Vilniaus Universiteto
Fizikos fakulteto
Taikomosios elektrodinamikos ir telekomunikacijų institutas

Karolis Stankevičius

DAUGIAJUOSČIŲ GNSS SIGNALŲ ĮRAŠYMO IR ATKŪRIMO TYRIMAS NAUDOJANT PROGRAMUOJAMO RADIJO ĮRANGĄ

Magistratūtos studijų baigiamasis darbas
Elektronikos ir telekomunikacijų technologijų studijų programa

Studentas	Karolis Stankevičius
Leista ginti	2023-05-24
Darbo vadovas	doc. Rimvydas Aleksiejūnas
Recenzentas	doc. Vytautas Jonkus
Instituto direktorius	prof. Robertas Grigalaitis

Vilnius 2023

Vilnius University
Faculty of Physics
Institute of Applied Electrodynamics and Telecommunications

Karolis Stankevičius

ANALYSIS OF MULTI-BAND GNSS SIGNAL RECORDING AND PLAYBACK USING SOFTWARE-DEFINED RADIO

Master Thesis
Electronics and Telecommunication Technologies Programme

Student	Karolis Stankevičius
Allowed to defend	2023-05-24
Academic supervisor	assoc. prof. PhD. Rimvydas Aleksiejūnas
Reviewer	assoc. prof. PhD. Vytautas Jonkus
Director of the Institute	prof. PhD. Robertas Grigalaitis

Vilnius 2023

Contents

1 Introduction

In 2022, the number of connected Internet of Things (IoT) devices has reached 14,4 billion globally and is projected only to grow further [1]. Some of these devices, in order to perform their functions, have a necessity to know their global position. This, obviously, is solved by integrating Global Navigation Satellite System (GNSS) receiver into IoT device. However, GNSS signals are very week, since transmit power varies from 20 W to 265 W [2] and orbit height varies from 20000 km to 23000 km [3]. At the receiver signal power can be as low as -163 dBW, which is bellow noise floor [4]. What, also, does not help, is the fact that in urban areas GNSS signals degrade significantly due to multipath caused by buildings [5]. For these reasons, before product can be manufactured and sold at scale, it must be thoroughly tested to determine it's GNSS capabilities.

To test an IoT device, one may use a GNSS simulator, which can record and replay GNSS signals. An example of such device could be LabSat 3 Wideband, which is capable of recording and replaying up to 20 simultaneous GNSS signals in multiple bands, for example, L1 and L5 [6]. However, in general, GNSS simulators are expensive and there is not much to choose from. Therefore, company which creates GNSS enabled devices, has to find a good enough justification to make an investment into such device, since all it will be able to do is simulate GNSS signals. The high price of GNSS simulators is reason enough for companies to explore alternatives, like simply driving around the city with a prototype and analyzing logs, but this is far from ideal.

Taking everything into account, there is a genuine need for GNSS simulation solution, which would be cheaper and would not be reliant on prototypes of unknown capabilities. Currently, there are Software Defined Radio (SDR) based solutions like Software-Defined GPS Signal Simulator, which can simulate only GPS L1 C/A signals [7], or GNSS record-replay systems described in [8, 9, 10], which mostly perform simulations only of GPS L1 C/A signals ([9] is an exception), use hardware, which in total will cost as much if not more than commercial solutions, are mostly not portable ([10] is an exception) and are almost 10 years old.

As of writing of this master thesis, there is not a single SDR based GNSS record–replay simulation solution, which is capable of recording and replaying all L1 and L5 GNSS signals, is relatively cheap and portable. Therefore, the goal of this work is to create a GNSS L1+L5 record–replay simulator, which can simulate L1 and L5 signals, and is affordable and portable.

Part of results produced during this work has been submitted for publication "Low–cost SDR–based GNSS signal recorder and replayer for equipment testing" to scientific journal titled Journal of Navigation, which is curated by University of Cambridge.

2 Overview of Literature

2.1 GNSS signals

Currently, there are 6 different GNSS constellations, which can be used independently, and more than 20 different types of GNSS signals. All signals are transmitted between 1164 MHz and 1610 MHz and the aggregate of all constellations and signals constitute what is known as global navigation satellite system. GNSS RF spectrum is illustrated in figure 1. The spectra was generated using GNSS-matlab library [11]. Most of civilian signals are located in L1 and L5 bands and signals of these bands are used in IoT as well. Therefore, for the purposes of this work, only signals in L1 and L5 band will be discussed further.

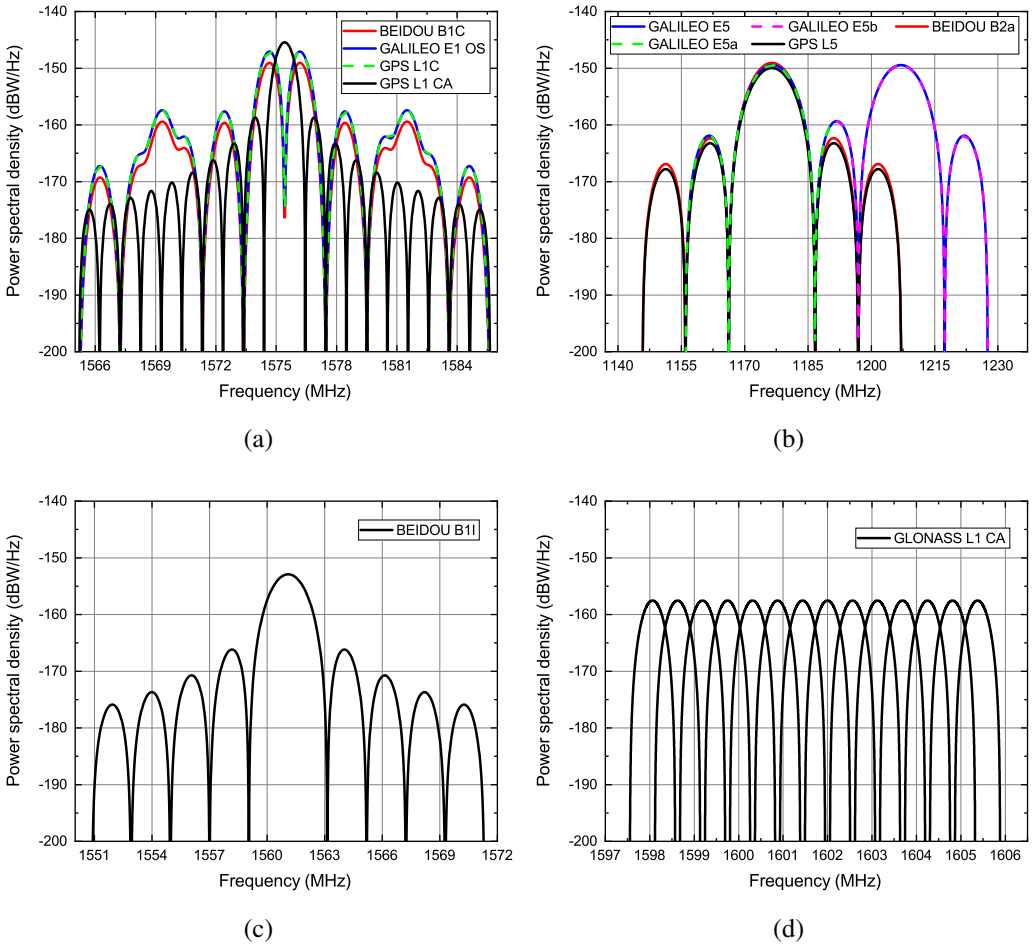


Figure 1: GNSS signal spectra: L1 (a), E5/L5 (b), B1I (c) and GLONASS L1 (d).

Main characteristics of GNSS signals located in L1 and L5 bands can be found in [12, 13, 14, 15, 16, 4, 17, 18]. However, there are some specific details that need to be addressed for successful record–replay operation. One of such specific characteristic is related to GPS L1C, GALILEO E1 and BEIDOU B1C. All of these signals are rather complex, because they use TMBOC(6,1,1/11),

CBOC(6,1,1/11) and QMBOC(6,1,3/44) modulations, respectively. These modulations are similar because essentially they are different possible combinations of binary offset carrier BOC(1,1) and BOC(6,1) modulations, which allow distributing signal power along data and pilot components of the signal [19, 20]. Frequency spectra of GPS L1C, GALILEO E1 and BEIDOU B1C signals are shown in figure 1a.

Another signal, which needs more attention, is GLONASS L1 C/A, because unlike other GNSS systems GLONASS used FDMA multiple access technique[17]. This means that to transmit GNSS data GLONASS uses 14 channels and each channel has its center frequency defined by formula:

$$f_{cp} = 1602 + 0.5625p \text{ MHz}, \tag{1}$$

where p is channel number in the interval $-7 \leq p \leq 6$. Each channels' width is only 0.511 MHz and all 14 channels need 7.3125 MHz of bandwidth as shown in figure 1d.

The last signal that needs to be discuses is GALILEO E5a. This signal is part GALILEO E5 signal, which uses altBOC(15,10) modulation. The spectrum of E5 signal is shown in Fig. 1c together with GPS L5 signal. From figure, it is clear that E5 main lobes are very similar to GPS L5 binary phase shift keying BPSK(10) modulation's main lobe. This was done by design, so that both E5 lobes can be received and processed independently as E5a and E5b. Receivers can even demodulate E5a as BPSK(10) signal [21].

GNSS signals, that were not discussed in detail, do not have any special characteristics, that need to be known before recording GNSS signals. It is enough to know center frequency and signal bandwidth to successfully record these signals. However, note, that GNSS signals are week and are below noise floor [22]. This means that if one may try to find these signals with spectrum analyzer, analyzer would show nothing at GNSS center frequencies. Therefore, it is important to use appropriate gain parameters in SDR to be able to record and replay GNSS signals.

2.2 White noise approximation of GNSS signals

Since GNSS signals are very weak and at the receiver signal power can be as low as -163 dBW [4], recorded signals are dominated by noise. Thermal noise power (N_{th}) at the receiver can be estimated by the formula (2) [22], where k - Boltzmann's constant, T - temperature in K and B - bandwidth in Hz.

$$N_{th} = kTB \tag{2}$$

Reception of GNSS signals is only possible due to spreading codes and requirement for noise floor to not be higher than -100 dBm [12]. In case of GPS L1 C/A signal, which uses BPSK(1)

modulation, 50 bit/s rate data signal is spread using gold spreading code, which has chip rate of 1.023 Mchips/s. Ratio of spreading code rate to data signal rate is called spreading gain, which for this GNSS signal is 46 dB. Lowest guaranteed GPS L1 C/A signal power is -130 dBm [12, 23]. Therefore, before despreading signal-to-noise ratio (SNR) was -30 dB after despreading it becomes 13 dB.

However, when SDR records GNSS signals no despreading is performed and simply raw signal is recorded with SNR as low as -30 dB. These signals should be the same as regular white Gaussian noise. This is a good property, because no additional noise source is needed to perform phase alignment.

SDR records signals in terms of IQ samples. Signal reconstructed from IQ samples will be complex white noise, where I samples are real part and Q samples are imaginary part. This signal can be analyzed in terms of probability density functions (PDF) of real part, imaginary part, amplitude and phase.

Real and imaginary parts of complex white noise both follow Gaussian distribution [24]. Also, real and imaginary parts have the same variance. There are 2 possible cases: when means of real and imaginary parts are equal to zero and when means are not equal to zero. Generally, non-zero means of real and imaginary parts do not need to be equal as well.

When means are equal to zero, amplitude of white noise is distributed according to Rayleigh distribution and phase is distributed according to uniform distribution respectively. When real and imaginary parts of complex noise have non-zero means, amplitude follows Rician distribution. However, phase does not have compact closed form, but clusters around phase of complex mean. Spread around this value is determined by total variance of complex white noise [24].

2.3 Signal phase description using circular statistics

For GNSS records analysis statistical approach will be necessary. However, for phase linear statistics cannot be used, because phase is periodic. Linear statistics can only be used for amplitude and IQ values.

There are significant differences in how basic descriptive statistics and how probability density functions are defined for cyclic data [25, 26, 27]. Suppose we have n points of cyclical data $\{\theta_i : i = 1, ...n\}$ and want to find circular mean $\bar{\theta}$. First we perform polar to rectangular transformation for all θ_i values using

$$A = \sum_{i=1}^{n} \sin(\omega\theta_i) \text{ and } B = \sum_{i=1}^{n} \cos(\omega\theta_i) \tag{3}$$

where A and B are the projections of resultant vector and ω is a scaling factor which scales θ_i to $[0, 2\pi)$. The direction of this resultant vector is the circular mean $\bar{\theta}$, which using projections is defined as

$$\bar{\theta} = \begin{cases} \omega^{-1} \arctan(A/B) & \text{if } B \geq 0, \\ \omega^{-1}(\arctan(A/B) + \pi) & \text{if } B < 0. \end{cases} \tag{4}$$

Another measure, mean resultant length $\bar{R}$ is defined by

$$\bar{R} = \frac{1}{n}(A^2 + B^2)^{\frac{1}{2}}. \tag{5}$$

$\bar{R}$ represent spread of data. The higher the $\bar{R}$, the less spread exist in data and all values are clustered around $\bar{\theta}$. circular mean and mean resultant length are the angular and amplitude components of the first uncentred trigonometric moment. By calculating moments relative to mean direction $\bar{\theta}$, centered trigonometric moments can be obtained:

$$\rho = n^{-1} \sum_{i=1}^{n} \cos(\omega(\theta_i - \bar{\theta})) = \bar{R}, \tag{6}$$

$$\rho_2 = n^{-1} \sum_{i=1}^{n} \cos(2\omega(\theta_i - \bar{\theta})). \tag{7}$$

From first ρ and second ρ_2 centered moments another measure of spread, circular dispersion can be formed, which is defined by

$$\hat{\delta} = \frac{1 - \rho_2}{2\bar{R}^2}. \tag{8}$$

Since data is distributed on the circumference of the circle, we can have two possible (and relevant to this works) extremes. When data is distributed uniformly around the whole circumference, we have uniform probability distribution function defined as

$$\Psi_{uniform}(\theta) = \frac{1}{2\pi}. \tag{9}$$

This function has no parameters and is independent from input data. When plotted in polar coordinates it is basically a circle around the origin. The other relevant extreme, when data distribution has a peak on the circumference of the circle. In this case, data distribution can be described by von Mises distribution, which is considered the Gaussian distribution for directional data. The probability density function of θ, distributed by von Mises distribution, is given by [25, 27]

$$\Psi_{von\ mises}(\theta|\mu, \kappa) = \frac{1}{2\pi\omega \mathrm{I}_0(\kappa)} \exp(\kappa \cos(\omega(\theta - \mu))) \tag{10}$$

where μ is the mean value of θ, κ is the shape parameter relates to the spread of data and $\mathrm{I}_0(\kappa)$ (the modified Bessel function of the first kind of order 0) is a normalization constant to ensure that PDF integrates to one. The von Mises distribution tends toward Gaussian distribution when κ values are large, meaning that data is closely clustered around mean value, and approaches uniform distribution when κ approaches zero. Given directional data $\{\theta_i : i = 1, ...n\}$, parameters μ and κ can be estimated by

$$\begin{cases} \mu = \bar{\theta} \\ \kappa = A_1^{-1}(\frac{1}{n}\sum_{i=1}^{n}\cos(\omega(\theta_i - \bar{\theta}))) \end{cases} \tag{11}$$

where A_1^{-1} is a function, which definition can be found in [25] or [27].

For directional data, besides descriptive statistics and distribution models, hypothesis test can be used as well. One of the tests, which is relevant to this work, is Watson test [28], which allows to determine if a set of von Mises distributions are statistically the same. This test can be utilized to determine if samples collected in different time intervals have the same distribution in statistical sense [29].

Watson test assumes that sample distribution in each time interval is the von Mises distribution and all intervals have a common κ. Under this assumption hypothesis that all intervals have identical circular means μ is tested. The test statistic of Watson test is defined by

$$Y_r = 2(N - R_p)/\hat{\delta}_0, \tag{12}$$

where

$$R_p = \left[\left(\sum_{i=1}^{r} n_i \cos(\omega\mu_i) \right)^2 + \left(\sum_{i=1}^{r} n_i \sin(\omega\mu_i) \right)^2 \right]^{\frac{1}{2}} \tag{13}$$

and

$$\hat{\delta}_0 = \sum_{i=1}^{r} n_i \hat{\delta}_i / N \tag{14}$$

where N is the total number of samples, n_i is the number of samples in a specific time interval i, r is the number of time intervals, μ_i is the circular mean of samples in interval i and $\hat{\delta}_i$ is the circular dispersion of samples in interval i. Watson statistic Y_r is compared against a χ^2 distribution with $r-1$ degrees of freedom. If p-value is less than 0.05, there is statistical evidence that hypothesis that all time intervals have the same circular mean is incorrect.

2.4 Software-defined radio and synchronization

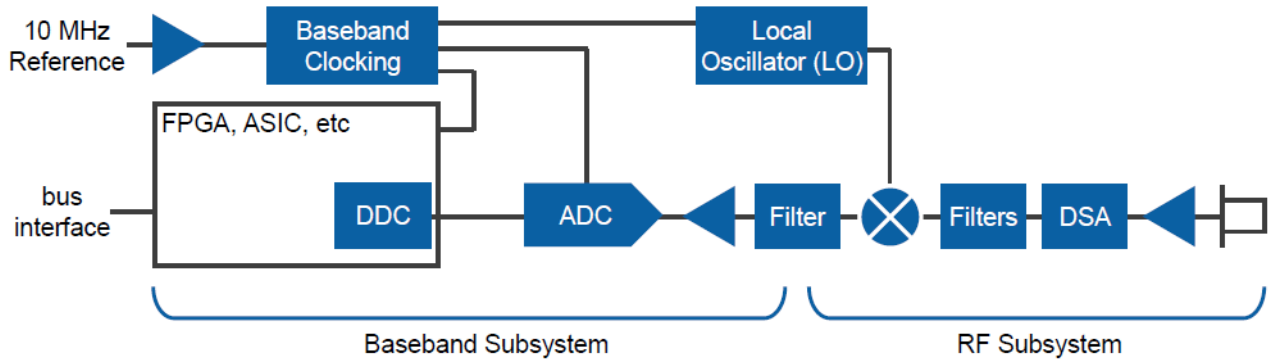


Figure 2: Standard SDR structure [30].

Software defined radios are usually systems in which all signal processing is performed in software. They are composed of analog front-end, which is a configurable analog transceiver, and analog-to-digital/digital-to-analog converter [31]. The front-end converts a specified part of spectrum to lower frequency band, where it can be converted to digital domain. The digital receiver then performs all the necessary digital signal processing. Therefore, SDR systems provide a great degree of flexibility. Block diagram of the typical SDR device is provided in figure 2.

SDR system can record any part of spectrum, which is within its operation limits and GNSS signals are no exception. However, to successfully record these signals more than one SDR is needed, because there are 4 center frequencies around, which L1 and L5 signal are located. This means that 4 separate SDR devices must work as phase-coherent transceiver, which simultaneously and synchronously receive or transmit signals on aforementioned 4 center frequencies. The need for such operation arise from the fact that signals coming from the same satellite, which is the case for L1 and L5, are synchronized, their code delay is about the same, since both signal traveled the same path, relative speed is the same and Doppler are proportional with a known factor [32]. Also, GNSS receivers manufacturers can utilize acquisition information gathered by acquiring L1 for L5 acquisition as an optimization and cost-cutting measure, which leads to operation mode where without L1 signals and their synchronization to L5 signals, L5 signals cannot be detected.

To realize phase-coherent transceiver from separate SDR, time, frequency and phase synchronization in hardware and software must be realized. Usually, SDR systems have 10 MHz frequency reference signal input and output, as well as, 1 pulse per second (1PPS) signal input for time synchronization, however synchronization using these inputs and outputs without additional efforts is not sufficient for application, which require channels coherence [31].

There are three possibles ways to distribute 10 MHz reference signal to the whole system. In

reference clock sharing topology, common reference signal is distributed to all SDRs and every SDR using its own phase lock loop (PLL) to produce its own local oscillator (LO) signal [33]. In practice this approach produces $\sim$ tens of degrees of phase drift due to reference signal drift and $\sim$ several degrees of phase drift due to noise caused by PLL [34].

Another possible approach is daisy chain topology, when reference signal is used only for first SDR in the chain and after PLL generates that SDR's LO signal, LO signal is used internally, but, as well, exported to the next SDR in chain, which in turn exports this signal to subsequent SDR [33]. This is a simpler and commonly used approach, because one can expect several degrees of phase drift per daisy chain stage and several decimal pars of degree of phase drift due to LO output and input circuits [34].

The last possible topology is star distribution, where first SDR's LO signal exported to all remaining SDRs via splitter. This topology is superior to the previous two, because it provides a stable phase relationship between all devices in the system, since phase noise in all channels becomes correlated in time [33]. Such configuration is used in large or phase sensitive applications [34].

After reference signal distribution is done, time synchronization must be addressed. Without it, previous actions do not make significant changes, because recorded basebands are misaligned in time. Time synchronization can be achieved by using 1PPS input if SDR has it. Using this signal all SDRs in the system can have a synchronous start and stop, also, internal components of each SDR can be triggered synchronously system–wide. Synchronous start and stop is sufficient to solve time misalignment [30, 33].

However, not all SDRs have 1PPS inputs. In such case, time alignment can be done in software. A noise source has to be connected to each SDRs' RF input and, by calculating cross-correlation between signal streams, time and phase alignment can be achieved [35]. Alignment is done by shifting cross-correlation functions' maximums' to the same time value. The same time shifts can be applied to recorded baseband signals.

Finally, after reference distribution and time alignment is done, phase alignment has to be done. Even if phase drift is minimized and stable phase relation between SDRs is established, phase differences between SDRs in the beginning are unknown and random due to PLL behavior and distribution equipment [33, 35]. Phase alignment can be done in software and must be done every time SDRs are started. For phase alignment noise source is used as well. Phase-correction coefficient α_n is calculated for all SDRs 1 to N using formula:

$$\alpha_n = \frac{\langle x_n{}^*, r \rangle}{|\langle x_n{}^*, r \rangle|}, \tag{15}$$

where r - reference SDR's baseband values. Coefficient α_n should be calculated over a few frames to reduce noise [35]. After correction coefficients are applied, it is assumed that SDRs are coherent until next restart. Also, note, that noise source should not be connected after alignment procedures are completed, because it will significantly reduce signal-to-noise ratios [35].

2.5 Host computer configuration for SDR applications

Usually, software defined radio is connected to a host computer via USB port. USB ports are administered by USB controller, which is part of computer's Platform Control Hub (PCH) also known as chipset and is shown in Figure 3. This hub contains controllers for all I/O interfaces and is connected to processor via Direct Media Interface in Intel based systems [36]. Additionally, Modern CPUs can be directly connected to graphics cards or solid state drives (SSD) via PCI Express.

