GNSS Spoofing Simulator

Tobias Reichel* , Mathias Gerstner,† Andreas Attenberger* , Klara Dološ*
* Central Office for Information Technology in the Security Sector
Munich, Germany
e-mail: {tobias.reichel, andreas.attenberger,klara.dolos}@zitis.bund.de
†Dept. Informatics and Mathematics, OTH Regensburg
Regensburg, Germany
e-mail: mathias.gerstner@oth-regensburg.de

***Abstract*—Modern systems rely heavily on satellite navigation systems for precise positioning, making resilience against spoofing attacks essential. Because satellite navigation signals are openly broadcast and unencrypted, robust anti-spoofing mechanisms must be developed and thoroughly tested. However, existing evaluation methods typically require costly hardware, limiting accessibility for research purposes. This work introduces a fully software-based satellite navigation spoofing simulator that enables realistic attack emulation and anti-spoofing validation without specialized equipment. It is capable of simulating two different satellite systems on one band at the same time, as well as generating two distinct signals, one for simulating the satellites and one for simulating the satellite navigation Spoofer. Validation results show that software-only spoofing provides an effective, low-cost method for advancing satellite navigation systems security research.**

Keywords-gnss; spoofing; simulator.

I. Introduction

Modern embedded systems rely heavily on navigation and location services, including Global Navigation Satellite System (GNSS). However, malicious attacks targeting GNSS data can be executed and remain largely undetectable at present, particularly in post-event analyses such as forensic investigations [1]. To mitigate data loss in such cases, several alternatives for increasing position data reliability have been proposed, including positioning via cellular towers [2] or Starlink [3], but each approach introduces significant drawbacks. Especially in remote environments, reliable and tamper-resistant GNSS functionality remains essential for accurate position determination.

Spoofing can be formally described in different ways. In this paper, the threat model is about data spoofing, where the aim is to replay or generate signals similar to those transmitted by GNSS satellites to miscalculate the position as given in [4]. Such attacks commonly employ a Software Defined Radio (SDR), a computer-controlled radio device capable of emitting synthesized signals. By transmitting a stronger signal than the authentic satellite broadcast, an attacker can force a receiver to lock onto the spoofed signal. However, this approach is inherently flawed: receivers can detect inconsistencies if the signal exhibits unrealistic physical properties, such as Doppler shift or differences in timing and amplitude. Furthermore, direction-finding antennas can determine the true source of a signal. These characteristics can be leveraged to design effective anti-spoofing mechanisms.

In Germany, and in most parts of the world, GNSS spoofing is prohibited, even for research purposes.[5] To avoid interference with critical infrastructure, researchers typically transmit signals over cables or use shielding tents to block high-frequency emissions. However, cable-based testing fails to reproduce the spatial characteristics of real-world spoofing, and shielding tents are often expensive and too small for realistic navigation experiments. The next section outlines existing testing approaches. Section III introduces a simulator that enables GNSS spoofing experiments without specialized hardware and discusses how hardware components could be integrated into the simulation. Section IV presents the validation of the proposed setup.

II. Related Work

The idea of GNSS signal simulation is not new, and numerous commercial tools are available. In, the first open-source project for Global Positioning System (GPS) signal generation was released [6]. Later, in, it was extended to support not only pre-generated and replayable data but also real-time operation [7]. Similar developments exist for other constellations, including Galileo [8], Beidou [9], and Global Navigation Satellite System (GLONASS) [10]. Based on our evaluations, only the open-source generators for GPS and Galileo currently provide sufficiently accurate signal quality. However, the proposed setup can be adapted to other generators once their signal fidelity improves.

For selecting a GNSS receiver that does not rely on dedicated hardware—i.e., one capable of processing recorded I/O files or streamed I/O data—two suitable options are SoftGNSS [11] and gnss_sdr [12]. Both tools can produce human-readable output in various formats and with different levels of detail.

There are multiple ways to study GNSS spoofing using these tools. Typically, they are deployed in hardware-based environments, as demonstrated in [13][14][15]. Even when gnss_sdr is used, it is usually combined with hardware components such as a GNSS receiver or an SDR. Spoofing is then simulated by adding an additional SDR and running multiple instances of the signal generators. In contrast, the setup described below requires no hardware at all; multiple generators can be synchronized and executed simultaneously in a fully software-based environment.

