Transport of RTCM via Internet Protocol (NTRIP)," in *Sanso F. (Ed.): A Window on the Future*, ser. Symposia Series, vol. 128, Sapporo, Japan: Springer Verlag, 2005, pp. 60–64.

[42] European GNSS Agency, *Galileo in high latitudes and harsh environments: A success story*, en, May 2020. [Online]. Available: https://www.euspa.europa.eu/newsroom/news/galileo-high-latitudes-and-harsh-environments-success-story (visited on 12/29/2021).

A HAS Message Type 2 Proposal

Name	Range	Bits	SF	Unit	Description
TOH	0-3599	12	1	s	Time of hour, based of GST
Mask Flag	-	1	-	-	3 flag bits, defining the classic MT1 blocks to be present ("1") or not ("0").
Code Bias Flag	-	1	-	-	
Phase Bias Flag	-	1	-	-	
Orbit Corr. Flag	-	1	-	-	2 flag bits, defining the new MT2 blocks to be present or not in the message.
Clock Full-set Flag	-	1	-	-	
Reserved	-	4	-	-	
Mask ID	0-31	5	-	-	Mask ID, identifying mask received in MT1
IOD Set ID	0-31	5	-	-	IOD set ID, identifying the IOD set in case of no transmitted orbit corrections

Table A1: Proposed MT2 Header

Name	Range	Bits	SF	Unit	Description
MASK					
Please refer to the Galileo HAS ICD [25]					
CODE BIASES					
Please refer to the Galileo HAS ICD [25]					
PHASE BIASES					
Please refer to the Galileo HAS ICD [25]					
ORBIT CORRECTIONS					
Validity Interval Idx	0-15	4	-	-	Validity Interval HAS [25]
Delta Radial Offset (GNSS ID 1)	±10.2375	13*[1]	0.0025	m	Delta radial offset value for all satellites of GNSS ID 1.
Delta Radial Individual Correction Bits (GNSS ID 1)	1-16	4	1	bits	How many bits are used for individual satellite radial corrections of GNSS ID 1.
Delta In-Track Offset (GNSS ID 1)	±16.376	12*	0.008	m	Delta in-track offset value for all satellites of GNSS ID 1.

[1]Parameters so indicated are two's complement, with the sign bit (+ or -) occupying the MSB.

Name	Range	Bits	SF	Unit	Description
Delta In-Track Individual Correction Bits (GNSS ID 1)	1-16	4	1	bits	How many bits are used for individual satellite in-track corrections of GNSS ID 1.
Delta Cross-Track Offset (GNSS ID 1)	±16.376	12*	0.008	m	Delta cross-track offset value for all satellites of GNSS ID 1.
Delta Cross-Track Individual Correction Bits (GNSS ID 1)	1-16	4	1	bits	How many bits are used for individual satellite cross-track corrections of GNSS ID 1.
...					
Delta Radial Offset (GNSS ID N_{sys})	±10.2375	13*	0.0025	m	See GNSS ID 1 description.
Delta Radial Individual Correction Bits (GNSS ID N_{sys})	1-16	4	1	bits	
Delta In-Track Offset (GNSS ID N_{sys})	±16.376	12*	0.008	m	
Delta In-Track Individual Correction Bits (GNSS ID N_{sys})	1-16	4	1	bits	
Delta Cross-Track Offset (GNSS ID N_{sys})	±16.376	12*	0.008	m	
Delta Cross-Track Individual Correction Bits (GNSS ID N_{sys})	1-16	4	1	bits	
GNSS IOD[2] (SV 1)	0-1023 or 0-255	var.	1	-	10 bits for Galileo; 8 bits for GPS
Delta Radial Mod. (SV 1)	var.	var.*	0.0025	m	Modifier to add to the corresponding GNSS correction offset. "b10..0" indicates data not available.
Delta In-Track Mod. (SV 1)	var.	var.*	0.008	m	
Delta Cross-Track Mod. (SV 1)	var.	var.*	0.008	m	
...					
GNSS IOD (SV_{nsat}[3])	0-1023 or 0-255	var.	1	-	See SV 1 description.

[2]IOD refers to IOD_{nav} for Galileo satellites and IODE/IODC for GPS satellites.

[3]SV_{nsat} denotes the last satellite corrections are provided for (as defined by the used mask), including all systems. Satellites are grouped by systems, with the last satellite of system 1 immediately being followed by the first one of system 2.

Name	Range	Bits	SF	Unit	Description
Delta Radial Mod. (SV$_{nsat}$)	±10.2375	13*	0.0025	m	See SV 1 description.
Delta In-Track Mod. (SV$_{nsat}$)	±16.376	12*	0.008	m	
Delta Cross-Track Mod. (SV$_{nsat}$)	±16.376	12*	0.008	m	
CLOCK CORRECTIONS					
Delta Clock C0 Offset (GNSS ID 1)	±10.2375	13*	0.0025	m	Delta Clock offset value for all satellites of GNSS ID 1.
Delta Clock C0 Individual Correction Bits (GNSS ID 1)	1-16	4	1	bits	How many bits are used for individual satellite clock corrections of GNSS ID 1.
Delta Clock C0 Multiplier (GNSS ID 1)	1-4	2	-	-	Multiplier for all Delta Clock C0 corrections of GNSS ID 1 (offset **and** modifier). "b00": x1 "b01": x2 "b10": x3 "b11": x4
...					
Delta Clock C0 Offset (GNSS ID N_{sys})	±10.2375	13*	0.0025	m	See GNSS ID 1 description.
Delta Clock C0 Individual Correction Bits (GNSS ID N_{sys})	1-16	4	1	bits	
Delta Clock C0 Multiplier (GNSS ID 1)	1-4	2	-	-	
Delta Clock C0 Mod. (SV 1)	var.	var.	0.0025	m	Delta Clock C0 modifier for SV 1. "b10..0" indicates data not available. "b01..1" indicates the satellite shall not be used.
...					
Delta Clock C0 Mod. (SV$_{nsat}$)	var.	var.	0.0025	m	See SV 1 description.

Table A2: Proposed MT2 Body