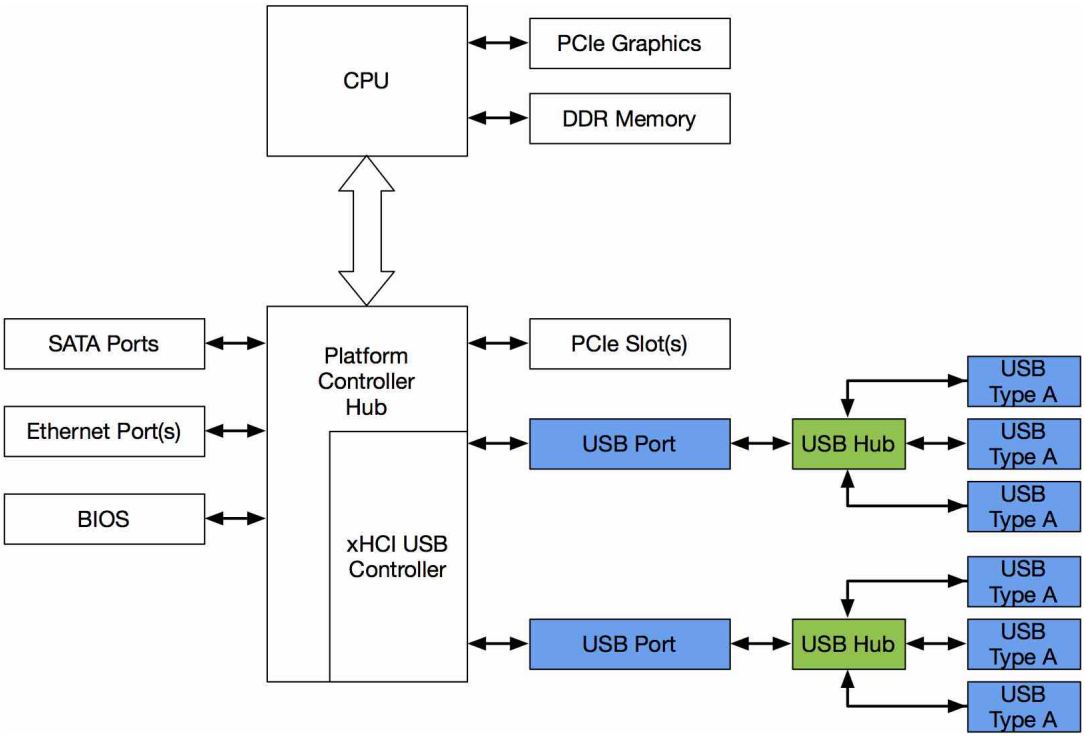


Figure 3: Typical PC architecture [37].

Assume, SDR is recording specific part of RF spectrum and recorded baseband samples are stored in SSD, which is directly connected to CPU and contains operating system and program files. In this scenario data generated by SDR has to travel through USB controller, Platform Control Hub and CPU until it reaches SSD.

Usually, computer performs many different concurrent task, which all require hardware resources. SDR generates data and sends it to host computer. This way SDR creates pressure on the host system. Other devices and tasks, which require the same hardware resources create back pressure

on SDR. Existence of back pressure creates overflow, which leads to data loss, since SDR generates data at constant rate. If this rate can not be sustained in the whole path from SDR to SSD data is lost. To mitigate this problem buffers are needed on SDR side, however size of buffers is small in low cost SDRs. In case of HackRF One SDR, which is used in this work, it is only 32 KB, which is less than 1 millisecond of data at maximum sampling of 20 Msps [38].

The situation becomes even worse than host computer has to interact with multiple SDRs simultaneously, which is exactly the case in GNSS simulations because up to 4 SDRs are needed. At 20 Msps this creates a constant data rate of 160 MB/s, which originates from 4 simultaneous sources. This may not look a lot, but since HackRF One uses USB 2.0, 160 MB/s is close to 240 MB/s theoretical maximum of USB 2.0 over 4 independent ports.

Due to limitations of USB, in multi–SDR system the most likely culprit of data loss will be USB controller. To solve this issue one may use USB PCI Express card with multiple USB controllers like this one [39]. The use of 3rd party card would allow to omit platform control hub and hardware path between SDR and SSD would be shorter (only USB card and CPU), however SSD may become a bottleneck as well.

Depending on NAND flash used in SSD and an over all construction of the solid state drive different sustained performance can be observed. Kingston A2000 sustained performance drops over time and average speed is only 480 MB/s [40], while Samsung 970 Pro maintains more or less constant performance with average speed of 2091 MB/s [41]. This difference is decided by the fact that Kingston A2000 has SLC buffer and the rest of the drive is made from TLC flash [42], while in Samsung 970 Pro case, whole drive is made from MLC flash [43]. SLC flash stores 1 bit per cell, MLC stores 2 bits per cell and TLC stores 3 bits per cell. The more bits per cell the higher the data density and the larger possible drive at the lower cost. However, the fewer bits per cell the better drives performance and higher the endurance of the drive. Therefore, Samsung 970 Pro is better for GNSS record–replay.

Sustained write performance is crucial to GNSS record–replay operation, because while recording GNSS signals, constant data load is put on SSD for long periods of time. If SSD can not maintain its write speed data loss may occur dues to lack of SDR buffers.

Overall, it is clear that a buffer is necessary for both SDR and SSD. As mentioned before, on SDR side buffer is small and not much can be done, however on host computer buffer can be created using random access memory (RAM) and software like RapidDisk [44]. RapidDisk allows to dynamically allocate part of RAM as cache for SSD. This way data from SDR can be written to RAM first and only after to SSD in a write–through manner.

In addition, the whole system should be configured as a real–time (RT) system via operating

system (OS) and BIOS. RT systems prioritizes latency over throughput. Also, they are more deterministic and can guarantee that a task will be performed until the deadline [45, 46]. Moreover, processes can be prioritized, meaning that SDR operations can be set to highest priority.

OS capable of RT is Linux and is used in RT systems by NASA, US Air Force Research Lab and SpaceX [45]. However, mainstream Linux need to be configured as RT OS. This is done by patching kernel with PREEMPT_RT patch or using a custom kernel like Xanmod RT-kernel [47]. Note, not all software supports RT-kernels and custom software may be needed. Also, in BIOS, system power management, hyper–threading, boosting and other feature, which introduce variability of runtime and responsiveness should be disabled [45].

Overall, RT system running RT Linux together with 3rd party USB card with multiple USB controllers, high performance SSD and RAM cache would be the most stable and reliable system for sustained SDR record–replay operations.

2.6 Classification using machine learning

GNSS records produced in this work can be classified in to two types: type 1, which requires initialization to be performed once, and type 2, which requires initialization to be performed twice. Without appropriate initialization only L1 signals are detected by receiver. Before recording, it is impossible to determine what type of record will be produced. This behavior is unnatural and should not exist if only GNSS signals are considered. Experimentation suggests that this behavior is caused by SDRs and to some extent by time synchronization, but internal processes, which are at fault, are currently unknown.

The solution to this unknown behavior is to consider system of SDRs as a black box and from a data set of GNSS records build machine learning model, which would classify the records into types based on features of the records. Features for a record can be generated using statistical analysis of records themselves. There are multiple machine learning classification algorithms, which can be used for this problem [48], but only random forest classifier (RFC) and support vector classifier (SVC) will be discussed in this work.

Classification using random forest algorithm consists of a collection of decision trees classifiers $\{h(\mathbf{x}, \Theta_k), k = 1, ...\}$ where Θ_k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\mathbf{x}$ [50]. This algorithm is visualized in Fig. 4. Basically, training data set is sub-sampled using bootstrap sampling, which means new data sets are sampled from original data with replacements (same values can appear in multiple new data sets). Then, from each new data set, a decision tree is built using random set of features. Finally, all trees

make a vote on which class data suppose to belong to. Majority vote is considered the final class.

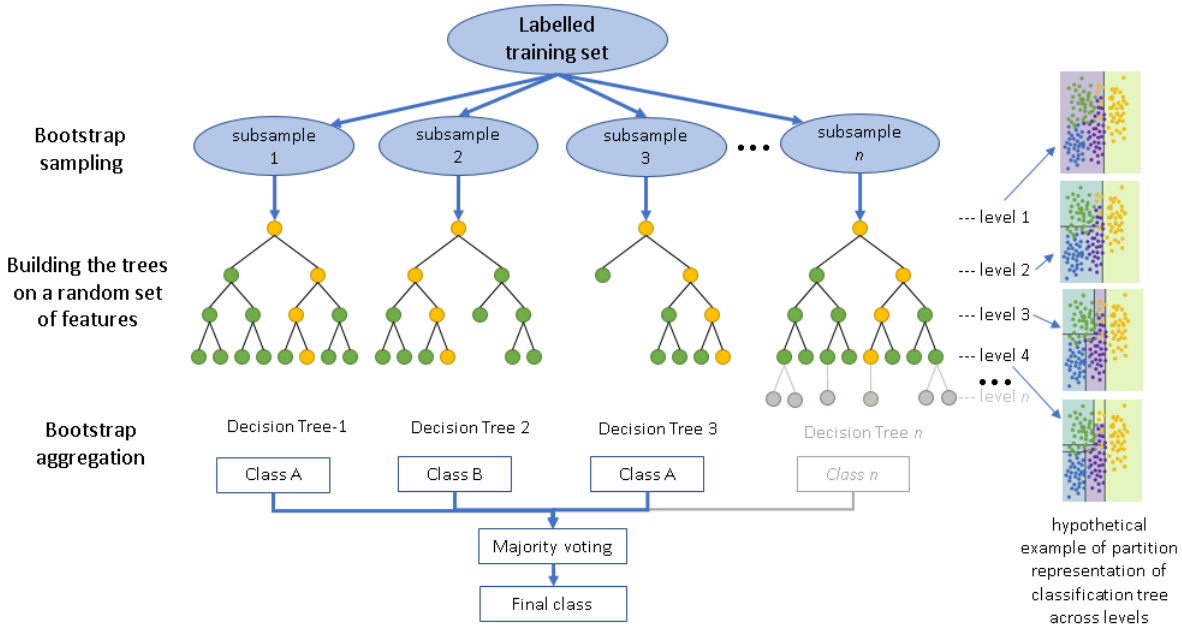


Figure 4: Simplified schematization of random forest classifier [49].

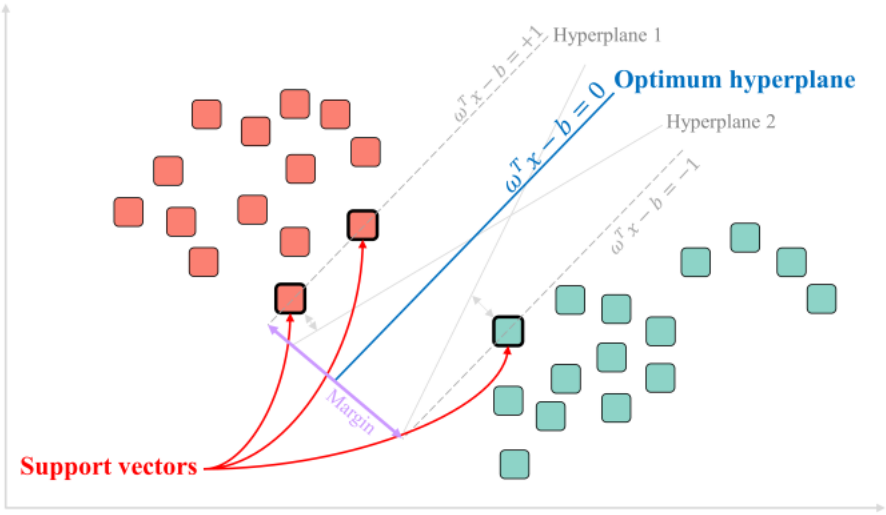


Figure 5: An SVM example for linearly separable data [48].

Classification using support vector classifier algorithm is illustrated in figure 5. This algorithm determines the separation boundary called hyperplane, which divides data into discrete classes. To maximize separation, data samples closest to the hyperplane are used as support vectors, since they are the hardest to classify and directly impact the location of the boundary [48]. In the simplest case hyperplane is a line, however, this limits the application of the algorithm only to linearly separable data. For overlapping or non-linear data, initial data can be mapped to higher dimensional space using

kernel function, in which samples will become linearly separable [51].

Both RFC and SVC have relatively high and comparable performance in terms of accuracy, however, SVC is better at solving high dimansionality and limited training data problems, while RFC is easier to use and can learn both simple and complex classification functions. Also, which algorithm is better depends on the data and situation in which it is used, because there are conflicting published results [48].

To train machine learning models an appropriate feature set is necessary. Even if statistical analysis will generate large set of features, it will not be known which features are more important than the other. Also, large number of features can over–fit the model, and it will be useless then used on new data. Small number of features can lead to the opposite problem - under–fit - model will not make accurate predictions. There are two possible solutions for this: feature selection and feature extraction [52].

Feature selection is simple. We train our model on all features. Then evaluate the significance of each feature, which can be easily done using random forest classifier [53]. Features, which have low significance, are removed and a new reduced list of features is used to train the model anew. The issue is that by performing feature selection, we are losing the variation of data, which can lead to underfitting.

Alternative is feature extraction, which performs mathematical transformations on original features, creating a new representation of data in extracted coordinate system. The main model to perform feature extraction is principal component analysis (PCA) [52].

The essence of PCA is to reduce the dimensionality of a data set, which has many non-independent variables, while retaining as much as possible of variation, which is present in data. This is achieved by creating a new set of uncorrelated variables, also known as, principal components (PC). PCs are ordered so that first ones will retain most of the variation present in all of original variables [52, 54].

Reduced dimensionality is also useful for data analysis and pattern identification, because data represented in terms of principal components can emphasize similarities and differences, which are not visible in original data [54]. 2D or 3D plots of principal components can display significant patterns and clustering in data, which was not evident or possible to display using original variables.

To evaluate model performance and improvement caused by feature selection or extraction, appropriate model must be validated using K–fold cross–validation. This algorithm works in several steps [52]. First, data is sliced in equal K slices, where K is usually 3, 5 or 10. Second, for each fold of cross–validation, K-1 of the sections is treated as training set and the remaining section is treated as test set. Third, for remaining folds different arraignment of training and test sections is used.

Fourth, Performance metrics are calculated for each fold of cross–validation. Finally, the average of calculated metrics is computed.

K–fold cross–validation produces a more accurate out of sample prediction error than a single 70–30 train–test split, because it performs several independent train–test splits and averages the results. Use of multiple independent splits means that utilization of the data set is more efficient and all data is used for testing and training instead of specific part as in single train–test split. However, K–fold cross–validation efficiency comes at the expense of computational resources. 10–fold cross–validation is 10 time more expensive computationally than a single train-test split.

3 Experimental Setup

All necessary equipment and interconnections to record and replay GNSS signals are shown in block diagram, in Fig. 6. This diagram represents the proposed GNSS L1+L5 record–replay simulator and can be split into two parts along main GNSS L1+L5 antenna Beitian BT-300S. All equipment to the left of this antenna are necessary for successful record-replay operation, while everything on the right side is for validation. Physical record–replay equipment is shown in Fig. 7.

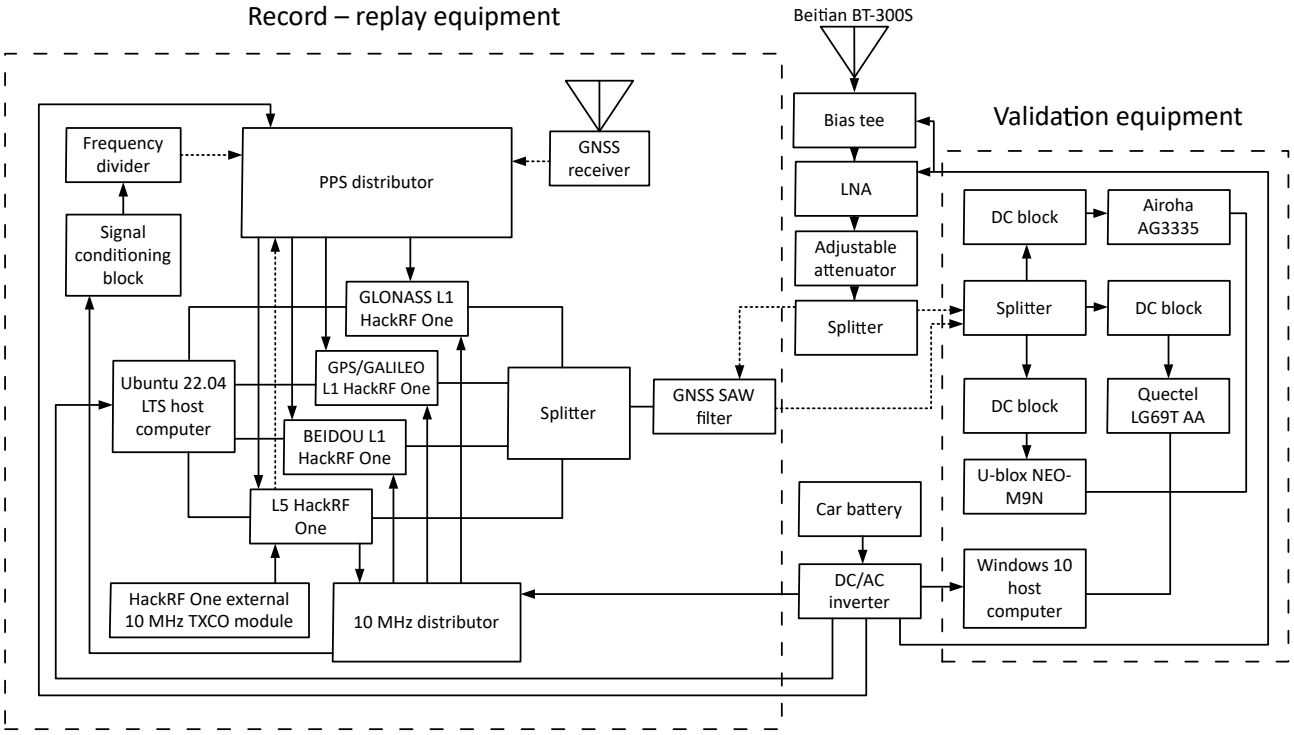
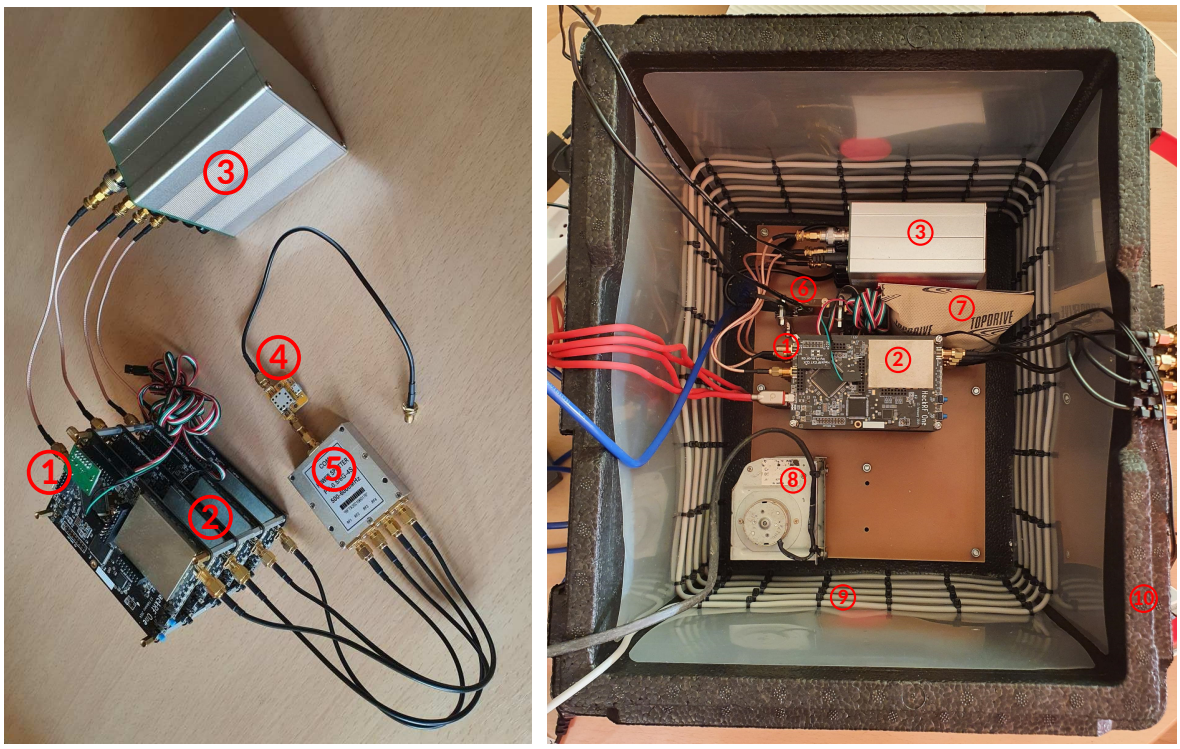


Figure 6: Block diagram of GNSS record–replay simulator and validation equipment.

Validation of GNSS records is performed using Airoha AG3335, Quectel LG69T AA and u-blox NEO-M9N GNSS receivers. Signal from antenna or from SDRs is delivered via three–way splitter. It is necessary to use DC blocks for all three receivers, because these receivers have DC voltage at the inputs for active antennas. All GNSS receivers are controlled via Microsoft Windows host computer, since receiver software only works on this operating system.

GNSS signals are collected using Beitian BT-300S active antenna, which is powered via a bias tee. To adjust signal level, after the bias tee, 40 dB LNA and 90 dB adjustable attenuator are used.

All equipment is powered by a separate car battery (not the one needed for the car to run) and DC/AC inverter. This is necessary to prolong operational time beyond what laptop batteries can provide, to power PPS and 10 MHz distributors, main antenna and, most importantly, for the SDR host computer to work without power limitations in performance mode.



(a) (b)

Figure 7: Components of GNSS record–replay simulator outside (a) and inside (b) of thermal insulated box: 1 – external 10 MHz oscillator module, 2 – rack of HackRF One SDRs, 3 – 10 MHz distributor, 4 – Surface Acoustic Wave (SAW) GNSS band filter, 5 – four–way splitter, 6 – 1 PPS distributor, 7 – dehumidifier, 8 – air circulation fan, 9 – heating wire and 10 – thermal insulated box.

3.1 Software defined radios

At the very center of developed GNSS L1+L5 Record–Replay simulator are software-defined radios. To record all GNSS L1 and L5 signals 4 HackRF One SDRs were used. HackRF One was chosen, because it can work in a frequency range from 1 MHz to 6 GHz, has a bandwidth of 20 MHz, can transmit and receive signals, has many gain control options, has hardware support for 10 MHz reference signal synchronization, has hardware support for 1PPS time synchronization, has 8 bit resolution and, last but not least, is relatively cheep [38]. It is important to note that 8 bit resolution is an advantage, because larger sample resolution can cause instability issues and record files would be very large, which require more storage. Actually in commercial applications like LabSat 3 Wideband, no more than 3 bit resolution is used [6], while many standard receivers use only 2 bits, and if needed only 1 bit is enough [55].

An alternative SDR could be USRP B210, which has all of HackRF One feature, but is expensive, and has 12 bit resolution. An attempt was made to record and replay GNSS signals using USRP B210, but in the beginning many instability issues were encountered, which there mainly caused by

software. When most of the instability issues were solved, it was discovered, that when recording two channels simultaneously at different center frequencies, USRP causes linearly time dependent phase difference between basebands. An attempt was made to eliminate this phase difference, but in the end it was not possible to remove this phase artifact, which destroys synchronization between L1 and L5.

3.2 SDR host computer

SDR host computer is no less important than SDRs themselves, because it is responsible for discontinuities in GNSS records, which effectively are momentary overflows or, in other words, data losses. To solve them, hardware path between SDRs and SSD must be shortened and computer must be appropriately configured in software. The issue is that for GNSS simulator to be portable, host computer is usually a lap–top. Installing a 3rd party USB PCIE card with multiple USB controllers is possible even if manufacturer of the lap–top never intended this type of use, but will be inconvenient to use in real life setting. Also, not any laptop can be used, because every SDR must be connected directly to host computer, without any hubs in order to not limit the bandwidth.