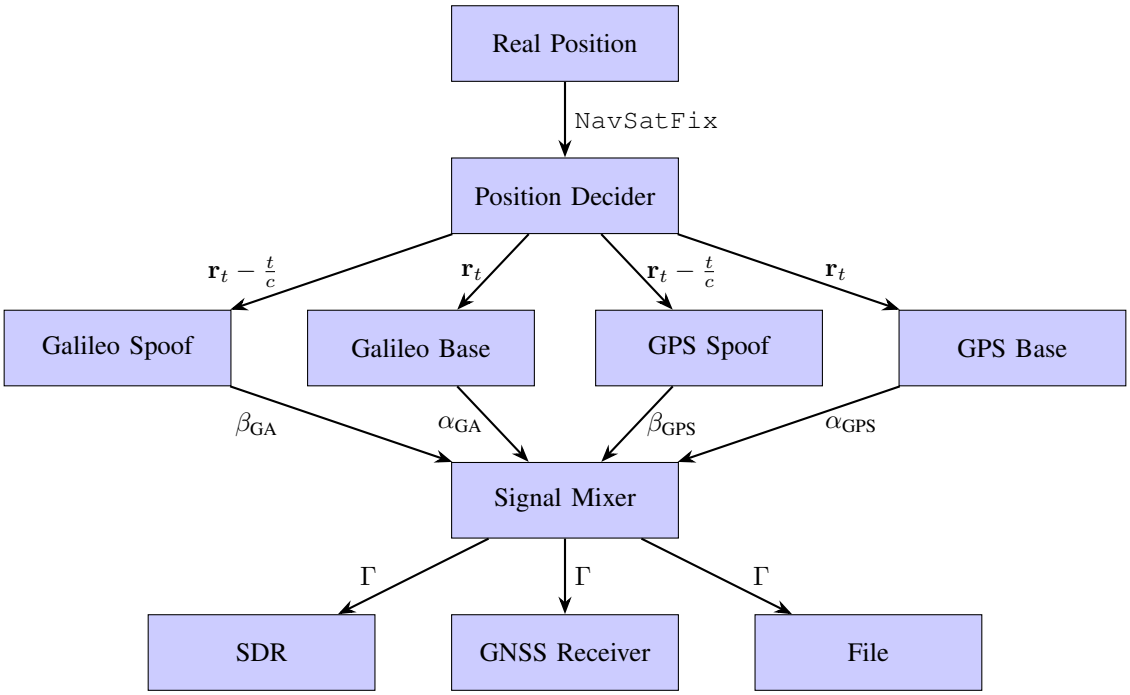


Figure 1. Systematic overview of the GNSS spoofing simulation setup.

III. Simulation Setup

To simulate a GNSS spoofing attack, a real acquisition of the receiver's position must be emulated first. For this purpose, a combination of GPS and Galileo signals generated by the tools in [7][8] can be produced using GNU Radio, a signal processing software. In the same manner, a spoofed version of the GNSS data can be synthesized. By combining the authentic acquisition with the spoofed data and forwarding it to a GNSS receiver, one can obtain a position fix or evaluate the receiver's anti-spoofing mechanisms.

The first requirement, illustrated in Figure 1, is a starting position representing the true receiver location, $\mathbf{r}_t = (x, y, z)_t$, which corresponds to the Cartesian coordinates (x, y, z) at time t. To integrate the simulation with real-world scenarios or external simulators, this position is transmitted dynamically via ROS2 using a `NavSatFix` message to the Position Decider, as shown in Figure 1. A brief description of the message fields is provided in Table I. The essential parameters are latitude, longitude, and altitude, as they are updated dynamically. The ROS2 message is then reformatted into a 24-byte vector, where each of the three parameters is encoded as an 8-byte value.