Software wise, Ubuntu operating system is highly recommended, because it does not load computer significantly and supports all the necessary software. But it should be installed in custom way. First minimal version of Ubuntu server operating system [56] is installed. Then Slim desktop manager [57] with Xfce desktop environment [58] is installed. After desktop environment, real–time Xanmod kernel [47] is installed. Then, SDR software [38] is installed and ramdisk using RapidDisk [44] software is configured.

Next BIOS is configured for real time operations. Frequency boost is disabled, hyper-threading is disabled, any kind of performance management is disabled and all kinds of power management is disabled. Additional settings and ways to change them varies from system to system, therefore, each system should be researched or tested via trail and error.

Finally, GNU Radio Volk profile [59] is generated for vector algebra on the new CPU configuration due to BIOS changes. Also, custom scripts and software for phase alignment, initialization and record–replay as real–time processes with the highest priority are installed.

Two systems with above–mentioned configuration were tested:

- HP EliteDesk 800 G5 mini with Intel Core i9-9900K processor, 2 TB Kingston KC3000 TLC NVME SSD disk, 64 GB DDR4 2666 CL19 RAM. 32 GB of RAM was used as cache. Only one NVME SSD was installed into this machine, therefore, cache was used for a partition of the SSD.

- HP EliteBook 850 G8 with Intel Core i7-1165G7 processor, 1 TB Samsung 970 EVO PLUS

TLC NVME SSD disk, 32 GB DDR4 3200 CL22 RAM. 28 GB of RAM was used as cache. Only one NVME SSD was installed into this machine, therefore, cache was used for a partition of the SSD.

Both systems performed without any issues during static and dynamic testing, when test duration varied from 45 minutes to 75 minutes. This means that no discontinuities occur while recording and the only factor limiting record length is the capacity of the SSD. However, it is important to mention, that in headless operation via VNC HP mini desktop PC still experienced discontinuities. The cause of this is LAN controller, which is integrated together with USB controller into PCH. Therefore, USB controller should always be prioritized and all other built in features, which may impact it should not be used or should be disabled.

3.3 PPS distributor for time synchronization

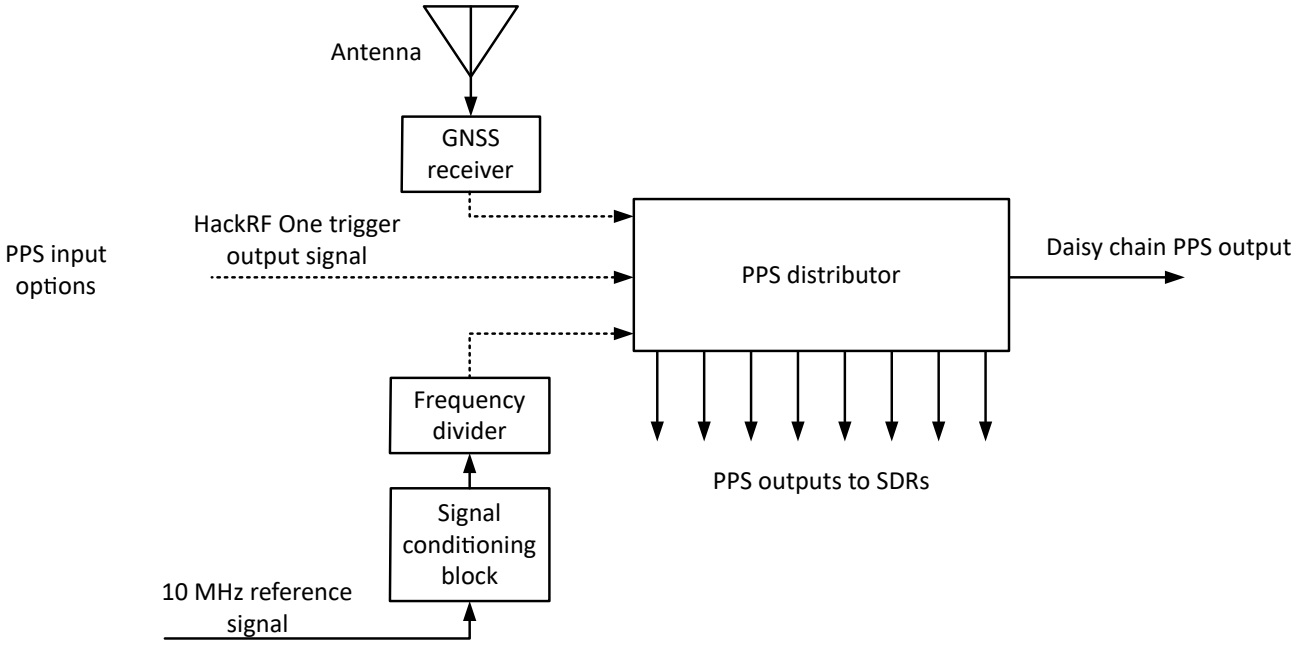


Figure 8: PPS Distributor block diagram.

To record and replay GNSS signals SDRs must be synchronized in time. To achieve such synchronization between HackRF Ones, one SDR's trigger output must be connected to the other SDRs' trigger inputs. According to this [60], achieved synchronization error is less than 1 sampling period. Also, HackRF Ones can be synchronized by using external 1PPS signal, which should be connected to trigger input of each SDR.

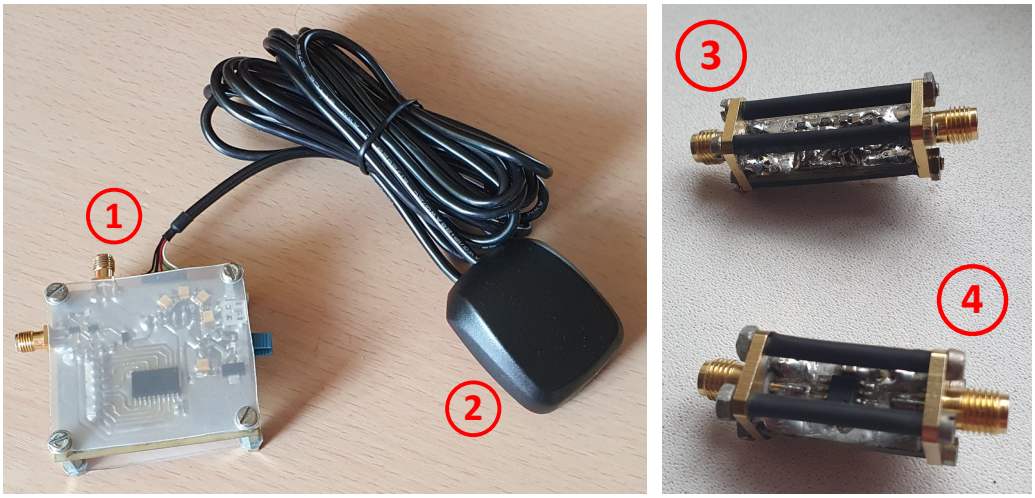
In practice, it was observed that using one of SDRs trigger output as a source is not reliable and using external 1PPS signal, produced by standard GNSS receiver, is better. However, to use an

external signal safely and reliably PSS signal distributor is necessary. Using [61] as an inspiration, PPS distributor was designed and build for GNSS L1+L5 Record-Replay simulator.

Block diagram of developed distributor is shown in figure 8. How this device looks in real life is shown in Fig. 9a. This device works by using GNSS receiver or any other external 1 PPS source, then this signal is buffered in order to not overload the source. Buffered signal is then distributed to 3 other buffers: one for daisy chain output, one for LED indication and one for 8 parallel outputs to SDRs. This PPS distributor supports both TTL and CMOS voltage levels (depending on jumper configuration) and can be powered via micro USB port.

During testing indoors, it was observed that using GNSS receiver as a source is often very inconvenient. Therefore, Using [62] as a reference, 10 MHz signal frequency divider was build. This module shown in Fig. 9b (marked 4). It is based on PIC12F675, 10 MHz reference signal is used as micro–controllers clock. One of the outputs is toggled every 5 million clock cycles. Additionally, it is configured such that output is high only for 10 ms, so that visual indication would be possible.

However, clock signal amplitude must be similar to supply voltage, therefore based on this [63], signal conditioner was designed and build. The final module can be seen in figure 9b (marked 3). This module can take any waveform and convert it to a square-wave of the same frequency and amplitude equal to the supply voltage. This module is built around Texas Instruments TLV3501 inverting hysteresis comparator with positive feedback.



(a) (b)

Figure 9: PPS distributor and GNSS receiver for time synchronization (a), signal conditioner and frequency divider (b): 1 – PPS distributor, 2 – GNSS receiver, 3 – Signal conditioner, 4 – Frequency divider.

By powering signal conditioner and frequency divider from PPS distributor same TTL or

CMOS voltage level can be used in the whole timing circuitry. Using 10 MHz signal for time synchronization completely eliminates dependency on external time synchronization and allows simulator to be used in anechoic chamber for transmission over air testing.

Lastly, note, that hardware time synchronization feature is only available in HackRF One's firmware release 2021.03.1 or newer. However, 2022.09.1 firmware release is advised, because SDR and software utilities work in a more stable and optimized way. Also, newest 2023.01.1 firmware should be avoided unless newest R9 revision of HackRF One is used, because currently hardware time synchronization does not work if 2023.01.1 firmware is used on older revision SDRs.

3.4 Common reference signal distribution scheme

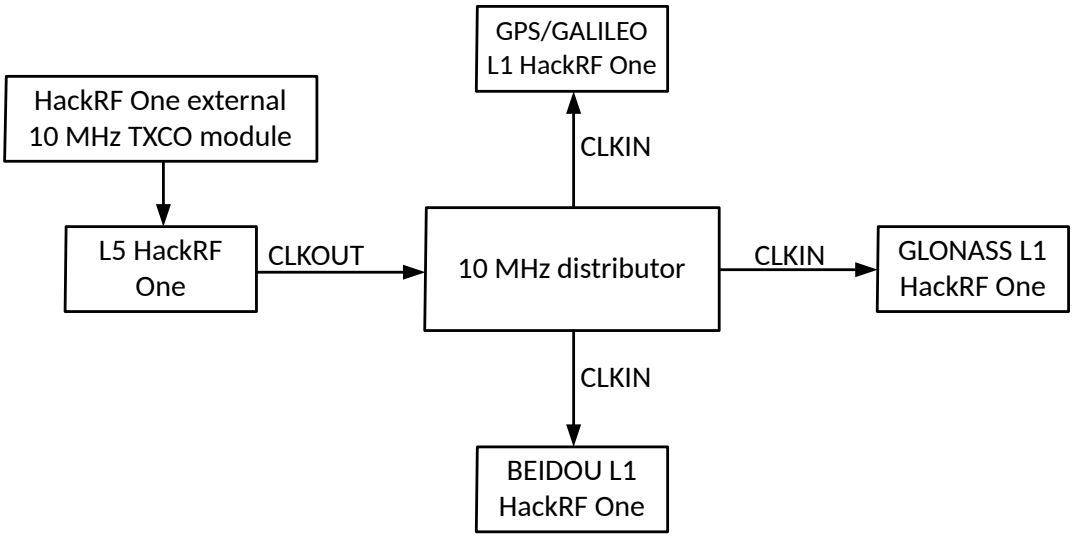


Figure 10: Common reference signal distribution block diagram.

To achieve frequency synchronization, which in turn will also have an effect on phase synchronization, star topology was used as shown in figure 10. First, an external 10 MHz oscillator module was used to improve frequency stability of HackRF One which records and replays L5 signals. This can be seen in Fig. 7 (marked 1). Then L5 HackRF One's reference signal, generated by on board Si5351C clock generator, was exported via CLKOUT output to 10 MHz distributor box, which can be seen in figure 7 (marked 3) as well. From distributor box reference signal is delivered to three remaining SDRs.

Since 10 MHz distributor has a built-in OCXO, it was attempted to realize reference clock sharing topology, but achieved synchronization was worse and had a significant effect on GNSS L1 and L5 signals, because, when replaying the record, L5 signals would sometimes randomly disappear after a few minutes into playback. Such behavior was not observed topology described above.

3.5 Software phase correction for phase synchronization

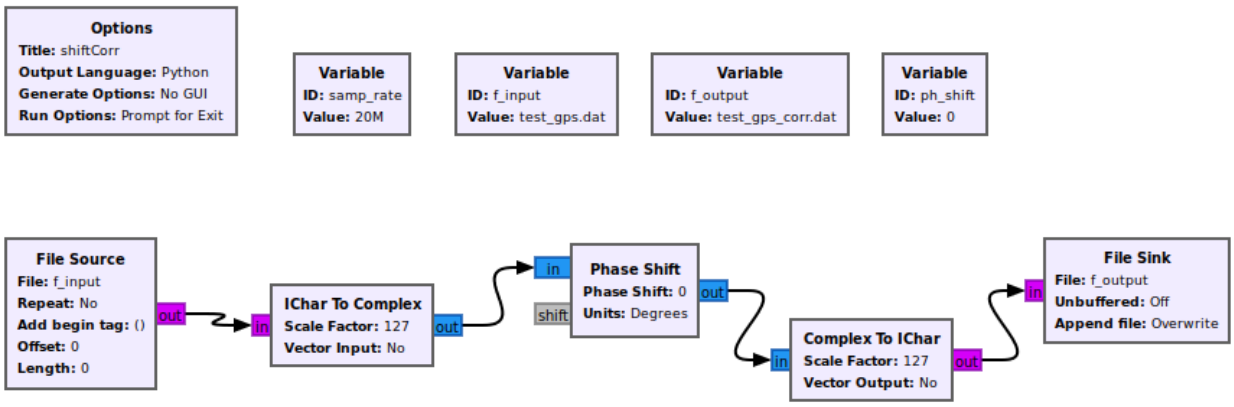


Figure 11: Applying Phase Correction Using GNURadio.

While using HackRF One, the only possible way to achieve phase synchronization is via software. This software was implemented in python and works by reading a part of GNSS records and calculating the necessary phase corrections to eliminate phase difference between signals. When reading signals, first 5 seconds are skipped, because HackRF PLL stabilizes only after 5 second from the time when SDR has started [64]. After skipping, only 30 seconds are read. This limitation is caused by random access memory and, as a result, minimum necessary amount of RAM for this software to work is 32 GB.

After records are loaded as complex samples into memory, necessary phase corrections are calculated in two ways: by calculating phase difference using 1 second worth of samples for each second in 30 seconds and averaging 30 calculations, and by calculating phase difference using all samples in 30 seconds. The final results are compared by the user and should not be different by more than 0,01 degrees. All of this is necessary to confirm that recorded signals fit the initial assumptions.

There are two initial assumptions. First assumption, recorded signals are comparable to noise generator output, because GNSS signals are below noise floor and no other signals suppose to be in L1 and L5 bands. This assumption is necessary, because a noise source is needed for phase synchronization [35], and if assumption is not violated, there is no need for special phase synchronization procedure like in KerberosSDR [65]. Second assumption, phase correction is a single complex number that can be applied to the whole record. If this assumption is invalid, final results produced by two different ways of phase correction calculations will differ by a lot more than 0,01 degrees. One of possible ways how these assumptions can be violated is by choosing very large values for gain parameters of HackRF One. The correct gain values should be determined by trial and error.

Furthermore, in written software GNSS L5 recording is used as reference and phase correction are calculated in such way, that phase difference between GNSS L5 record and any other record would be 0 degrees.

Since GNSS records can be very long, calculated phase corrections are applied using GNU Radio software as shown in figure 11. GNU Radio generates the constructed flow diagram as a Python script, which can be used inside other Python scripts [59]. By doing this, main script calculates corrections and then starts GNU Radio script for each record to apply calculated corrections. Note, that for scripts to work, GNU radio 3.9 or newer is needed.

After phase corrections are calculated and applied, user will find new, generated GNSS record files, which are phase wise aligned. Obviously, since new set of records is generated, computer must have sufficient storage.

3.6 Synchronization and sampling rate

GNSS L1 and L5 signals are transmitted on different center frequencies and use different modulations. Different modulation mean that signals' bandwidths are also different. Therefore, in theory, different GNSS signals could be recorded using different sampling rates. This would be very beneficial to the whole system, because multiple SDRs could be connected to the same USB port via hub, less storage would be used by records and overall load on the host computer would be reduced.

However, despite extensive efforts, L1 and L5 signals were not successfully recorded and replayed in a multi-sampling-rate system. None of L1+L5 capable receivers, used in experiments, were able to detect both L1 and L5 signals. But this does not mean that no signals can be recorded using different sampling rates for different center frequencies. If the goal is to record only L1 signals, there are no issues in using different sampling rates. All receivers were able to detect L1 signals transmitted on all center frequencies. Problems would begin only when an attempt to record multiple bands was made.

The most likely explanation for such behavior is that when recording GNSS signals using different sampling rates synchronization between signals is destroyed. This is not an issue for L1, because receivers do not use acquisition results from GPS L1 signals to acquire, for example, GALILEO L1 signals. No synchronization between GPS and GALILEO could even be expected, because these signals are transmitted from different satellites.

Unfortunately, situation for L1 and L5 bands is different, because these signals are transmitted by the same satellites, therefore, these signals are expected to be synchronized and receivers do use acquisition results from L1 signals to acquire L5 signals [32].

The fact that GNSS receivers heavily rely on L1 for L5 is clearly seen in subsequent results, which are discussed later. Also, it is worth noting, that commercially available solutions do not use different sampling rates for multiple bands as well. For example, LabSat 3 Wideband, can record 3 different bands, but all of them are recorded using the same bandwidth and, therefore, sampling rate [6]. However, this creates situations, where more bandwidth is recorded than needed and system is stressed more than necessary. Well, in case of LabSat 3 Wideband, this problem is solved by reducing resolution from 3 bits to 1 bit.

Overall, it can be said, that to record multiple GNSS signals in the same band, different sampling rates for different center frequencies can be used, because the loss of synchronization is not important. However, to record multiple GNSS bands, the same sampling rate must be used for all center frequencies, because synchronization is imperative.

Since HackRF One SDR does not have a possibility to adjust resolution, the only possible optimization for the system is to record GNSS signals with reduced bandwidth.

3.7 Thermal stabilization

It was observed that temperature of GNSS simulator must be stable and constant during record and playback procedures. Otherwise, it will be impossible to replay records even after a few hours due to ambient temperature fluctuations. To solve this issue, all internal components are placed into thermal insulated box with heating wire mounted on the walls as can be seen in Fig. 7b. There is also an air circulation fan (marked 8 in Fig. 7b), which forces air to move diagonally towards SDR rack in closed box.

To maintain temperature in this setup, firstly temperature inside the box is raised above ambient temperature to the point, at which there is a natural tendency for temperature to drop if heating wire is turned off and SDRs do not transmit or receive signals. At this temperature, when inside the box heats up due to tasks being performed, all the heat is dissipated via holes in the cover of the box. If more heat is dissipated than SDRs can provide and temperature drops below preset operating temperature, heating wire turns on to provide additional heat. In order to achieve temperature stabilization the whole system must be in such a state that heat added to the system by SDRs and other equipment is equal to the heat lost by the system.

The rate of heat loss can be controlled by adjusting previously mentioned holes in the cover of the box. Heating wire and chamber volume should be chosen accordingly as well, since they determine how long initial heating takes. Also, heat added by the heating wire, at the stable temperature state, should be comparable to heat added by the SDRs. Additionally, it was observed that HackRF

One SDRs produce more heat while transmitting compared to when they are receiving signals.

Thermal stabilization has an observable impact on dependencies over time of carrier to noise ratio and number of visible satellites, which are shown in Fig. 12. These dependencies their measured during dynamic testing, while driving the same trajectory with temperature stabilization and without it. As can be seen in Fig. 12a, there are clear differences related to signal origin: in case of direct satellite reception, C/N_0 dependencies of L1 and L5 signals are similar and close to each other, but in case of SDR–based reception, C/N_0 dependencies of L1 and L5 signals are far apart and the difference between them is increasing with time without thermal stabilization.

Differences can be observed in the number of visible satellites dependencies as well (Fig. 12b), SDR–based dependencies reporting smaller overall number of satellites compared to direct satellite reception. Also, SDR–based L1 curve follows direct satellite L1 curve closely, while SDR–based L5 curve and direct satellite L5 curve show unrelated behavior.

However, despite these differences, when temperature stabilization is present, there is a stable difference of carrier to noise ratio and number of visible satellites between L1 and L5 signals replayed by SDRs as shown in Figs. 12c and 12d. When no thermal stabilization is applied, absolute values of these differences increase throughout the whole test. Difference fluctuation around a constant value matches what is observed in direct satellite reception as well. Therefore, not only thermal stabilization solved simulators playback dependency on temperature fluctuations, it also improves the overall performance of the simulator, because replayed L1 and L5 signals no longer display independent behavior of their parameters.

Furthermore, fluctuation around a constant difference value is the expected behavior, because if GNSS L1+L5 receiver detects L1 signal from L1 and L5 capable satellite, almost immediately L5 signal is detected as well. This means that total input into the difference of visible number of satellites between L1 and L5 is 0. Difference can only be comprised of L1 only satellites and can have a significant change only in the beginning of receivers operation, while not all visible satellites are detected. After all visible satellites are detected it should be constant with fluctuations caused by multipath and atmospheric fluctuations.

When signal originates from discrete SDRs without temperature stabilization, stable difference is impossible, because SDRs replay noise–like signals, which are influenced by the thermal noise of the SDR itself. Thermal noise depends on temperature directly according to (2). Temperature fluctuation of individual devices are independent, therefore, different GNSS signals will be affected differently over time and differences between GNSS signals parameters observed by GNSS receiver will not be constant.

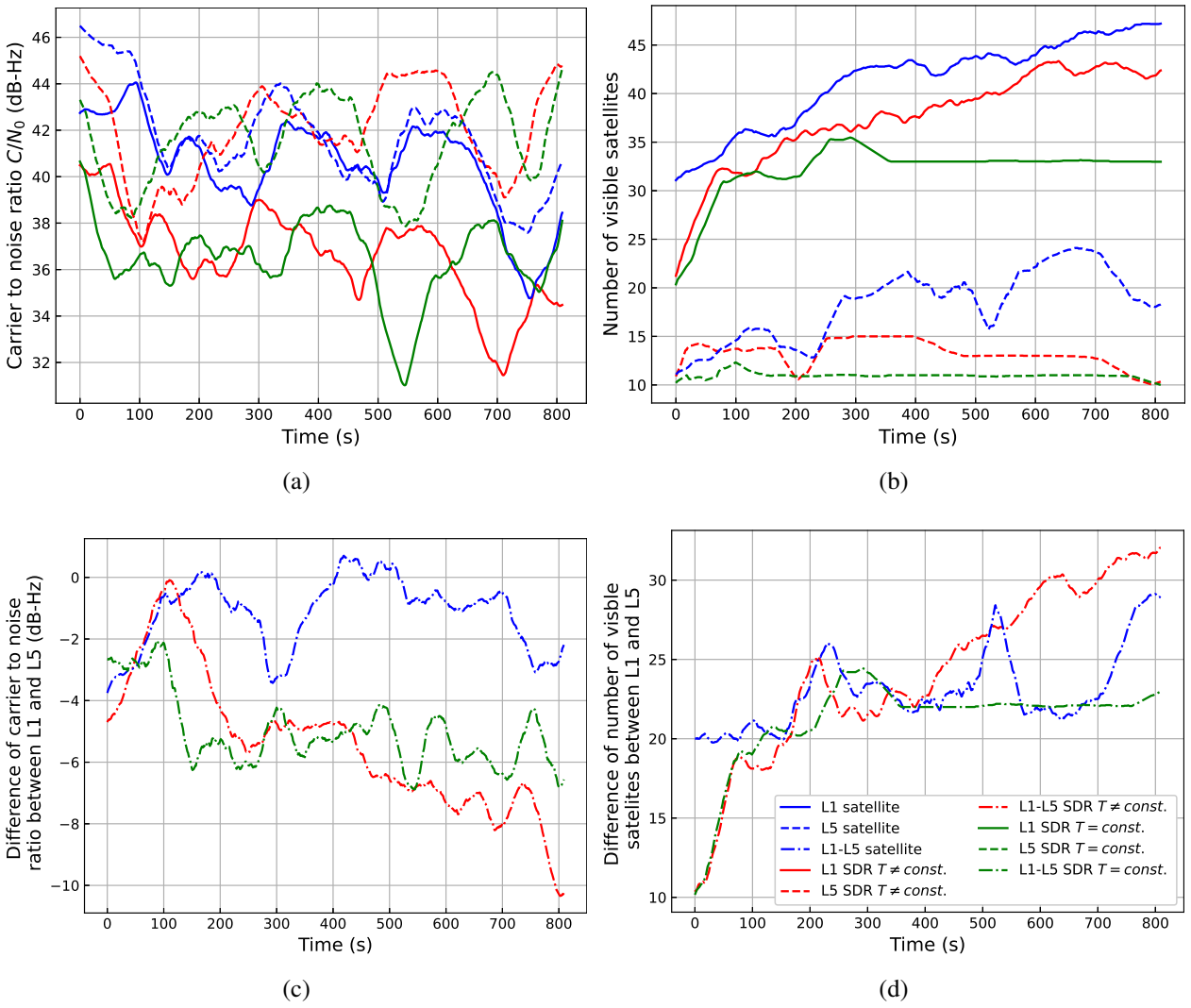


Figure 12: Dependency over time of carrier to noise ratio (a), the number of visible satellites (b), difference of carrier to noise ratio between L1 and L5 (c) and difference of number of visible satellites between L1 and L5 (d) for L1, L5 signals received directly from satellites and replayed by SDRs with temperature stabilization and without temperature stabilization as shown in the common legend. Data logged using Airoha AG3335 GNSS receiver.

3.8 Initialization

For successful operation of GNSS simulator, it was observed that initialization procedure is necessary. Initialization is done by performing 4 steps:

1. Connect a 50 Ω terminator at the input of the simulator;
2. Record for 3 minutes;
3. Perform phase alignment;
4. Fully replay aligned recording.

After initialization is done, 50 Ω terminator is disconnected and, after connecting GNSS antenna with LNA and attenuator, GNSS signals can be recorded. However, during playback, after SDRs were fully turned off and turned on again, initialization may need to be performed once or twice in order to replay both L1 and L5 signals. If appropriate number of initializations is not performed before replay, after full power cycle, only L1 signals will be visible by GNSS receivers.

This behavior is related to the fact that GNSS receiver manufactures utilize L1 acquisition results for L5 acquisition and to how time synchronization circuitry is implemented. It was observed that after switching from external synchronization source to 10 MHz internal time synchronization source, records, which required one initialization, would require no initialization and record, which required two initializations, would require only one initialization. Note, synchronization sources' wave-forms were different, but PPS distributor mitigates those differences. It was not investigated further, because after the switch more than 200 records there made and all of them required one initialization. Previously, recorded signals, which needed one or two initializations using external source were investigated further using statistical and machine learning approach.

4 Results and Discussion

4.1 Statistics of GNSS records

For GNSS records statistical analysis more than 250 12 MHz bandwidth record sets were collected, while simulator was stationary. Each set contains 4 GNSS records: one L5 and three L1 for each center frequencies. For further discussion, analysis results of one particular set are used unless otherwise is stated. This set illustrates observations and patterns discovered during analysis of the whole data set.

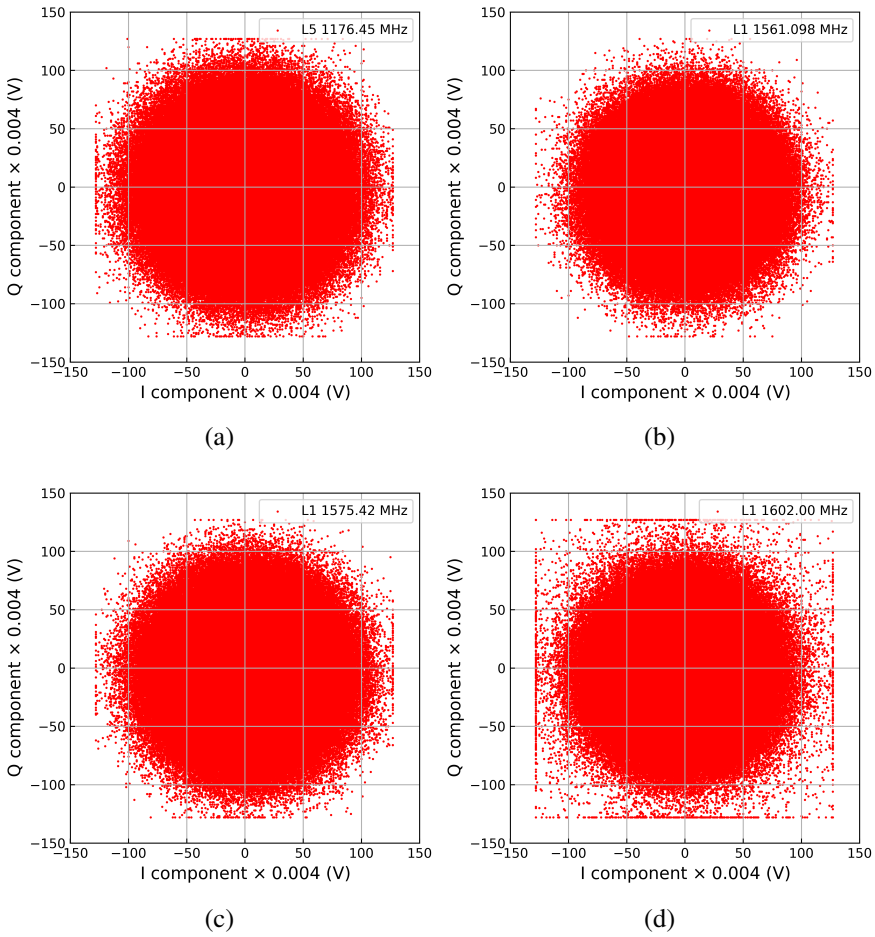


Figure 13: GNSS signals' samples represented in IQ plane: L5 1176.45 MHz (a), L1 1561.098 MHz (b), L1 1575.42 MHz (c), L1 1602.00 MHz (d).

Analysis starts from investigating primary IQ data, which is visualized in Fig. 13. This data represents first 100 ms of the record. From figure, it is clear that all record signals were recorded at the similar signal level and all signal experience similar amount of amplitude clipping. Such conditions can only be achieved via trail and error by configuring gain settings of each SDR individually, because discrete SDRs, although the same, behave differently at different frequencies.

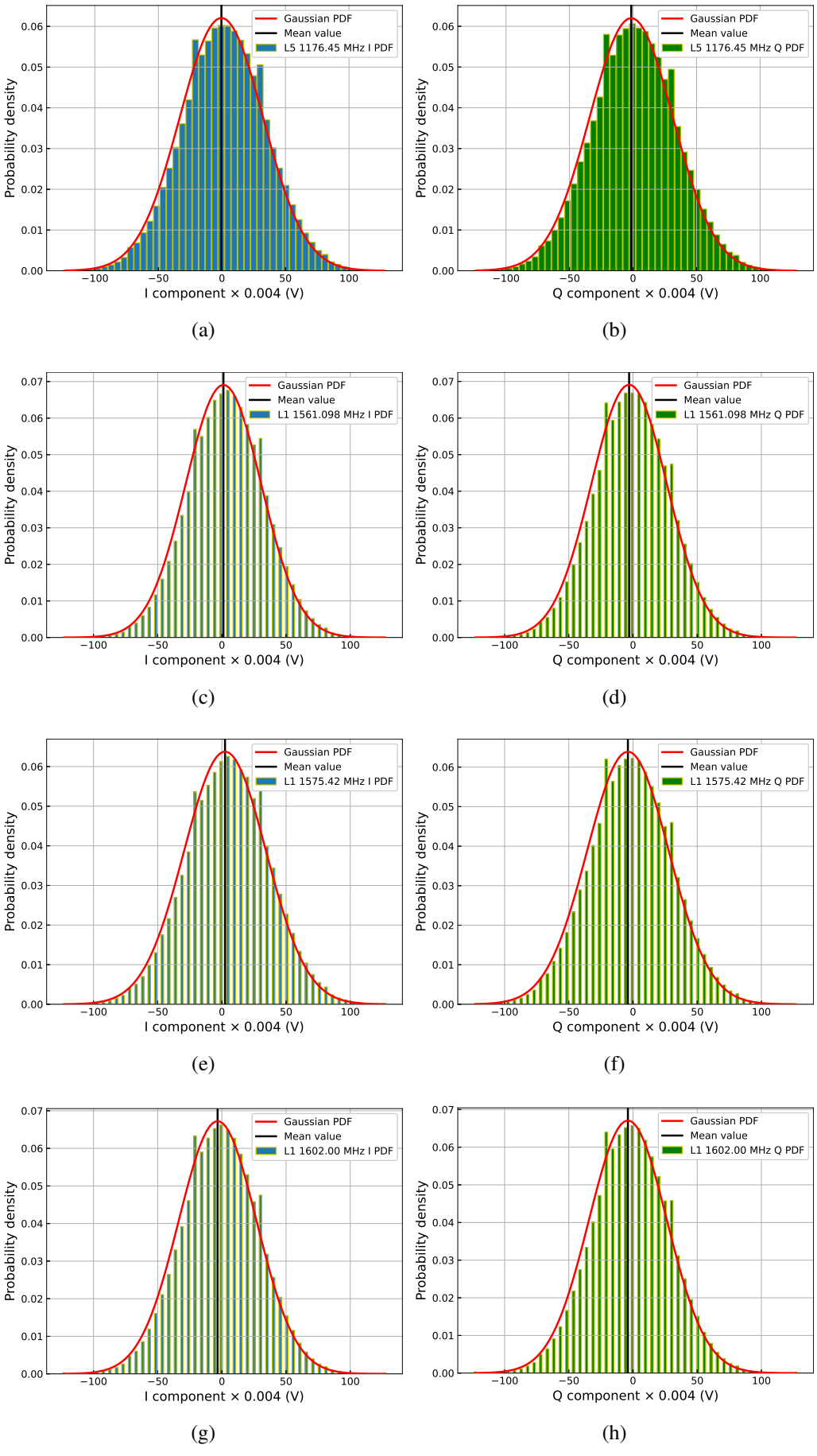


Figure 14: Distributions of I and Q components of GNSS records: L5 1176.45 MHz I (a) and Q (b), L1 1561.098 MHz I (c) and Q (d), L1 1575.42 MHz I (e) and Q (f), L1 1602.00 MHz I (g) and Q (h).

Further insight into IQ data can be obtained by analyzing I and Q component separately. Results are shown in Figure 14. As expected of the noise–like signal, I and Q components have distributions that can be modeled using Gaussian distribution. Root mean squared error (RMSE), when comparing measured PDF to Gaussian PDF, is from 0.002 to 0.003. Also, both I and Q components have similar standard deviations. This further shows that GNSS records can be modeled using complex white noise model. However, mean values (vertical black lines) are non-zero and different for I and Q components. This means amplitude distributions should be modeled using Rician distribution. But situation is different as can be seen in Figure 15.

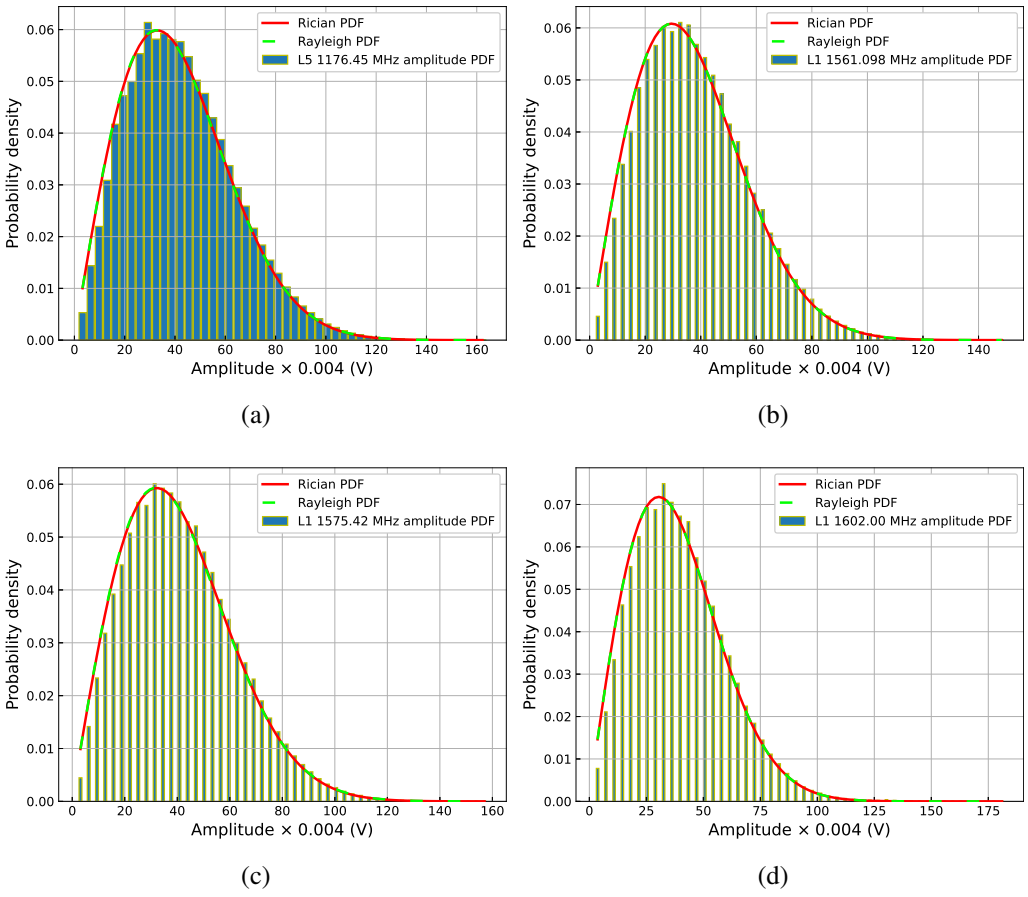
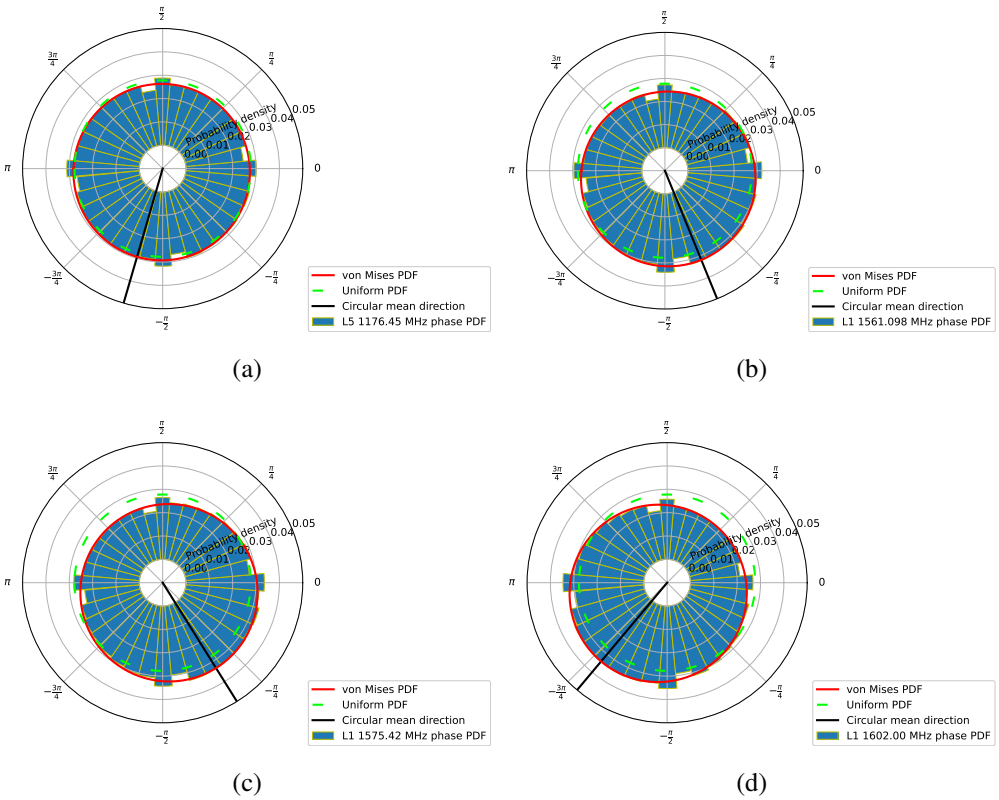


Figure 15: Distributions of amplitudes of GNSS records: L5 1176.45 MHz (a), L1 1561.098 MHz (b), L1 1575.42 MHz (c), L1 1602.00 MHz (d).

Both Rician and Rayleigh distributions can be used to model GNSS signal amplitude distribution. Main differentiating factor between these distributions is the existence of line–of–sight (LOS) component in the data. LOS component exists in GNSS record data, but, since GNSS signals are below noise floor, LOS component is very weak. This allows to use both possible distributions for amplitude distribution modeling. Wellness of fit is confirmed by RMSE when comparing measured PDFs to Rician or Rayleigh PDFs and is equal to 0.002 for both distributions.

However, existence of non-zero IQ means have significant effect on phase distributions of GNSS records, which are shown in Figure 16. Here all GNSS records have mean circular directions around which all other values cluster. From the figures it is obvious that von Mises distribution fits better compared to uniform distribution.



(a) (b) (c) (d)

Figure 16: Phase distributions of GNSS signals: L5 1176.45 MHz signal phase (a), L1 1561.098 MHz signal phase (b), L1 1575.42 MHz signal phase (c), L1 1602.00 MHz signal phase (d).

While all GNSS signals have a circular mean direction in some signals values cluster around this direction more than in others. To evaluate this behavior von Mises distributions κ parameter can be used. For L5 1176.45 MHz (Fig. 16a) this parameter is 0.057, for L1 1651.098 MHz (Fig. 16b) κ is 0.14, for L1 1575.42 MHz (Fig. 16c) κ is 0.18 and for L1 1602.00 MHz (Fig. 16c) κ is 0.22.

The reason why L5 1176.45 MHz (Fig. 16a) has such a low κ values is likely, because this signal was recorded using reduced bandwidth, which will be discussed later. Also, It was observed that GNSS records, which have amplitude clipping present, have less pronounce circular mean direction and lower κ values. In this signals case, as was shown in Fig. 16a, small amount of clipping is present, since some points are at $\pm 127 \times 0.004$ V. This however, does not affect the performance of GNSS simulator and GNSS receivers can determine position from clipped record without any issues.

Mean direction itself is less important because it depends on the center frequency of the signals and can be easily adjusted by changing the length of the cable between antenna and GNSS simulator.

However, existence of this mean direction is unusual, because this is not phase difference and just the phase of the signal itself. In this case uniform distribution should be more likely.

Another thing to note, is that for comparing which distribution fits measured distribution better root mean squared error was used. Usually, to determine if distribution fits measured data, statistical test like Kolmogorov–Smirnov test is used [66]. However, it was observed that statistical test can only be used when number of data points is up to a 1000. If data set is larger, which is the case when working with GNSS records, statistical test will always report that distribution does not fit the data. This behavior does not mean that statistical test is wrong. Test is right, because with larger data set it can be evaluated with greater confidence if distribution model fits the data or not. However, from practical point of view this is incorrect, because statistical test considers practically insignificant difference as significant. This means that one cannot compare distribution models and their wellness of fit using statistical test. Since this problem was encountered in this work, root mean squared error was chosen as an alternative, because it can be used to compare linear and non-linear curves.

To determine if statistical parameters are stationary in time GNSS records were analyzed by dividing 5 minute record into 100 ms windows and calculating descriptive statistic as well as fitting uniform and von Mises distributions and calculating errors for each window. After plotting obtained results, no significant behavior was observed. Through the whole record, fluctuations were not higher than hundredths or thousandth parts of the constant values around which fluctuations occurred.

Therefore, it was possible only statistically to determine if significant changes were happening over the 5–minute interval. Since most of the parameters are derivative from primary IQ values, it would be sufficient to check only one parameter, which is directly derived from IQ. Such parameter is complex signal phase. Since phase can be modeled using von Mises distribution, which has circular mean and κ as parameters, it is important to determine if these parameters are statistically the same for the whole time interval. This can be done using Watson statistical test [28, 29].

To illustrate time dependencies a different record set was chosen than the one used for previously shown statistics, because it better shows possible behavior. For L1 and L5 signals, circular mean dependencies on time and Watson test p-value results are shown in Figures 17 and 18. Circular mean fluctuates around a constant value, however statistically it cannot be said that the same circular mean apply for the whole 5 minute interval.

In GNSS L5 1176.45 MHz case (Figs. 17a and 17b), same circular mean and κ values can be used for the whole interval, because p-value never drops below 0.05 (red horizontal line). The same situation can be seen in GNSS L1 1561.098 MHz case (Figs. 17c and 17d). However, this time p-value approached 0.05 closely around 50 seconds mark, which corresponds to global minimum in circular mean moving average curve. In GNSS L1 1575.42 MHz case (Figs. 18a and 18b), 5 minute interval

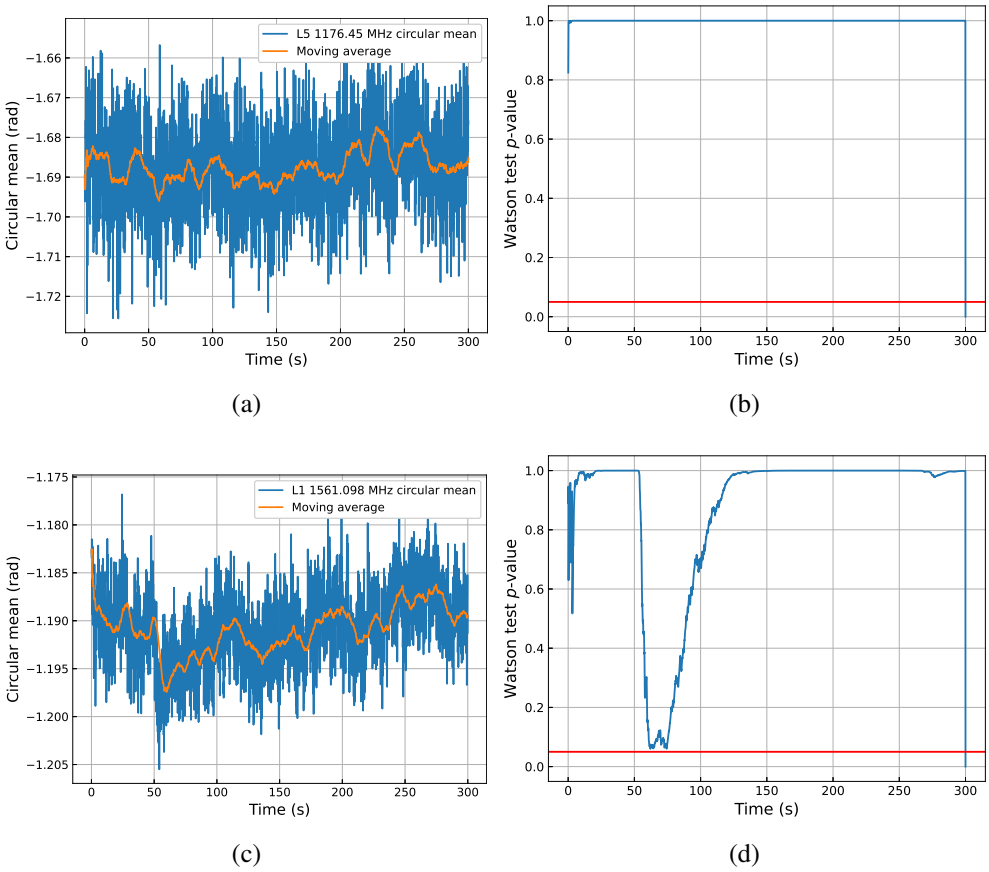


Figure 17: Circular mean dependency on time (a, c) and time intervals of stationary statistics (b, d) of GNSS L5 1176.45 MHz (a, b), L1 1561.098 MHz (c, d) signals.

splits into 5 intervals. In this case, splitting can also be related to behavior of circular mean moving average curve. The most interesting situation is in GNSS L1 1602.00 MHz case (Figs. 18c and 18d). This time 5 minutes interval is split into 108 intervals.