For spoofing, the transmitted position is not the true location but a slowly drifting value, $\mathbf{r}_t - \frac{t}{c}$, where c is a constant, such that $\frac{t}{c} \approx 10^{-4}$. Because the simulation does not include a physical receiver capable of orienting itself in the Galileo or GPS reference frames, ground-truth satellite signal data must be generated. These baseline signals are denoted as α_{GPS} and α_{GA}. In contrast, the spoofed GPS and Galileo signals, affected by the slow drift, are denoted as β_{GPS} and β_{GA}, representing

TABLE I. Description of a ROS NavSatFix message structure.

Field	Type	Description
header.stamp	time	Timestamp of the message
header.frame_id	string	Reference frame for the data
status.status	int8	Status of the satellite fix (e.g., STATUS_FIX, STATUS_NO_FIX)
status.service	uint16	Type of service (e.g., GPS, GLONASS)
latitude	float64	Latitude in degrees
longitude	float64	Longitude in degrees
altitude	float64	Altitude in meters above the WGS 84 ellipsoid
position_covariance	float64[9]	Row-major 3x3 covariance matrix for position
position_covariance_type	uint8	Type of covariance (e.g., unknown, approximated, diagonal known, known)

TABLE II. COMPARISON OF MESSAGE COUNTS, FIRST TIMESTAMP, MIN-MAX LATITUDE/LONGITUDE DIFFERENCES, AND FIX METADATA (0 –INVALID, 1 – GPS FIX, 2 – DGPS FIX 3 – 3D FIX, 4 – REALT-TIME KINEMATIC (RTK) FIX, 5 – RTK FLOAT, 6 – DEAD RECKONING, 7 – MANUAL INPUT.) ACROSS GNSS DATA SOURCES (GOOD AND SPOOFED SCENARIOS).)

Data Source	Num of mgs	Timestamp	Lat diff	Lon diff	Fix
Original ublox log	552	14:16:05	0.00002133	0.00002899	5
gnss_sdr α_{GPS}	223	00:13:13	0.00007283	0.00004543	6
gnss_sdr α_{GA}	605	00:02:43	0.00001224	0.00002130	5
gnss_sdr $\alpha_{\text{GPS}}, \alpha_{\text{GA}}$	2862	00:02:18	0.00012332	0.00011897	7
ublox α_{GPS}	226	No time fix	0.726741	0.586268	1
ublox α_{GA}	–	No time fix	–	–	0
ublox $\alpha_{\text{GPS}}, \alpha_{\text{GA}}$	231	No time fix	0.000392	0.000271	3
gnss_sdr $\alpha_{\text{GPS}}, \beta_{\text{GPS}}$	11552	00:00:24	0.000144	39.000095	4
gnss_sdr $\alpha_{\text{GA}}, \beta_{\text{GA}}$	4142	00:01:21	0.006285	20.047235	4
gnss_sdr $\alpha_{\text{GPS}}, \alpha_{\text{GA}}, \beta_{\text{GPS}}, \beta_{\text{GA}}$	142	00:01:13	0.777552	34.434319	4
ublox $\alpha_{\text{GPS}}, \beta_{\text{GPS}}$	178	00:00:35	0.000222	0.001033	0
ublox $\alpha_{\text{GA}}, \beta_{\text{GA}}$	–	No time fix	–	–	0
ublox $\alpha_{\text{GPS}}, \alpha_{\text{GA}}, \beta_{\text{GPS}}, \beta_A$	186	No time fix	0.000224	0.000220	0
ublox $\alpha_{\text{GPS}}, \alpha_{\text{GA}}$ and $\beta_{\text{GPS}}, \beta_A$	185	No time fix	32.666489	45.748938	0