The observed behavior can be explained by several factors. Firstly, existence of splitting depends on how much amplitude clipping was present while recording the particular signal. Out of recorded signals, L5 experienced the most clipping, therefore the same statistics apply in the whole interval. Secondly, the amount of splitting can be explained by von Mises κ parameter. For L5 1176.45 this parameter is 0.062, for L1 1651.098 MHz κ is 0.30, for L1 1575.42 MHz κ is 0.44 and for L1 1602.00 MHz (Fig. 16c) κ is 0.55. L1 1602.00 MHz had the largest κ value out of all signals shown, therefore, the most splitting occurred.

In general, if data set is taken or recorded at random, the most likely situation will be the one shown figure 17. This also applies even if record is recorded while GNSS simulator was moving in the car. However, sometimes without changing gain settings, GNSS record set with high values of κ will be recorded. Those data sets manifest the situation illustrated in figure 18.

To gain further insight into statistical phase characteristics of GNSS signals, GNSS signals

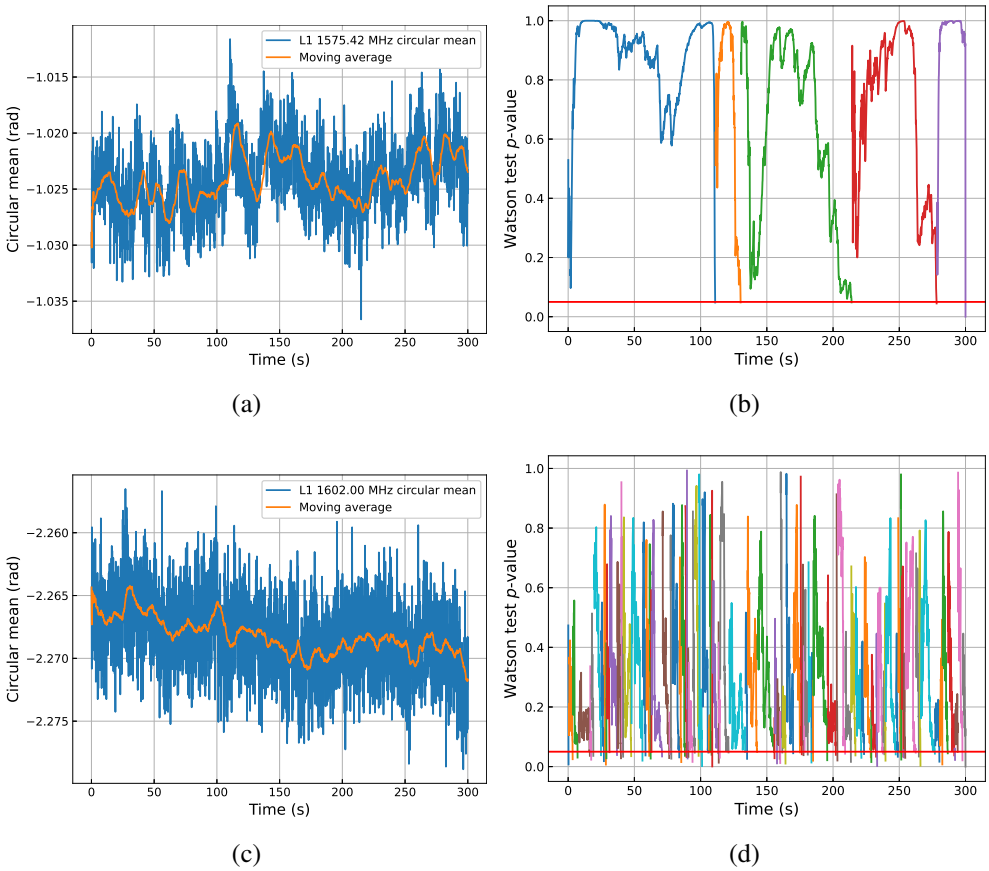


Figure 18: Circular mean dependency on time (a, c) and time intervals of stationary statistics (b, d) of GNSS L1 1575.42 MHz (a, b) and L1 1602.00 MHz (b, d) signals.

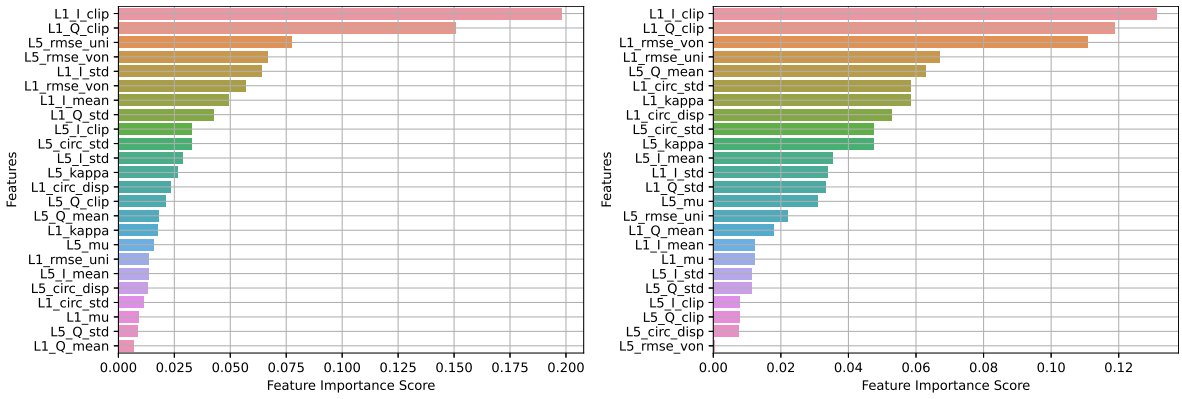
from a specific satellite should be analyzed. The GNSS signals used for above described statistic are the resultant signals of all GNSS signals transmitted at the specific center frequency. However, to obtain phase information of a single GNSS signal a GNSS receiver with raw data functionality is necessary. Unfortunately, during the time of this research such receiver was not available.

4.2 Classification of GNSS records using machine learning classifiers

As mention in initialization subchapter, GNSS records produced by proposed GNSS simulator can be classified into 2 types. Type one requires one initialization procedure, while type 2 requires that the same initialization would be performed twice. Without appropriate number of initializations only L1 signals are visible by receiver. However, to determine the type additional experimentation is needed. To avoid it two classifier models were built using machine learning random forest classifier (RFC) algorithm and support vector classifier (SVC) algorithm which are available in *scikit-learn* machine learning python library [67, 68].

For model building 56 GNSS records were used, because they were recorded under the same

conditions and clearly indicate the existence of two types. Out of these records 34 were type 1 and 22 were type 2. Only GNSS L1 1575.42 MHz and L5 1176.45 MHz signals were used because the majority of L5 signals are paired with L1 signals transmitted on 1575.42 MHz center frequency. To increase the number of items for model training and validation, each record was further divided into statistically different sections according to Watson test performed in time dependency analysis. Since time analysis was performed using 100 ms window, each section had multiple values of each feature, which statistically were identical, therefore an average was calculated for each feature in each section. Final number of items for model development was 68. The increase is not big due to reason discussed previously.



(a) (b)

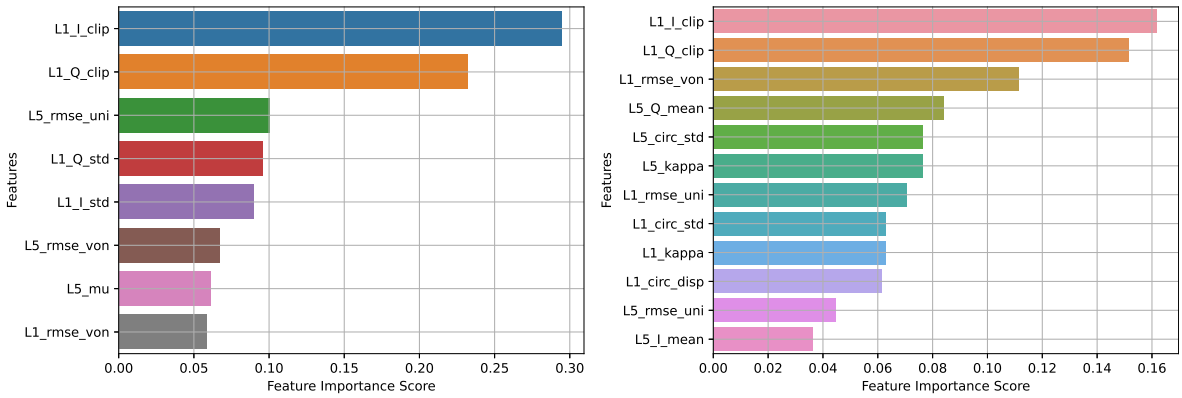
Figure 19: Feature importance scores when using all features: random forest classifier (a) and support vector classifier (b).

Each item had 24 features. For L5 signal: I component amplitude clipping (L5_I_clip), Q component amplitude clipping (L5_Q_clip), I component mean (L5_I_mean), I component standard deviation (L5_I_std), Q component mean (L5_Q_mean), Q component standard deviation (L5_Q_std), circular dispersion (L5_circ_disp), circular standard deviation (L5_circ_std), circular mean (L5_mu), κ (L5_kappa), root mean squared error, when comparing L5 PDF to von Mises PDF (L5_rmse_von), root mean squared error, when comparing L5 PDF to uniform PDF (L5_rmse_uni). For L1 signal: I component amplitude clipping (L1_I_clip), Q component amplitude clipping (L1_Q_clip), I component mean (L1_I_mean), I component standard deviation (L1_I_std), Q component mean (L1_Q_mean), Q component standard deviation (L1_Q_std), circular dispersion (L1_circ_disp), circular standard deviation (L1_circ_std), circular mean (L1_mu), κ (L1_kappa), root mean squared error, when comparing L1 PDF to von Mises PDF (L1_rmse_von), root mean squared error, when comparing L1 PDF to uniform PDF (L1_rmse_uni).

While building RFC and SVC models, feature importance scores can be obtained. They are

shown in figure 19. Both models have evaluated L1 I and Q components amplitude clipping as the most important features. However, different features were chosen for the third importance place. Random forest classifier has chosen L5 root mean squared error, when comparing L5 PDF to uniform PDF (Fig. 19a), while support vector classifier has chosen L1 root mean squared error, when comparing L1 PDF to von Mises PDF (Fig. 19b).

Models were evaluated using 10–fold cross–validation. Using all 24 features both RF and SVC classifier had the same, following performance scores: accuracy 97.14 %, precision 100.00 %, recall 95.00 % and F1 96.67 %. Total performance score was 97.20 %. To reduce over–fitting and generalize the models, feature set was reduced. First, feature extraction was attempted using PCA, but new independent variables did not show any significant clustering or other kind separation in the groups. Therefore, instead of feature extraction, feature selection was used.



(a) (b)

Figure 20: Feature importance scores after feature selection: random forest classifier (a) and support vector classifier (b).

Features selected for the final models and their importance scores are shown in Figure 20. We can see that the top 3 position did not change. Also, for random forest classifier performance score did not change as well, but now it only needs 8 features to make a prediction of record type. However, support vector classifier experience an improvement and now had all scores equal to 100 % and only needs 12 features to make a prediction.

Record type dependencies on the most important features are shown in figure 21. It is clear that all data point lay in the plane of L1 I and Q components amplitude clipping (Fig. 21b). The less clipping is present during recording, the more likely that type 1 record will be produced. Also, from RMSE and clipping dependencies (Fig. 21a and Fig. 21c), it can be said that at smaller clipping RMSE values are lower and type 1 is more likely, but obviously there are outliers.

Interesting situation is in figure 21d. This figure suggests that for type 1 record to be made

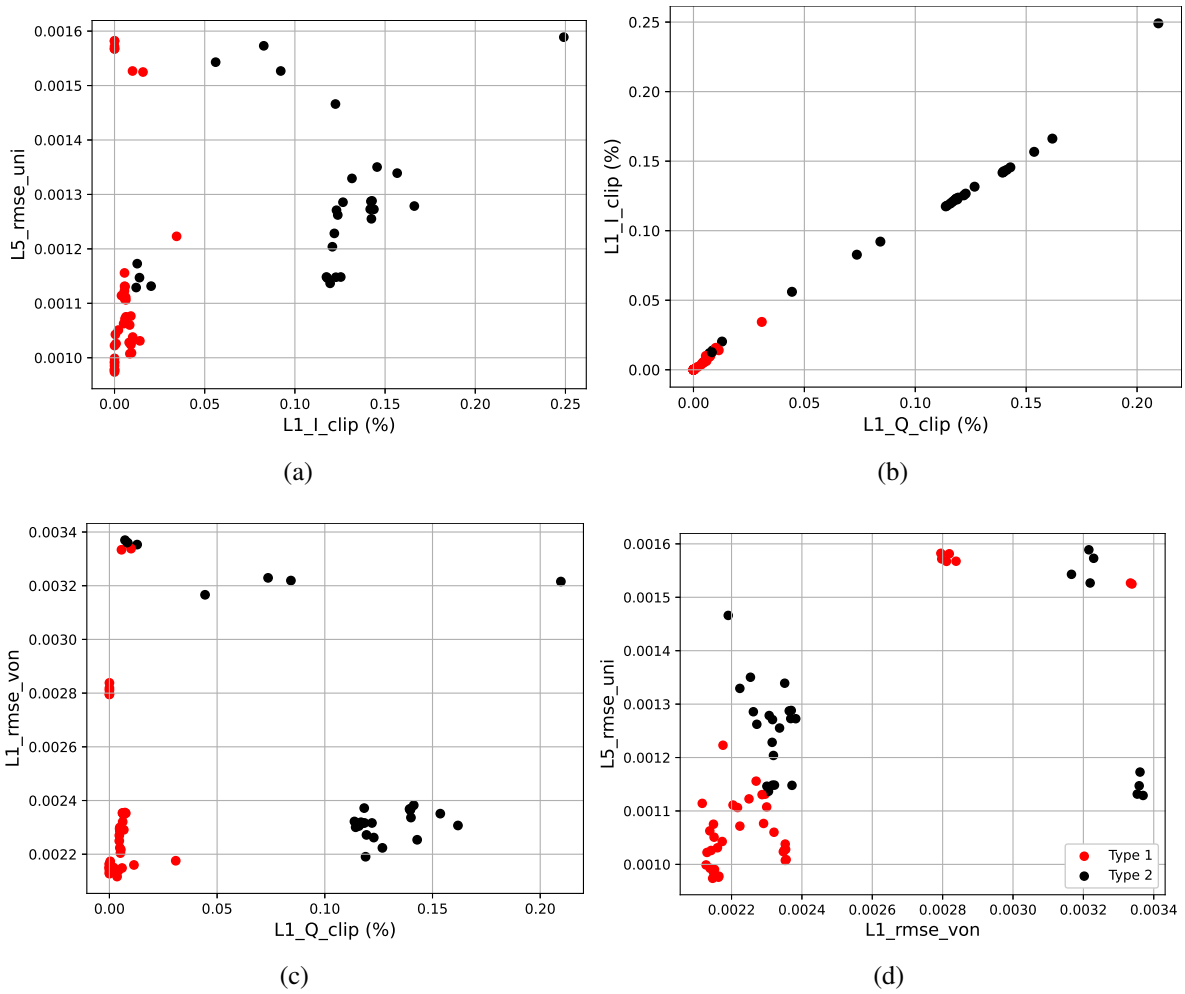


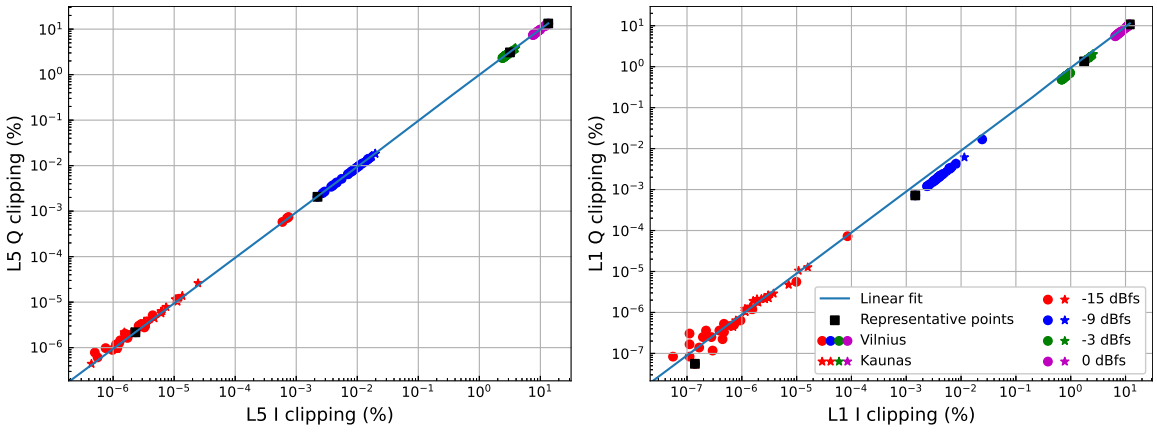
Figure 21: Dependency of record type on L5 root mean squared error, when comparing L5 phase PDF to uniform PDF and L1 I component amplitude clipping (a), dependency of record type on L1 I and Q component amplitude clipping (b), dependency of record type on L1 root mean squared error, when comparing L1 phase PDF to von Mises PDF and L1 Q component amplitude clipping (c) and dependency of record type on L1 root mean squared error, when comparing L1 phase PDF to von Mises PDF and L5 root mean squared error, when comparing L5 phase PDF to uniform PDF (d).

both L1 root mean squared error, when comparing L1 phase PDF to von Mises PDF and L5 root mean squared error, when comparing L5 phase PDF to uniform PDF must be low. Since it is known from result and discussion in the next subchapter that amplitude clipping affects distribution type and distribution parameters, this suggests that different amounts of clipping must be present for L1 and L5 signals while recording for type 1 record to be created. This is only possible due to mismatch in gain settings, difference of SDR revisions or frequency dependent radio channel behavior.

Overall, it must be said, that obtained results are expected, because everything is derived from IQ samples, therefore, features that directly represent IQ have high importance in machine learning models. The same can be said about RMSE values, because they, also, reflect effects of clipping.

4.3 Influence of IQ samples amplitude clipping on GNSS record statistics

IQ amplitude clipping has noticeable effect on GNSS simulator performance. Statistical analysis has indicated that clipping distorts distributions of GNSS records. The influence of such signal limiting have been investigated further at different signal levels. These levels are expressed in decibels relative to full scale (dBfs) as reported by HackRF One SDR with firmware release 2022.09.1 or newer.



(a) (b)

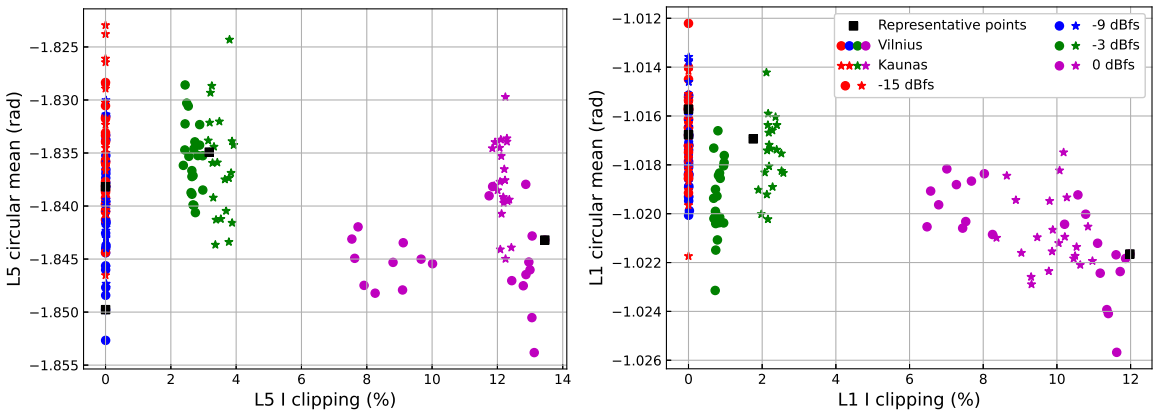
Figure 22: Symmetry of I and Q components amplitude clipping at different signal levels: L5 1176.45 MHz signal (a) and L1 1575.42 MHz signal (b).

For this analysis data was collected after switching to internal 10 MHz reference signal for time synchronization. Measurements were done in two different cities: Kaunas and Vilnius. A total of 22 GNSS L5 1176.45 MHz and L1 1575.42 MHz record pairs per city per signal level were collected. In total 176 record pairs are represented in figures 22 - 26.

First, symmetry of clipping was investigated. The results are shown in Fig. 22. Dependencies are represented in logarithmic scale for readability reason. Both L5 (Fig. 22a) and L1 (Fig. 22b) IQ clipping have linear dependencies on each other with tangent of the line of best fit equal to 0.99 for L5 and 0.95 for L1. This result indicates that clipping is symmetrical and for further analysis only I component amplitude clipping can be used as independent variable.

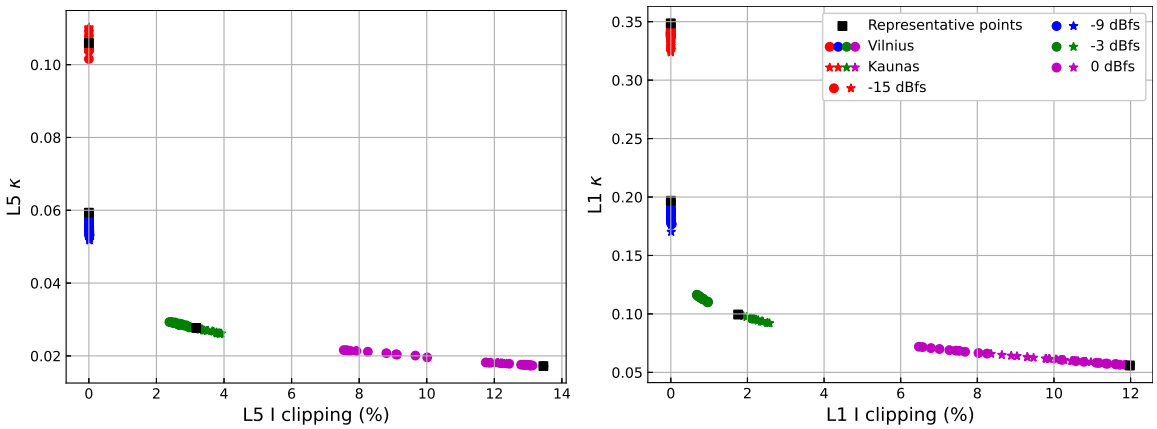
The effect on record phase distribution is investigated via von Mises distribution circular mean and κ parameters. In figure 23 it can be seen that for L5 (Fig. 23a) and L1 (Fig. 23a) circular mean does not depend on clipping. Values are spread 2 to 3 hundredths parts of the radian. This leads to believe that circular mean is caused by the PLL in the SDR and signal propagation through cable and the SDR. Increasing spread of data points with increasing signal level is not considered to be significant, because data is portrayed in linear scale and if logarithmic scale was used spread would

decrease with increasing signal level.