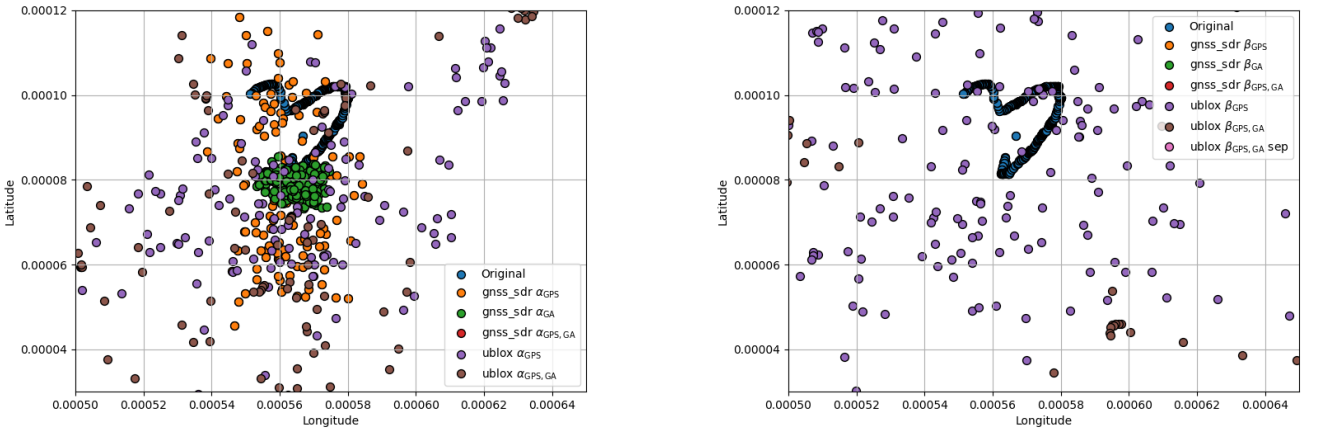
the signals that a typical GNSS spoofer would emit.

The second requirement is valid pre-recorded reference data for the signal generators, which determines both the radius of the data-gathering region and the simulation time. This radius is defined by the set of satellites visible from both the true and spoofed positions, which should be similar. Assigning different gains to the generators $\alpha_{\text{GPS}}, \alpha_{\text{GA}}, \beta_{\text{GPS}}$, and β_{GA} produces different outcomes. Since receivers generally lock onto signals with higher gain—and satellite signals are inherently weak—the spoofed generators β_{GPS} and β_{GA} must be assigned higher gain than the baseline generators. In the validation setup, a gain difference of 30 dB is applied. By combining the signal data $\alpha_{\text{GPS}}, \alpha_{\text{GA}}, \beta_{\text{GPS}}, \beta_{\text{GA}}$ from all four sources as complex number in the following way

$$\Gamma = (\alpha_{\text{GPS}} + \alpha_{\text{GA}}) + (\beta_{\text{GPS}} + \beta_{\text{GA}}),$$

the resulting signal Γ can be written to a file, transmitted via an SDR, or fed directly into a GNSS receiver. Sending Γ to gnss_sdr, a software-based GNSS receiver, is most easily achieved using ZeroMQ (ZMQ). GNU Radio provides multiple sinks capable of transmitting data to common SDRs, sending it over ZMQ, or writing it to a file, all of which are handled in the Signal Mixer as seen in Figure 1.

Ultimately, the receiver determines whether it accepts the spoofed signal, the authentic signal, or fails to acquire a fix. To support this evaluation, various algorithms can be configured in gnss_sdr under the acquisition and tracking modules. For acquisition, we selected Parallel Code Phase Search (PCPS), which employs Fast Fourier Transform techniques to accelerate processing. Advanced anti-spoofing mechanisms, such as those described in [16], may also be integrated. In the tracking module, the configuration specifies how carrier-to-



(a) Location points of the different simulations and the test recording.

(b) Location points of the different spoofed simulations and the test recording.

Figure 2. Comparison of GNSS location points under normal and spoofed conditions.

TABLE III. COMPARISON OF GENERATED MESSAGE TYPES ACROSS THE SIMULATIONS.