(a) (b)

Figure 23: Dependency of circular mean on I component amplitude clipping at different signal levels: L5 1176.45 MHz signal (a) and L1 1575.42 MHz signal (b).

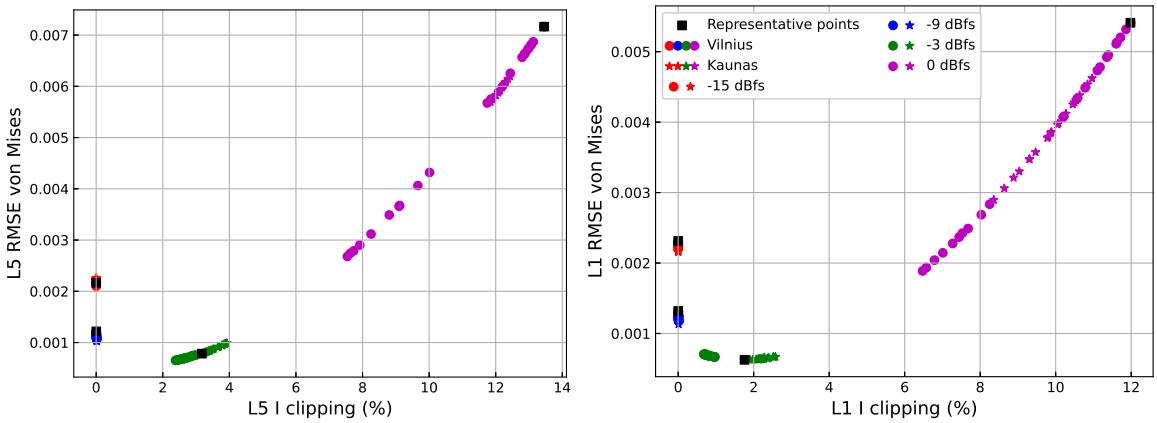


(a) (b)

Figure 24: Dependency of von Mises distribution κ parameter on I component amplitude clipping at different signal levels: L5 1176.45 MHz signal (a) and L1 1575.42 MHz signal (b).

IQ clipping influence on von Mises κ parameter is illustrated in figure 24. As was implied during statistical results discussion, with increasing clipping κ decreases. However, values of this parameter are smaller for L5 (Fig. 24b) compared to L1 (Fig. 24b). This means that L1 will always have more pronounced clustering around circular mean than L5. This is confirmed by theoretical distributions in figure 27. All distribution plots were calculated using representative records indicated by black squares.

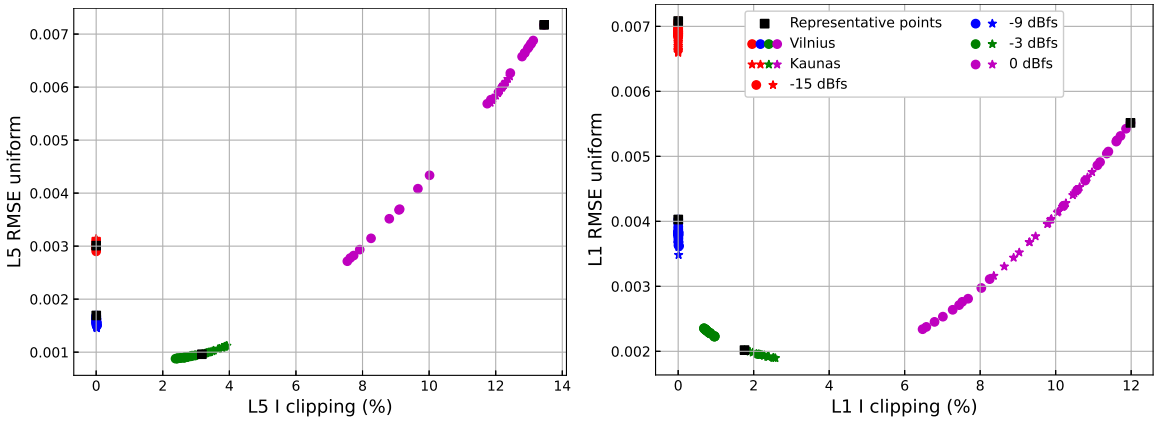
This also shows, that despite clipping, with increasing frequency clustering, also, becomes more pronounce, because L1 has higher center frequency than L5. The same trend was observed between L1 frequencies in previous results. This behavior is, also, expected to be caused by the SDR



(a) (b)

Figure 25: Dependency of root mean square error, when comparing record phase PDF to von Mises PDF, on I component amplitude clipping at different signal levels: L5 1176.45 MHz signal (a) and L1 1575.42 MHz signal (b).

platform used for this work.



(a) (b)

Figure 26: Dependency of root mean square error, when comparing record phase PDF to uniform PDF, on I component amplitude clipping at different signal levels: L5 1176.45 MHz signal (a) and L1 1575.42 MHz signal (b).

Situation with κ parameter would lead one to believe that with decreasing clipping von Mises distribution RMSE value would decrease and uniform distribution RMSE value would increase, however as indicated by figures 25 and 26, this is not the case. For both distributions, and both L5 (Fig. 25a and 26a) and L1 (Fig. 25b and 26b) signals error values increases as clipping both increases and decreases. This result is very clearly illustrated in distribution plots in figure 27. One can see that the reason for such behavior is another distortion caused SDR, specifically by quadrature sampling. Distribution plot have periodic peaks at multiples of $\frac{\pi}{2}$.

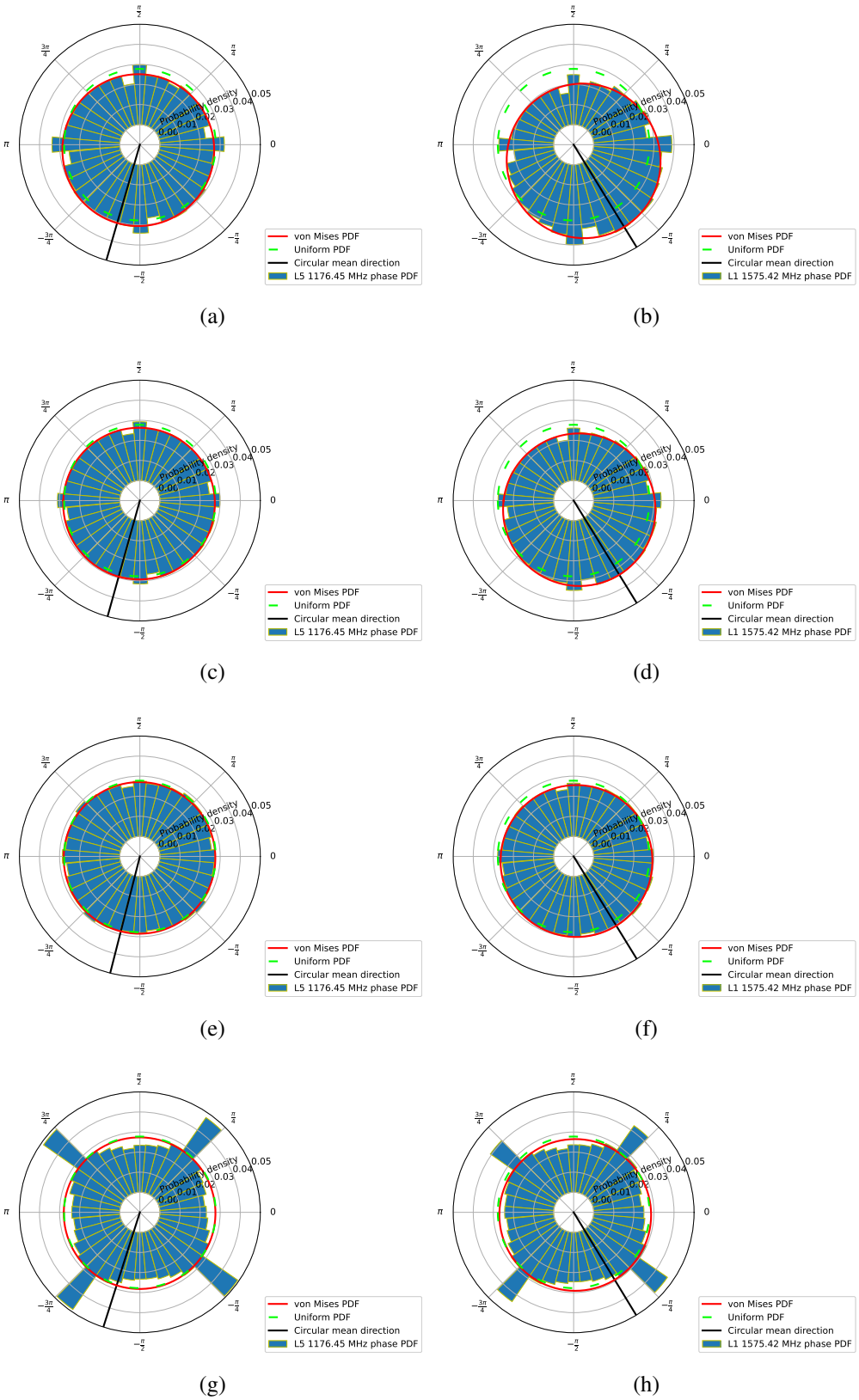


Figure 27: GNSS L5 1176.45 MHz record (a, c, e, g) and L1 1575.42 MHz record (b, d, f, h) phase probability density functions at different signal levels: -15 dBfs (a, b), -9 dBfs (c, d), -6 dBfs (e, f) and 0 dBfs (g, h).

The statistical importance of these peaks increases at lower and higher clipping levels as indicated by Figs. 27a, 27b, 27g, 27h. While at -3 dBfs signal level (Figs. 27e and 27f) statistically peaks caused by the SDR are no more important than the rest of the possible phases. This signal level coincides with lowest RMSE values for both distributions as well. Note, at all signal levels von Mises distribution is a better fit, however visually it is very clear that this only the case for small values of amplitude clipping where statistical importance of SDR peaks is comparable with the rest of the possible phases. Also, figures 25 and 26 signify the existence of the optimal clipping level at which SDR impact on phase distribution is minimal. In other words, optimal amplitude clipping mitigates the distortions caused by SDR from statistical point of view. Therefore, RMSE dependencies on clipping at every frequency of interest could be used as calibration curves to determine the optimal signal level.

4.4 GNSS L1+L5 record bandwidth measurements

In order to investigate quality of GNSS recordings and determine new possibilities of stability improvement for proposed GNSS L1+L5 record–replay simulator, dependencies of average carrier to noise ratio C/N_0 and the number of visible satellites on GNSS record bandwidth were measured. These two parameters were chosen, because they directly reflect the quality of GNSS signals, and they are mutually complementary, since it was observed during experimentation, that large values of C/N_0 can be measured when number of visible satellites is too small for positioning and, vice versa, large number of visible satellites are possible when C/N_0 is too small for reception. The bandwidth is an important factor, since it directly affects the reception quality, as insufficient bandwidth may cause data loss. However, smaller bandwidth means lower sampling rate and smaller load on host computer resulting in better stability of the recording system.

Measured results were analyzed considering all main GNSS systems as a whole for both L1 and L5 bands and also for each main GNSS system separately. When referring to all systems GNSS acronym will be used and when referring to a specific system, its name will be used.

For GNSS L1 band, dependency of C/N_0 on GNSS L1 recording bandwidth is shown in Fig. 28a and the number of visible satellites in Fig. 28b. In these figures, dependencies vary significantly when comparing receiver to receiver. However, all curves behave similarly after crossing down 4 MHz since at smaller bandwidths sharp drop in both average carrier to noise ratio and the number of visible satellites is observed.

Regardless of this behavior of measured values, all receivers are able to determine position even at 1 MHz record bandwidth, because C/N_0 and the number of satellites is sufficient. u-blox

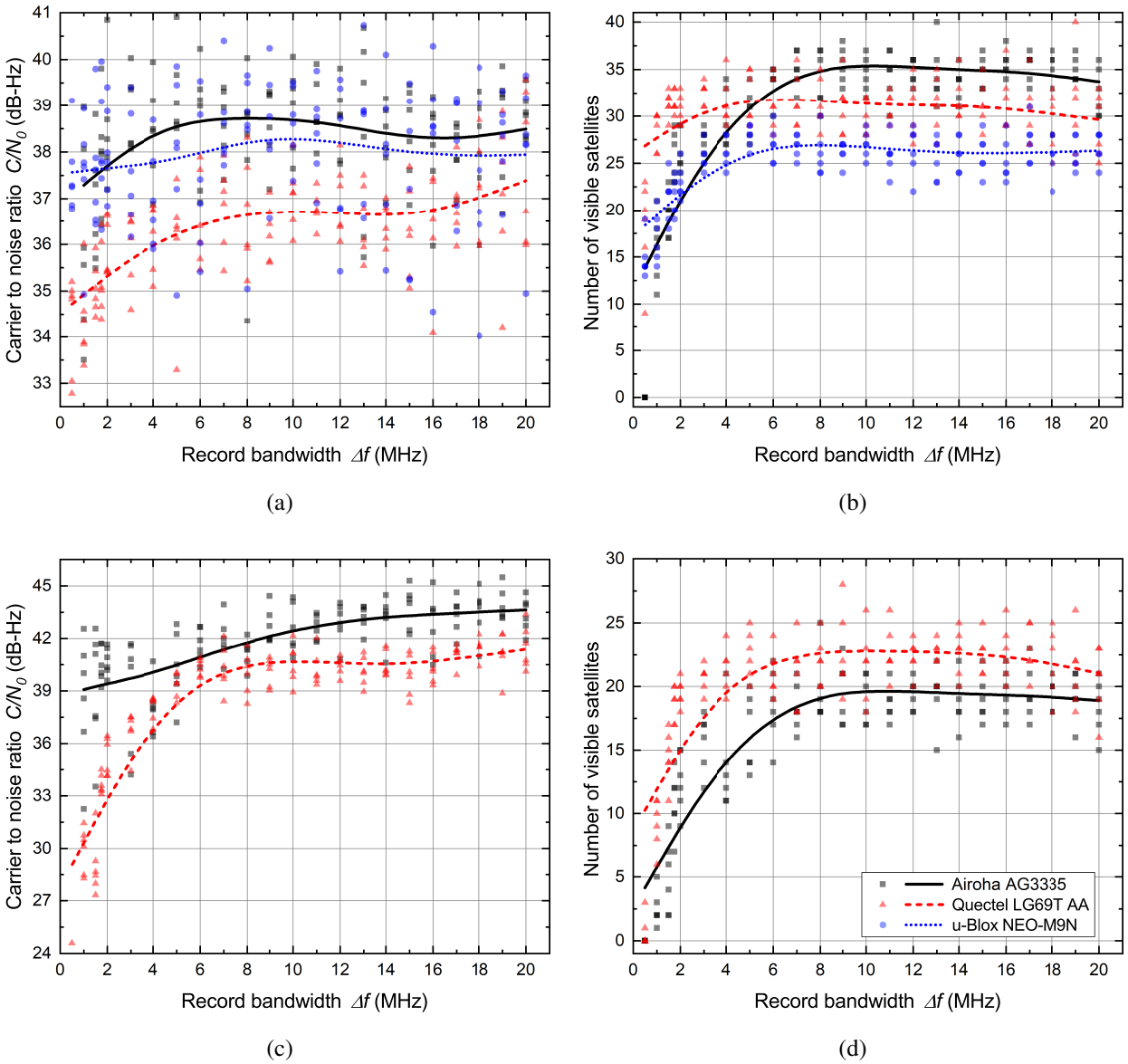


Figure 28: Dependency of carrier to noise ratio (left column) and the number of visible satellites (right column) on the GNSS record bandwidth over L1 (top row) and L5 (bottom row) bands for three GNSS receivers shown in the common legend. Points are measured values and lines correspond to statistically averaged results.

NEO-M9N receiver is able to determine position even at 0.5 MHz bandwidth. Despite absolute values being different comparing receiver to receiver, the general behavior of the curves is very similar. Therefore, it is likely that other receivers will show similar results.

For GNSS L5 case, dependency of C/N_0 on GNSS L5 recording bandwidth is shown in Fig. 28c and dependency of the number of visible satellites on bandwidth is shown in Fig. 28d. For L5 measurements, only Airoha AG3335 and Quectel LG69T AA GNSS receivers were used, because u-blox NEO-M9N is L1 only. From these results it is observed that while the number of visible

satellites dependencies show similar behavior, C/N_0 curves are very different. These differences are caused by implementation differences between receivers, which descriptions are not publicly available. Overall, one can see that any GNSS L5 signal can be recorded using no more than 6 MHz of bandwidth, because only after this limit sharp decline in measured values begins.

Now, considering only GPS L1 (Figs. 29a and 29b), the situation differs significantly compared to GNSS L1. There is almost no relation between recording bandwidth and the number of visible satellites and only some variation in case of C/N_0. The drop in values occurs only after crossing down 4 MHz bandwidth.

This result is expected, because despite necessity of almost 15 MHz bandwidth to fully record GPS TMBOC(6,1,1/11) signal, only pilot signals are transmitted in the range between 4 MHz and 15 MHz. These pilot signals use BOC(6,1) modulation, and they do not carry any information and are mostly used for channel state estimation. After crossing down 4 MHz limit, data loss occurs and GPS L1C data signals, which use BOC(1,1) modulation, no longer can be fully recorded, however GPS L1 C/A signals need only 2 MHz due to BPSK(1) modulation, therefore even at 1 MHz bandwidth GPS satellites are still visible by all receivers with sufficient signal quality for navigation.

Considering GPS L5 band (Figs. 29c and 29d), curve behavior is comparable to GNSS L5 case, except for the fact that according to dependency of the number of visible satellites, only 4 MHz of bandwidth is needed before sharp drop in values occurs and according to C/N_0 dependency, 6 MHz of bandwidth is needed. Unfortunately, such behavior cannot be explained by signal structures and modulations, because GPS use BPSK(10). This means that to record this signal without data loss one needs 20.46 MHz of bandwidth and after crossing 20 MHz the situation should be similar to what was observed in GPS L1 after crossing 4 MHz.

The most likely explanation is that observed behavior with BPSK(1) and 1 MHz of bandwidth, should be possible in case of BPSK(10) and 10 MHz of bandwidth as well. Further reduction of bandwidth is possible due to utilization of L1 acquisition results in L5 acquisition. However, this cannot be confirmed due to unavailability of documentation from manufacturers.

Situation in GALILEO L1 case (Figs. 30a and 30b) is very similar to GPS case, because GALILEO E1 signals use CBOC(6,1,1/11) modulation, which is similar to TMBOC(6,1,1/11), however data is carried by both BOC(1,1) and BOC(6,1). Despite this, there is almost no change in C/N_0 dependency and even an improvement with shrinking bandwidth in the number of satellites dependency can be seen in the range from 4 MHz to 15 MHz. It is believed that this behavior is observed due to the fact that majority of signal power is allocated to BOC(1,1) [14]. Also, as with GPS L1 case, a decline of values after bandwidth becomes smaller than 4 MHz can be observed, and the decline is sharper than in GPS L1 case, since GALILEO L1 does not have BPSK(1) signals.

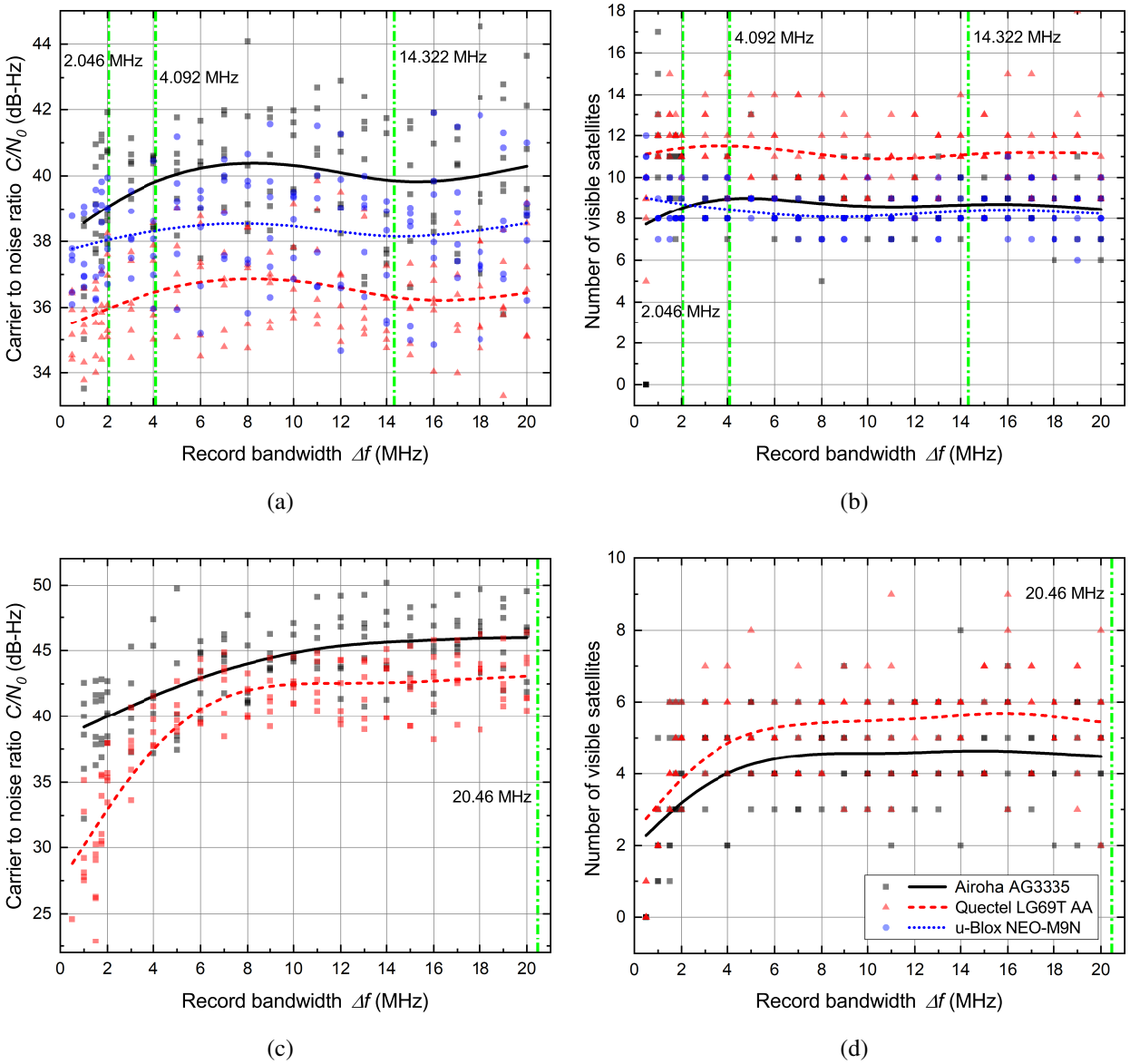


Figure 29: Dependency of carrier to noise ratio (left column) and the number of visible satellites (right column) on the GPS record's bandwidth over L1 (top row) and L5 (bottom row) bands for three GNSS receivers shown in the common legend. Points are measured values and lines correspond to statistically averaged results. Vertical green lines in this and the following figures indicate typical bandwidths of L1 and L5 signal modulations.