Data Source	RMC	GGA	GSA	GSV	Observation File	C/N_0
Original ublox Log	yes	yes	yes	yes	yes	yes
gnss_sdr α_{GPS}	yes	yes	yes	no	no	no
gnss_sdr α_{GA}	yes	yes	yes	no	no	no
gnss_sdr $\alpha_{\text{GPS}}, \alpha_{\text{GA}}$	yes	yes	yes	yes	yes	yes
ublox α_{GPS}	yes	yes	yes	yes	yes	yes
ublox α_{GA}	yes	no	no	no	yes	no
ublox $\alpha_{\text{GPS}}, \alpha_{\text{GA}}$	yes	no	no	no	yes	yes
gnss_sdr $\alpha_{\text{GPS}}, \beta_{\text{GPS}}$	yes	yes	no	yes	no	no
gnss_sdr $\alpha_{\text{GA}}, \beta_{\text{GA}}$	yes	yes	yes	no	no	no
gnss_sdr $\alpha_{\text{GPS}}, \alpha_{\text{GA}}, \beta_{\text{GPS}}, \beta_{\text{GA}}$	yes	yes	yes	yes	yes	yes
ublox $\alpha_{\text{GPS}}, \beta_{\text{GPS}}$	yes	yes	yes	yes	yes	yes
ublox $\alpha_{\text{GA}}, \beta_{\text{GA}}$	yes	no	no	no	yes	no
ublox $\alpha_{\text{GPS}}, \alpha_{\text{GA}}, \beta_{\text{GPS}}, \beta_A$	yes	no	no	no	yes	yes
ublox $\alpha_{\text{GPS}}, \alpha_{\text{GA}}$ and $\beta_{\text{GPS}}, \beta_A$	yes	no	no	yes	yes	yes

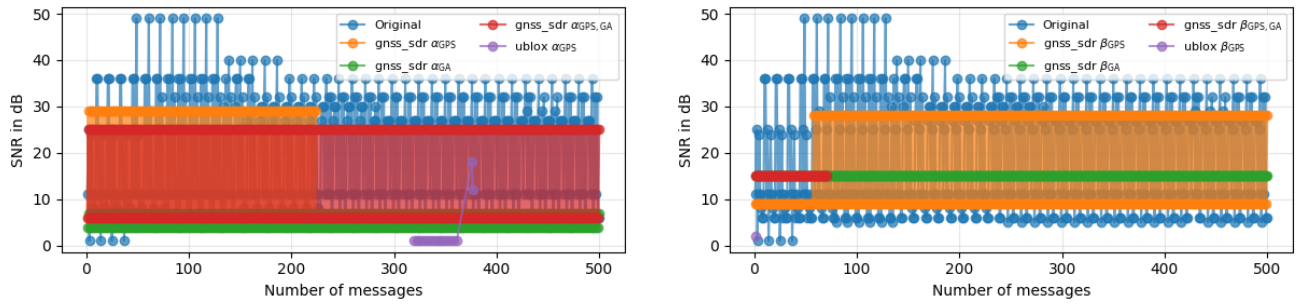
noise density, code and carrier lock, discriminators, and low-pass filters are applied. We selected a Delay Lock Loop (DLL) architecture and a Phase-Locked Loop (PLL) architecture, both of which are already implemented in the framework.

IV. VALIDATION

For the validation, a test recording from the 332nd day of the year 2025 was collected using a u-blox ZED-X20P together with a Survey GNSS Tripleband + L-band antenna from Ardusimple. The GNSS receiver's position was approximately 49°N, 12°E and was logged with six decimal places of accuracy. The receiver recorded raw data (UBX format), which were converted into an observation file, an National Marine Electronics Association (NMEA) file, and a GeoJSON file to enable analysis of the different data types. The observation file contains information about all satellites in view, whereas the NMEA and GeoJSON files are only generated when the receiver obtains a valid fix. The ephemeris files required for the signal generators were obtained from the open-access databases Crustal Dynamics Data Information System (CDDIS)[17] for GPS and International GNSS Service (IGS)[18] for Galileo.

To validate the data generation, a comparison was performed between the test recording, the files generated by our simulation setup, and measurements from the same GNSS receiver receiving signals from two Ettus B200 devices and our simulation inside a shielding tent. Table II lists all measurements and basic statistics. Notably, while gnss_sdr produces significantly more messages, it remains accurate and obtains an appropriate fix. Its timestamps, however, begin at the start time of the ephemeris file. The hardware GNSS receiver was more difficult to deceive. Despite attempts to disable built-in spoofing protection, these mechanisms likely remained partially active, resulting in compromised data. The following analysis focuses first on the position data, then on the signal strength extracted from the NMEA messages, and finally on the Carrier-to-Noise Ratio (C/N_0) values from the observation files.