GALILEO L5 (Figs. 30c and 30d) dependencies are similar to GNSS L5 situation. Sharp drop in values starts at 6 MHz, what is less than 30% of the necessary bandwidth to demodulate GALILEO E5a signal. But since its main lobe can be compared to and demodulated as BPSK(10) [14], measured curves behave similarly to those observed in GPS L5 case.

In BEIDOU system case (Figs. 31a and 31b) one can observe dependency on both the bandwidth and receiver type. For Quectel LG69T AA and u-Blox NEO-M9N receivers, one can observe

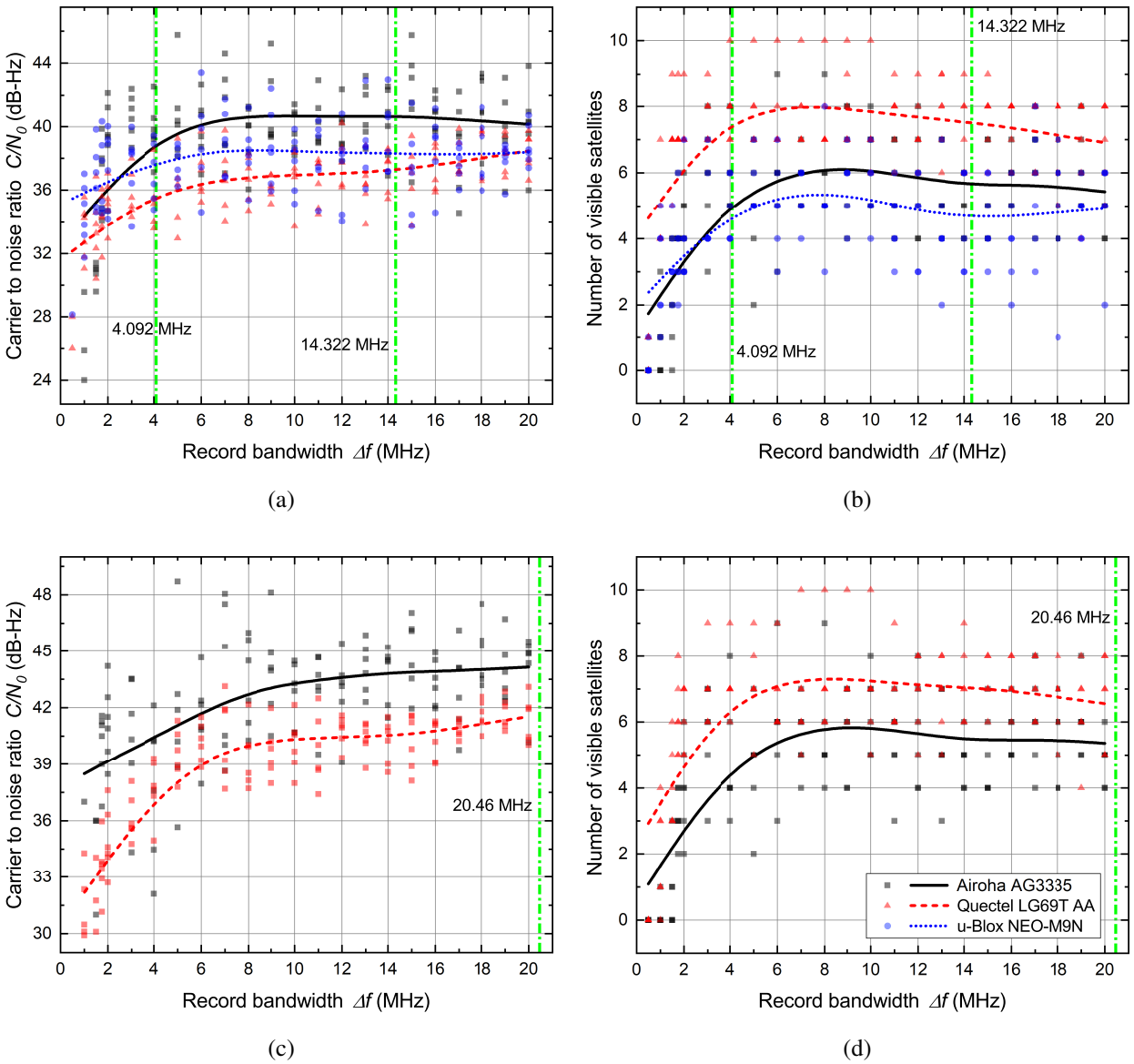


Figure 30: Dependency of carrier to noise ratio (left column) and the number of visible satellites (right column) on the GALILEO record's bandwidth over L1 (top row) and L5 (bottom row) bands for three GNSS receivers shown in the common legend. Points are measured values and lines correspond to statistically averaged results.

that curve behavior is similar to GPS L1 case. The reason for this is that BEIDOU B1C uses QMBOC(6,1,4/33), which is comparable to TMBOC(6,1,1/11) and also BOC(6,1) is used only for pilot signals. Additionally, BEIDOU transmits B1I signal, which uses BPSK(2) modulation. What is interesting, is that, just as BOC(1,1), BPSK(2) needs 4.092 MHz of bandwidth, but unlike BOC(1,1), and much more like BPSK(1), the drop of C/N_0 values starts only when record bandwidth is reduced below 4 MHz, but it is significantly less sharp than in GALILEO case. It is likely that BPSK(2) has a similar effect on this dependency as BPSK(1) in GPS case.

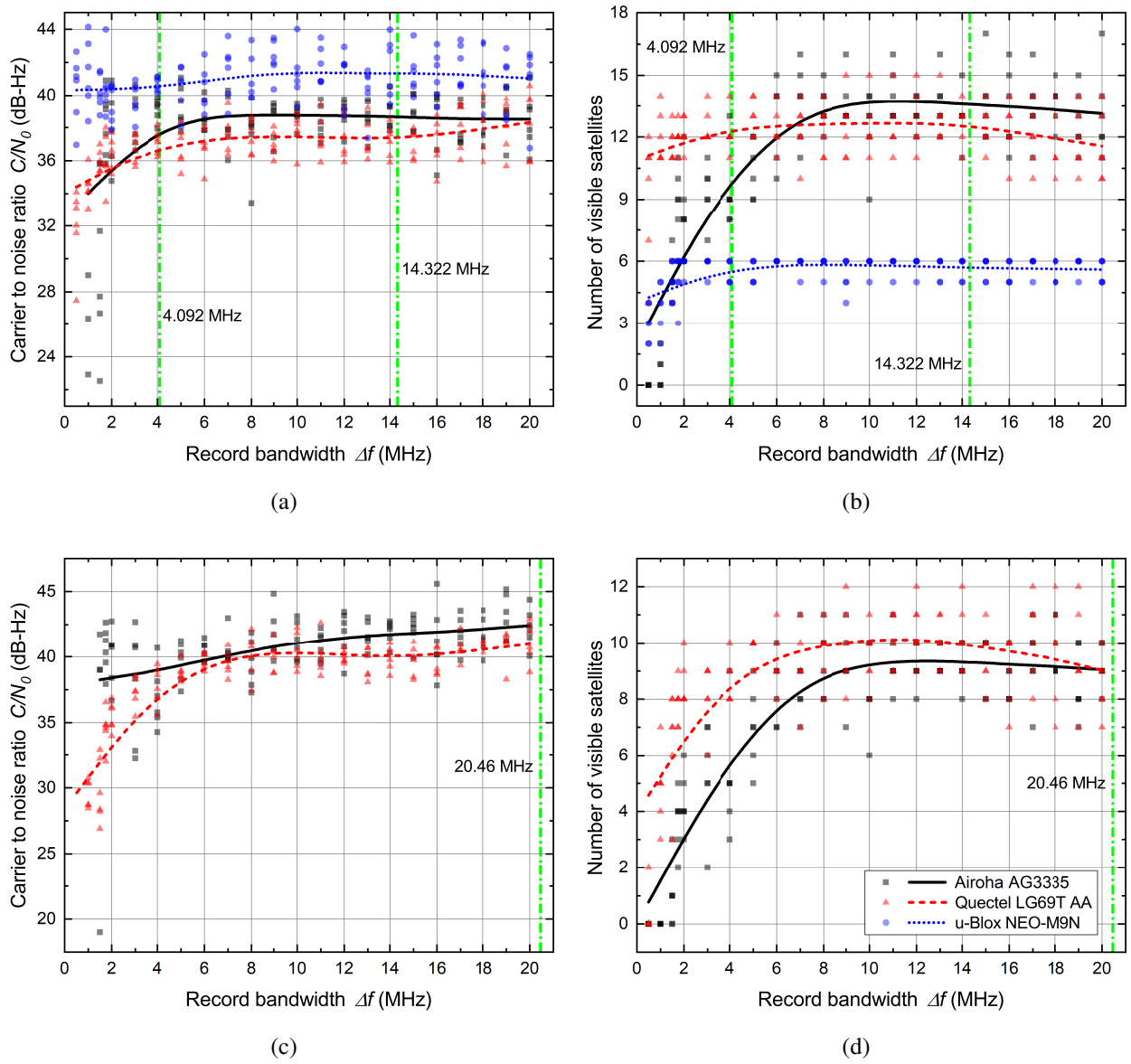


Figure 31: Dependency of carrier to noise ratio (left column) and the number of visible satellites (right column) on the BEIDOU record's bandwidth over L1 (top row) and L5 (bottom row) bands for three GNSS receivers shown in the common legend. Points are measured values and lines correspond to statistically averaged results.

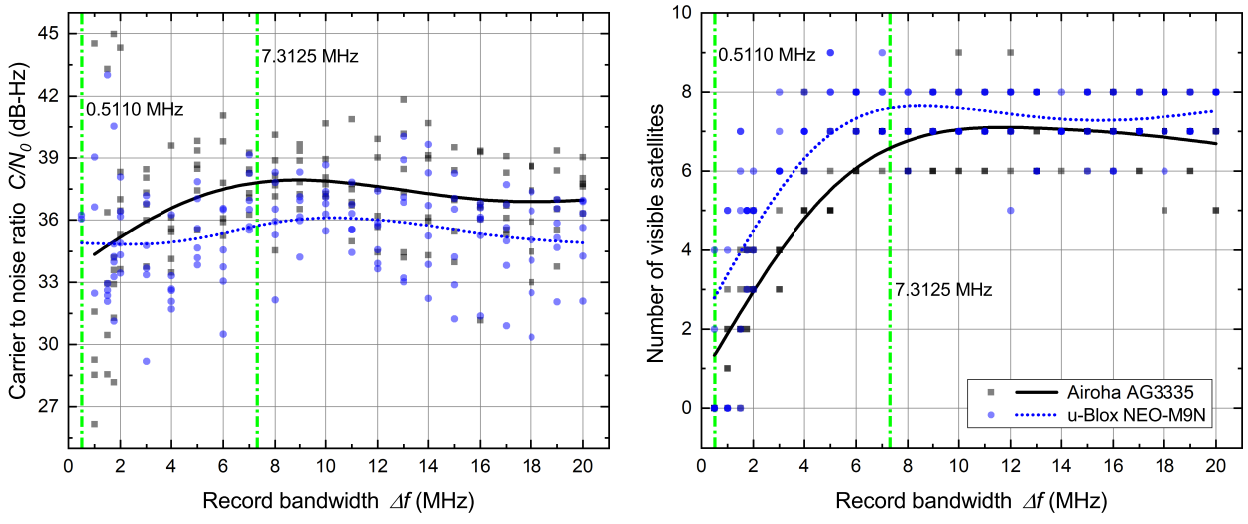
On the other hand, Airoha AG3335 receiver shows a very different behavior. One can observe in Fig. 31b that significant drop in the number of satellites begins after crossing down 6 MHz, which is comparable to what is observed in L5 cases. However, even at 4 MHz receiver is still able to see around nine satellites, which is sufficient for navigation. Therefore, bandwidth can be reduced to as little as 4 MHz, which agrees with other receiver results.

BEIDOU L5 band situation (Figs. 31c and 31d) is similar to GPS and GALILEO systems. Sharp decline is observed after crossing down 6 MHz, because, as GPS, BEIDOU B2a signal uses

BPSK(10) modulation, and as discussed before, receivers use information acquired via L1 for L5 signal processing.

Overall, it should be said that the number of visible satellites is a superior parameter, because it is less affected by GNSS receiver differences and reports similar relations across all GNSS systems. However, this parameter can imply that Quectel LG69T AA is a superior receiver, because it can detect more GNSS satellites at all tested bandwidths for almost each GNSS system. However, this assumption is incorrect, because it was observed, that without L1 Quectel LG69T AA cannot detect a single L5 satellite and L1 has to be present all the time, while Airoha AG3335 only needs L1 at the beginning and, after satellites are detected, L1 is no longer needed for L5 tracking and thus L5 can be used independently. But as far as L1 is concerned, Quectel LG69T AA detects more satellites, since while testing it only worked with GPS, GALILEO and BEIDOU systems, but the total number of visible satellites is comparable to other receivers as seen in Fig. 28b.

The last system to discuss is GLONASS. This system uses FDMA instead of CDMA for multiple access, which causes very different behavior of C/N_0 curves compared to other GNSS systems as shown in Fig. 32a. Only small variations can be observed, which are most likely caused by changing reception conditions, because of a four hour difference between measurements using 2 MHz and 20 MHz bandwidths. This behaviour is explained by the fact that as a FDMA system GLONASS needs almost 8 MHz of bandwidth for its 14 channels, but each channel transmits only BPSK(0.5) modulated signal, which needs only 0.511 MHz of bandwidth.



(a) (b)

Figure 32: Dependency of carrier to noise ratio (a) and the number of visible satellites (b) on the GLONASS record's bandwidth over L1 band for two GNSS receivers shown in the common legend. Points are measured values and lines correspond to statistically averaged results.

The effects of FDMA are clearly shown in number of visible satellites graph (Fig. 32b). There is no dependency on record bandwidth while the bandwidth stays above 8 MHz. After moving below this limit, the number of visible satellites starts to gradually decrease until only one satellite is visible at 0.5 MHz, because only one GLONASS FDMA channel can fit in 0.5 MHz. It should be mentioned, that if the bandwidth is limited to 6 MHz as in GNSS L5 case, there are still more than six available GLONASS satellites, which is sufficient to determine position using GLONASS signals alone.

4.5 Dynamic testing of GNSS L1+L5 record–replay simulator

After optimal SDR conditions have been determined, proposed GNSS L1+L5 record–replay simulator was optimized as a whole by reducing record bandwidth without observable quality degradation. When it was clear that it can work in stationary conditions, dynamic testing while driving a car was performed. GNSS Signals were logged by Airoha AG3335, Quectel LG69T AA and u-blox NEO-M9N GNSS receivers and recorded by simulator simultaneously. After finishing the whole circuit shown in Fig. 33, GNSS records were replayed and logged by receivers. Then, logs from direct reception and from simulator were compared.

Additionally, in order to evaluate viability of bandwidth reduction in dynamic situations, the sampling rate for all signals was set to 12 MHz. As discussed before, 12 MHz is 3 times the minimum bandwidth for L1 and 2 time the minimum bandwidth for L5 signal. This testing is important because sampling rate reduction allows to make longer recordings, while system performs in a more stable manner.

Log analysis has shown that all receivers behaved similarly during testing, therefore only measurements produced by Airoha AG3335 receiver will be discussed. Figure 33 show that despite different signal origin, GNSS receiver produces very similar trajectory results for both direct reception signals and recorded using SDRs signals. The differences can only be observed at small radius corners as can be seen in zoomed–in windows on the map, but these differences are not caused by simulator but by nearby buildings and vegetation, which in turn block clear line of sight between satellites and antenna or introduce reflections, therefore, degrading record quality.

Furthermore, Both L1 and L5 signals were successfully recorded and replayed. This is evident in C/N_0 and the number of visible satellites dependencies over time (Figs. 34a and 34b). Here, as well, it is evident that despite different signal origin both direct reception and replayed signals have similar curves, which display similar evolution over time. The only difference is that, in direct satellite signal case, reported number of L1 satellite signals is higher and also C/N_0 values are also slightly higher for both L1 and L5 compared to replayed signals.

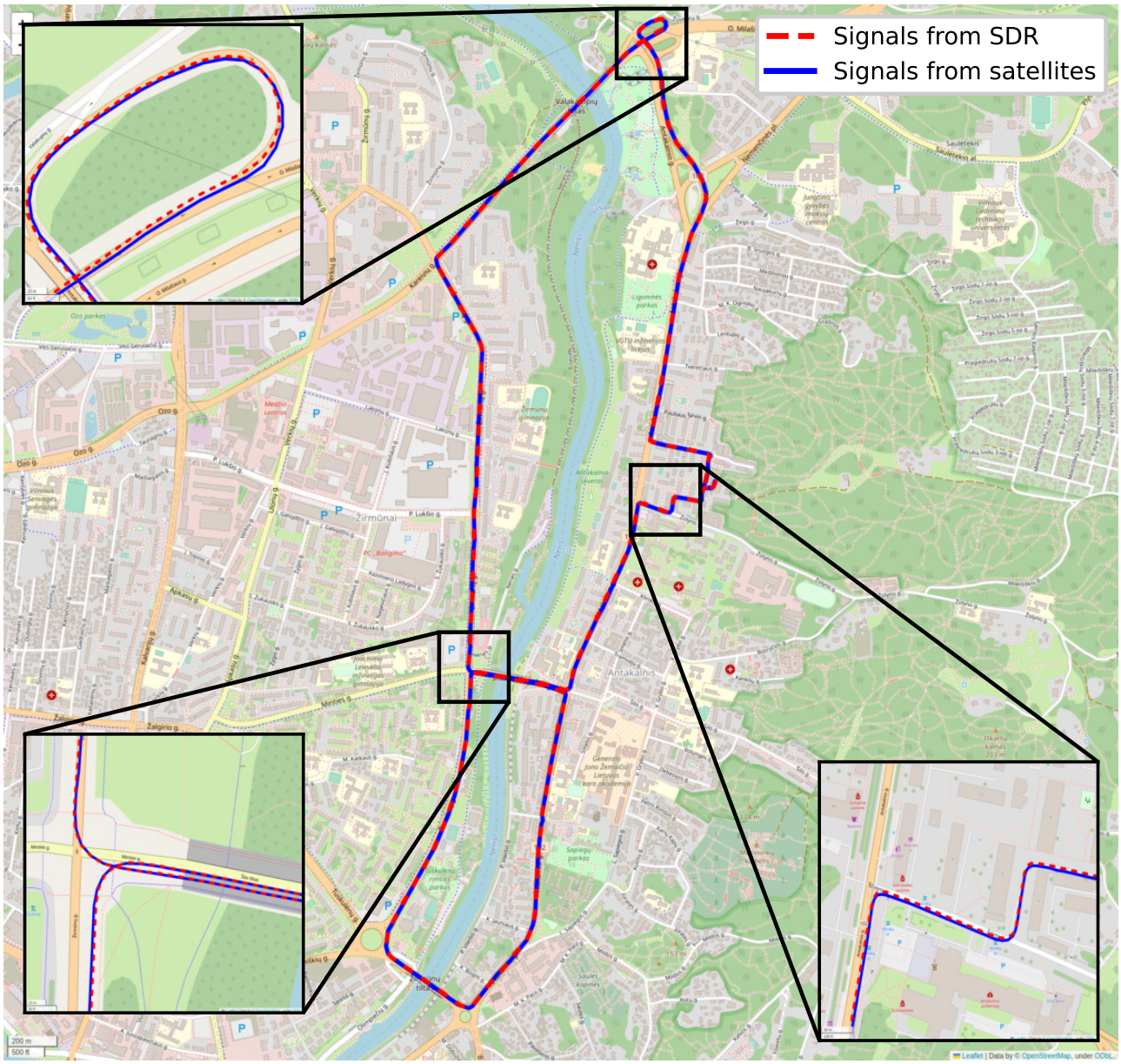


Figure 33: Route of dynamic testing as logged by Airoha AG3335 GNSS receiver for signals obtained directly from satellite and recorded by SDRs.

Since Airoha AG3335 needs L1 signal only in the beginning, L5 only mode was tested using this receiver together with 12 MHz sampling rate for L5 recording. This test is not possible with Quectel LG69T AA receiver due to necessity of L1 being present all the time. The same can be said about uBlox NEO-M9N reciever, since it is only capable of L1 reception. To perform L5 only test, first all L1 and L5 signals are replayed and once L5 satellite signals are detected, L1 playback is stopped.

Results in Figs. 34a and 34b show that even in L5 only mode Airoha AG3335 GNSS receiver is able to perform positioning tasks since sufficient number of satellites is detected. Of course, the number of visible satellites is more than 4 times lower compared to signals directly from satellites

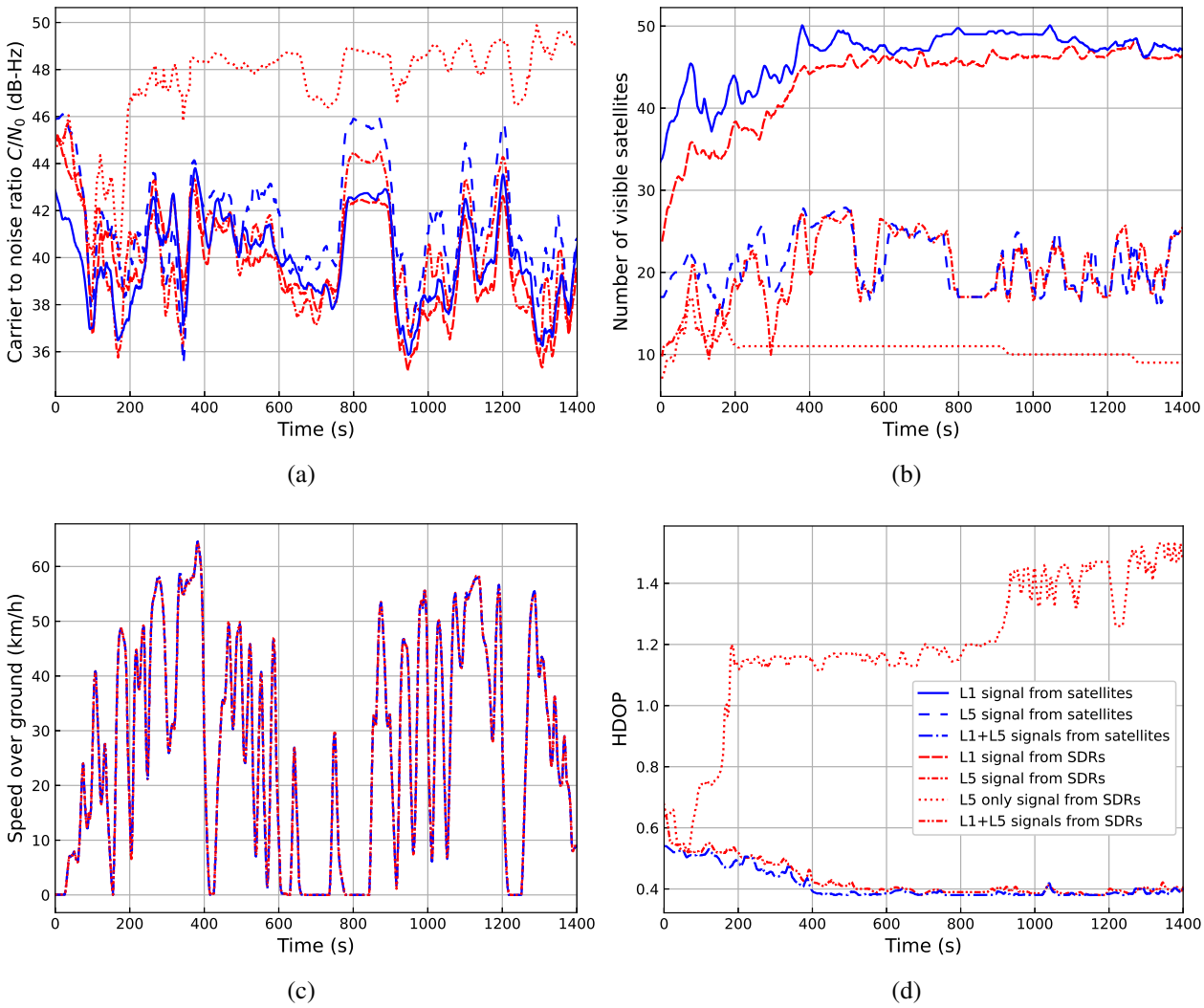


Figure 34: Dependency over time of carrier to noise ratio (a), the number of visible satellites (b), receiver speed over ground (c) and HDOP for L1, L5 and L1+L5 signals received directly from satellites and replayed by SDRs as shown in the common legend. Data logged using Airoha AG3335 GNSS receiver.

or during simultaneous L1 and L5 playback, because in L5 only mode receiver works with the same satellites, which were detected while L1 was present for the whole duration of the test without detecting any new signals.

The effect of low number of visible satellites is clearly visible on Horizontal Dilution of Precision (HDOP) dependency on time (Fig. 34d), since over time HDOP only rises in L5 only case compared to direct satellite reception and L1+L5 record playback where HDOP improves with rising number of satellites and remains below 0.5 for the majority of the test. This is an expected behavior, because in L5 only mode number of satellites detected by the receiver is close to theoretical minimum necessary for trilateration.

As for speed over ground signal origin and different receiver operation modes have no differ-

ence as shown in Fig. 34c. As mentioned before, L5 signals were recorded with limited bandwidth during all tests, therefore simulators behavior during high speed situations was important. After additional testing where driving speed was up to 130 km/h no changes in performance ware observed. Therefore, simulator usage is not limited by speed.

Lastly, it should be stressed that L5 record used in L5 only testing was recorded using only 12 MHz bandwidth, which in turn means that a signal was captured with a data loss. 12 MHz is almost 2 times less than necessary for BPSK(10) modulated signal. This proves that band–limited L5 record can be used in standalone mode without L1 signals. Also, worse performance results of L5 only testing are not caused by band–limiting, but by constrain of GNSS receiver design, which prevent acquisition of new signals without L1 signals.

5 Main Results and Conclusions

In this master thesis, overview of literature and extensive experimental work have enabled appropriate configuration and development of hardware, as well as, the creation of software necessary for successful operation of SDR based GNSS L5+L1 record-replay simulator. Testing has shown that thermal stabilization of discrete SDRs and synchronization equipment during record and replay is imperative to capture and maintain stable relation between numbers of visible satellites and carrier to noise ratios of L1 and L5 signals as well as mitigated sudden disappearances of L5 signals during playback. Also, data losses during long term sustained recording operations can be prevented only if computer is configured as a real-time machine.

Analysis of recordings using circular statistics allowed to statistically characterize GNSS signals. Recorded GNSS signals share many similarities with complex white noise and have phase distributions, which can be well modeled using von Mises distribution. Phase distribution can be influenced by the amplitude clipping of the input signal, but appropriate clipping mitigates phase distribution distortions caused by quadrature sampling of SDRs.

Generic SDR based simulation approach puts large load on SDR host computer, therefore, ways to increase system stability were explored. One of explored optimizations was recording of GNSS signals using lower than necessary sampling rate. The results show that any GNSS L1 signal can be successfully recorded using sampling rate as low as 4 MHz. In case of L5, any GNSS L5 signal can be recorded using sampling rate of 6 MHz.

Regardless of absolute lower limits of sampling rate, it is imperative, that all recorded signals would use the same sampling rate, when L1 and L5 are being recorded simultaneously. If this condition is not satisfied, GNSS receivers do not recognize L5 signals, only L1. Multiple sampling rates are only viable, when signals are being recorded in the same band.

Results obtained during dynamic testing show close overlap of all performance metrics like trajectory, speed, number of visible satellites or carrier to noise ratio when comparing direct reception signals with signals produced by SDRs. This implies that from point of view of GNSS receiver signal origin is indistinguishable and records produced by proposed GNSS L1+L5 simulator can be used for equipment testing purposes.

Overall, results produced in this work show that the goal of this master thesis (to create a GNSS L1+L5 record-replay simulator, which can simulate L1 and L5 signals, and is affordable and portable) is achieved and also allow making the fallowing conclusions:

1. Computer configured for real–time applications is essential for long term (more than 20 minutes) sustained record–replay operations, because otherwise momentary data losses occur in a

GNSS recording effectively making it unusable for testing purposes.

2. SDR caused signal phase distribution distortions can be compensated by choosing an appropriate input signal amplitude clipping level, because clipping caused distortion of phase statistics mitigates SDR caused distortions.

3. GNSS signals can be recorded using significantly lower bandwidth than necessary, because GNSS L5 signals can be recorded using more than 3 times and GNSS L1 using almost 4 time smaller bandwidth and recordings would still have enough information for receiver to determine position.

4. To record multiple GNSS signals in the same band, different sampling rates for different center frequencies can be used, because the loss of synchronization is not important. However, to record multiple GNSS bands, the same sampling rate must be used for all center frequencies, because synchronization is imperative.

5. Thermal stabilization of discrete SDRs and synchronization equipment allows to synchronize temperature and noise related processes in the subsystems of the whole system, because when recording and replaying noise-like signals at the same constant temperature, dependencies between parameters of synchronous signals are captured and maintained.

References

[1] M. Hasan, "State of iot 2022: Number of connected iot devices growing 18% to 14.4 billion globally," 2022. https://iot-analytics.com/number-connected-iot-devices/#:~:text=In%202021%2C%20IoT%20Analytics%20expects,to%2012.3%20billion%20active%20endpoints.. Accessed 2023-05-15.

[2] P. Steigenberger, S. Thoelert, and O. Montenbruck, "Gnss satellite transmit power and its impact on orbit determination," *Journal of Geodesy*, vol. 92, no. 6, pp. 609–624, 2018.

[3] C. Jeffrey, *An introduction to GNSS: GPS, GLONASS, GALILEO and other global navigation satellite systems*. NovAtel, 2015.

[4] China Satellite Navigation Office, "BeiDou navigation satellite system signal in space interface control document open service signal B1I (Version 3.0)," 2019.

[5] N. Vagle, A. Broumandan, A. Jafarnia-Jahromi, and G. Lachapelle, "Performance analysis of gnss multipath mitigation using antenna arrays," *The Journal of Global Positioning Systems*, vol. 14, no. 1, pp. 1–15, 2016.

[6] Racelogic, "LabSat 3 Wideband," 2023. https://www.labsat.co.uk/index.php/en/products/labsat-3-wideband. Accessed on 2023-05-15.

[7] T. Ebinuma, "Software-Defined GPS Signal Simulator," 2023. https://github.com/osqzss/gps-sdr-sim. Accessed on 2023-05-15.

[8] J. Chen, S. Zhang, H. Wang, and X. Zhang, "Practicing a record-and-replay system on usrp," in *Proceedings of the second workshop on Software radio implementation forum*, pp. 61–64, 2013.

[9] R. Di, S. Peng, S. Taylor, and Y. Morton, "A usrp-based gnss and interference signal generator and playback system," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pp. 470–478, IEEE, 2012.

[10] A. Hennigar, *Analysis of Record and Playback Errors of GPS Signals Caused by the USRP*. PhD thesis, 2014.

[11] D. Pascual, "Sdanipascual/GNSS-matlab, GitHub," 2023. https://github.com/danipascual/GNSS-matlab. Accessed on 2023-05-15.

[12] GPS Directorate, "IS-GPS-200M, Navstar GPS Space Segment/User Segment Interfaces," 2021.

[13] GPS Directorate, "IS-GPS-800H, Navstar GPS Space Segment/User Segment L1C Interfaces," 2021.

[14] European Union, *European GNSS (Galileo) open service: Signal in space interface control document*. Office for Official Publications of the European Communities, 2021.

[15] China Satellite Navigation Office, "Beidou navigation satellite system signal in space interface control document open service signal b2a (version 1.0)," 2017.

[16] China Satellite Navigation Office, "BeiDou navigation satellite system signal in space interface control document open service signal B1C (Version 1.0)," 2018.

[17] Russian Institute of Space Device Engineering, "Glonass interface control document," 2008.

[18] GPS Directorate, "IS-GPS-750H, Navstar GPS Space Segment/User Segment L5 Interfaces," 2021.

[19] G. W. Hein, J.-A. Avila-Rodriguez, S. Wallner, A. R. Pratt, J. Owen, J.-L. Issler, J. W. Betz, C. J. Hegarty, L. S. Lenahan, J. J. Rushanan, *et al.*, "Mboc: the new optimized spreading modulation recommended for galileo l1 os and gps l1c," vol. 2006, pp. 884–892, 2006.

[20] Z. Yao, M. Lu, and Z. Feng, "Quadrature multiplexed boc modulation for interoperable gnss signals," *Electronics letters*, vol. 46, no. 17, pp. 1234–1236, 2010.

[21] J. A. A. Rodriguez, *On generalized signal waveforms for satellite navigation*. PhD thesis, Universitätsbibliothek der Universität der Bundeswehr München, 2008.

[22] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, *A software-defined GPS and Galileo receiver: a single-frequency approach*. Springer Science & Business Media, 2007.

[23] H. Hu and N. Wei, "A study of gps jamming and anti-jamming," in *2009 2nd international conference on power electronics and intelligent transportation system (PEITS)*, vol. 1, pp. 388–391, IEEE, 2009.

[24] M. A. Richards, "The discrete-time fourier transform and discrete fourier transform of windowed stationary white noise," *Georgia Institute of Technology, Tech. Rep*, 2013.

[25] S. R. Jammalamadaka and A. SenGupta, *Topics in circular statistics*, vol. 5. world scientific, 2001.

[26] K. V. Mardia, P. E. Jupp, and K. Mardia, *Directional statistics*, vol. 2. Wiley Online Library, 2000.

[27] N. I. Fisher, *Statistical analysis of circular data*. Cambridge University Press, 1995.

[28] G. S. Watson, *Statistics on spheres*. Wiley-Interscience, 1983.

[29] C. Brunsdon and J. Corcoran, "Using circular statistics to analyse time patterns in crime incidence," *Computers, Environment and Urban Systems*, vol. 30, no. 3, pp. 300–319, 2006.

[30] D. Jepson, "RF System Synchronization – Baseband," 2019. https://www.gnuradio.org/grcon/grcon19/presentations/Synchronization_Core_Concepts_and_Applications/Daniel%20Jepson%20-%20Baseband%20Synchronization.pdf. Accessed 2023-05-15.

[31] M. Krueckemeier, F. Schwartau, C. Monka-Ewe, and J. S. Technische, "Synchronization of multiple usrp sdrs for coherent receiver applications," in *2019 Sixth International Conference on Software Defined Systems (SDS)*, pp. 11–16, IEEE, 2019.

[32] J. Leclère, R. Landry, and C. Botteron, "Comparison of l1 and l5 bands gnss signals acquisition," *Sensors*, vol. 18, no. 9, p. 2779, 2018.

[33] C. Tröster-Schmid and T. Bednorz, "Generating multiple phase coherent signals–aligned in phase and time," *Rohde & Schwarz GmbH & Co. KG, München*, 2016.

[34] D. Baker and B. Avenell, "RF System Synchronization – LO's," 2019. https://www.gnuradio.org/grcon/grcon19/presentations/Phase_Synchronization_Techniques/Dan%20Baker%20-%20Phase%20Synchronization%20Techniques.pdf, accessed 2023-05-15.

[35] M. Laakso, R. Rajamäki, R. Wichman, and V. Koivunen, "Phase-coherent multichannel sdr-sparse array beamforming," in *2020 28th European Signal Processing Conference (EUSIPCO)*, pp. 1856–1860, IEEE, 2021.

[36] T. Langley and R. Kowalczyk, "Introduction to intel architecture: The basics," *White paper*, 2009.

[37] Acroname, "How many usb devices can i connect?," 2019. https://iot-analytics.com/number-connected-iot-devices/#:~:text=In%202021%2C%20IoT%20Analytics%20expects,to%2012.3%20billion%20active%20endpoints.. Accessed 2023-05-15.

[38] M. Ossmann, "Hack RF." https://greatscottgadgets.com/hackrf/. Accessed 2023-05-15.

[39] FebSmart, "Fs-4c-u8s-pro." https://febsmart.com/showproduct.php?id=251. Accessed 2023-05-15.

[40] W1zzard, "Kingston a2000 1 tb m.2 nvme ssd review - 8firmware," 2020. https://www.techpowerup.com/review/kingston-a2000-1-tb-m-2-nvme-ssd/6.html. Accessed 2023-05-15.

[41] W1zzard, "Samsung 970 pro 512 gb review," 2018. https://www.techpowerup.com/review/samsung-970-pro-ssd-512-gb/6.html. Accessed 2023-05-15.

[42] Kingston, "A2000 nvme pcie ssd," 2022. https://www.kingston.com/datasheets/A2000_us.pdf. Accessed 2023-05-15.

[43] Samsung, "Samsung v-nand ssd 970 pro," 2018. https://semiconductor.samsung.com/resources/data-sheet/Samsung_NVMe_SSD_970_PRO_Data_Sheet_Rev.1.0.pdf. Accessed 2023-05-15.

[44] P. Koutoupis, "Rapiddisk," 2023. https://github.com/pkoutoupis/rapiddisk/. Accessed 2023-05-15.

[45] M. M. Madden, "Challenges using linux as a real-time operating system," in *AIAA Scitech 2019 Forum*, p. 0502, 2019.

[46] G. Sharma, D. Gauta, and A. Hussain, "Analysis of linux as real time operating system," in *Proceedings of International Conference on Advancements in Computing & Management (ICACM)*, 2019.

[47] A. Frade, "Xanmod kernel," 2023. https://xanmod.org/. Accessed 2023-05-15.

[48] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, "Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 6308–6325, 2020.

[49] PCI Geomatics Enterprises Inc., "Random trees classifier," https://catalyst.earth/catalyst-system-files/help/concepts/focus_c/oa_classif_intro_rt.html. Accessed 2023-05-15.

[50] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[51] T. Kavzoglu and I. Colkesen, “A kernel functions analysis for support vector machines for land cover classification,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 11, no. 5, pp. 352–359, 2009.

[52] S. Ozdemir, *Principles of data science*. Packt Publishing Ltd, 2016.

[53] L. Breiman, *Manual on setting up, using, and understanding random forests v3. 1*, 2002.

[54] S. P. Mishra, U. Sarkar, S. Taraphder, S. Datta, D. Swain, R. Saikhom, S. Panda, and M. Laishram, “Multivariate statistical data analysis-principal component analysis (pca),” *International Journal of Livestock Research*, vol. 7, no. 5, pp. 60–78, 2017.

[55] C. J. Hegarty, “Analytical model for gnss receiver implementation losses,” *Navigation*, vol. 58, no. 1, pp. 29–44, 2011.

[56] Canonical, “Ubuntu Server Operationg System,” 2023. `https://ubuntu.com/download/server`, Accessed 2022-05-15.

[57] S. Rota, J. Winkelmann, and N. Iwamatsu, “SLiM - Simple Login Manager for X11,” 2023. `https://github.com/axs-gentoo/slim-git`, Accessed 2022-05-15.

[58] XFCE Development Team, “Xfce Desktop Environment,” 2023. `https://www.xfce.org/`, Accessed 2022-05-15.

[59] GNU Radio Website, “GNU Radio,” 2023. `https://wiki.gnuradio.org/`, Accessed 2023-05-15.

[60] M. Bartolucci, J. A. del Peral-Rosado, R. Estatuet-Castillo, J. A. Garcia-Molina, M. Crisci, and G. E. Corazza, “Synchronisation of low-cost open source sdrs for navigation applications,” in *2016 8th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, pp. 1–7, IEEE, 2016.

[61] O. Kutkov, “1PPS distribution circuit,” 2021. `https://olegkutkov.me/2021/02/28/1pps-distribution-circuit/`, Accessed 2023-05-15.

[62] tvb, “picDIV – Single Chip Frequency Divider,” 2011. `http://www.leapsecond.com/pic/picdiv.htm`, Accessed 2023-05-15.

[63] S. Lourenço, "Módulo condicionador de sinal "SinCronos"," 2015. http://www.bloguetronica.com/2015/09/modulo-condicionador-de-sinal-sincronos.html, Accessed 2023-05-15.

[64] M. Razgūnas, S. Rudys, and R. Aleksiejūnas, "Gnss 2× 2 antenna array with beamforming for multipath detection," *Advances in Space Research*, vol. 71, no. 10, pp. 4142–4154, 2023.

[65] KrakenRF Inc, "KerberosSDR Quickstart Guide," 2022. https://www.rtl-sdr.com/ksdr/, Accessed 2023-05-15.

[66] I. M. Chakravarti, R. G. Laha, and J. Roy, *Handbook of methods of applied statistics*. Wiley, 1967.

[67] Scikit-learn developers, "Randon forest classfier." https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Accessed 2023-05-15.

[68] Scikit-learn developers, "Support vector classifier." https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC. Accessed 2023-05-15.

Santrauka

Daugiajuosčių GNSS signalų įrašymas ir atkūrimas naudojant programuojamo radijo įrangą

Autorius: Karolis Stankevičius

Dėl plataus daiktų interneto technologijų (IoT) paplitimo ir kitų elektronikos prietaisų, kurie turi žinoti savo poziciją, išaugo pasaulinės palydovinės navigacijos sistemos (GNSS) imtuvų testavimo poreikis. Testavimas gali būti atliktas naudojant GNSS simuliatorius, tačiau GNSS testavimo įrenginių kaina yra labai didelė ir IoT gamintojams ją sunku pateisinti.

Situacijai spręsti gali būti panaudota programuojamo radijo įranga (SDR), tačiau šiuo metu sukurti pasiūlymai sugeneruoja arba įrašo - atkuria tik GPS L1 C/A signalus, naudoja SDR įrangą, kurios kaina prilygsta komercinių sprendimų kainai ir dažniausiai šie sprendimai nėra mobilūs. Kitaip sakant, šiuo metu nėra SDR paremtos GNSS L1+L5 visų signalų testavimo įrangos, kuri būtų tinkama IoT gamintojams, bei būtų nebrangi ir mobili.

Kadangi GNSS testavimo poreikis išaugo, o prieinamos, SDR paremtos testavimo įrangos nėra, šio darbo tikslas sukurti simuliatorių, kuris galėtų įrašyti ir atkurti visus L1 ir L5 GNSS signalus, būtų nebrangus ir tinkamas naudoti mobiliose situacijose.

Darbo metu sukurto įrenginio pagrindą sudaro HackRF One SDR moduliai, kurie tarpusavyje sinchronizuoti, laiko, fazės ir temperatūros prasme. Simuliatorius buvo išbandytas su GPS, GALILEO, BEIDOU ir GLONASS palydovais, bei Airoha AG3335, Quectel LG69T AA ir u-blox NEO-M9N GNSS imtuvais.

Bandymai parodė, kad diskrečių SDR ir sinchronizacijos įrangos terminis stabilizavimas įrašymo ir atkūrimo metu yra būtinas norint neprarasti ir palaikyti stabilų ryšį tarp L1 ir L5 signalų matomų palydovų skaičių ir nešlio ir triukšmo santykių. Taip pat, duomenų praradimo ilgalaikio įrašymo metu galima išvengti tik tada, kai kompiuteris sukonfigūruotas realaus laiko operacijoms.

Įrašų analizė, naudojant apskritimine statistiką, leido statistiškai apibūdinti GNSS signalus. Įrašyti GNSS signalai turi daug panašumų su kompleksiniu baltuoju triukšmu, bei turi fazės pasiskirstymą, kurį galima gerai modeliuoti naudojant von Mises skirstinį. Fazės pasiskirstymas gali būti iškraipytas įvesties signalo amplitudinio apribojimo, tačiau tam tikras apribojimas lygis sumažina fazės pasiskirstymo iškraipymus, kuriuos sukelia kvadratūrinė SDR diskretizacija.

Kadangi bendro pobūdžio SDR moduliais pagrįstas simuliatorius labai apkrauna kompiuterį į kurį įrašomi signalai, buvo ištirti būdai, kaip padidinti sistemos stabilumą. Viena iš tyrinėtų optimizacijų buvo GNSS signalų įrašymas naudojant mažesnį nei būtina diskretizavimo dažnį. Rezul-

tatai rodo, kad bet kuris GNSS L1 signalas gali būti sėkmingai įrašytas naudojant 4 MHz diskretizavimo dažnį. L5 atveju, bet kuris GNSS L5 signalas gali būti įrašytas naudojant 6 MHz diskretizavimo dažnį.

Nepriklausomai nuo mažiausio diskretizacijos dažnio, būtina, kad visi įrašyti signalai naudotų tą patį dažnį, kai L1 ir L5 įrašomi vienu metu. Jei ši sąlyga netenkinama, GNSS imtuvai neatpažįsta L5 signalų, tik L1. Keli skirtingi diskretizacijos dažniai gali būti naudojami tik tada, kai signalai įrašomi toje pačioje GNSS juostoje.

Dinaminio testavimo metu gauti rezultatai rodo, kad visos signalų kokybinės metrikos kaip, kad trajektorija, greitis, matomų palydovų skaičius ar nešlio ir triukšmo santykis glaudžiai sutampa, kai lyginami signalai tiesiai iš palydovų su signalais iš SDR. Tai reiškia, kad iš GNSS imtuvo perspektyvos signalo kilmė yra neatpažįstama ir siūlomas GNSS L1+L5 simuliatorius gali būti naudojamas įrangos testavimo tikslais.

Apibendrinant, šio darbo rezultatai rodo, kad baigiamojo darbo tikslas yra pasiektas ir leidžia padaryti šias išvadas:

1. Norint atlikti ilgalaikes (daugiau nei 20 min) įrašymo ir atkūrimo operacijas yra būtinas kompiuteris sukonfigūruotas realaus laiko režimui, nes kitaip GNSS įraše įvyksta trumpalaikiai duomenų praradimai ir jis tampa netinkamas testavimo tikslams.

2. SDR sukeltus signalo fazės pasiskirstymo iškraipymus galima kompensuoti parenkant atitinkamą įvesties signalo amplitudės apribojimo lygį, nes signalo apribojimo sukeltas fazės statistikos iškraipymas sumažina SDR sukeltus iškraipymus.

3. GNSS signalus galima įrašyti naudojant žymiai mažesnį dažnių juostos plotį nei būtina, nes GNSS L5 signalus galima įrašyti naudojant daugiau nei 3 kartus, o GNSS L1 naudojant beveik 4 kartus mažesnį dažnių juostos plotį, o įrašai vis tiek turi pakankamai informacijos, kad imtuvas nustatytų poziciją.

4. Norint įrašyti kelis GNSS signalus toje pačioje GNSS juostoje, galima naudoti skirtingus diskretizavimo dažnius skirtingiems centriniams dažniams, nes sinchronizacija nėra būtina. Tačiau norint įrašyti signalus esančius skirtingose GNSS juostose, visiems centriniams dažniams turi būti naudojamas tas pats diskretizavimo dažnis, nes sinchronizacija yra būtina.

5. Diskrečių SDR ir sinchronizacijos įrangos terminis stabilizavimas leidžia sinchronizuoti su temperatūra ir triukšmais susijusius procesus visos sistemos posistemėse, nes įrašant ir atkuriant į triukšmą panašius signalus toje pačioje pastovioje temperatūroje, priklausomybės tarp sinchroninių signalų parametrų nėra prarandamos.