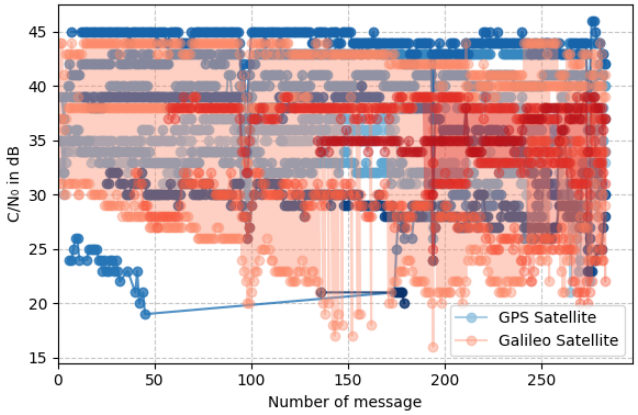
Figure 2a shows the location data extracted from the GeoJSON files for both the base simulations and the test recording. The test recording exhibits much lower variance than the simulated data, although the overall error remains small, as confirmed by the latitudinal and longitudinal differences in Table III. In Figure 2b, which includes all spoofed simulations and the original data, a clear shift to the east and south is visible. This demonstrates that the spoofing attack had an effect, although



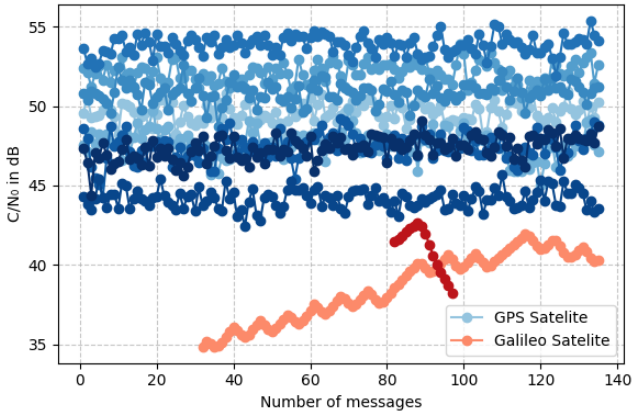
(a) SNR values of the different simulations and the test recording.

(b) SNR values of the spoofed simulations and the test recording.

Figure 3. SNR values of simulation with gnss_sdr and the test recording.



(a) C/N_0 of the test recording.



(b) C/N_0 of GPS and Galileo Signal Input.

Figure 4. Comparison of test recording and simulation with gnss_sd.r

the simulated variance is slightly larger than in the real-world data. This is a relevant aspect for anti-spoofing algorithm development.

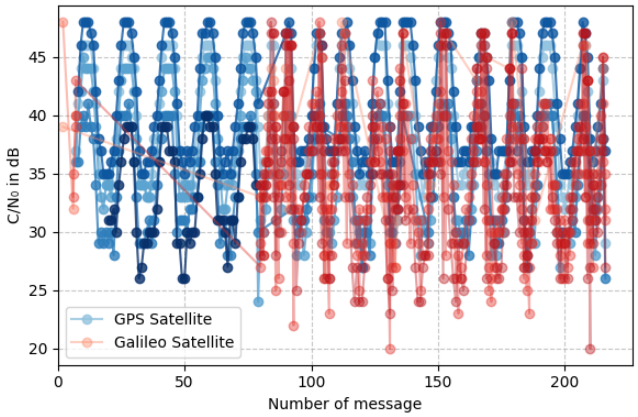
In the next step, only a subset of the NMEA data was analysed, since gnss_sdr produces only Recommended Minimum Navigation Data (RMC), Global Positioning System Fix Data (GGA), GNSS DOP and Active Satellites (GSA), and GNSS Satellites in View (GSV) messages. Table III lists which simulations produced which message types. The first timestamps appear in the RMC and GGA messages. Table II shows that the hardware GNSS receiver did not accept the spoofed time, whereas gnss_sdr did. The Signal to Noice Ratio (SNR) values, contained in the GSV messages, were then examined. Figure 3a overlays all SNR values per message and shows that the mean and variance match well across setups. For the spoofed simulations, Figure 3b shows similar behaviour, although the Galileo-only and combined GPS and Galileo spoofing simulations exhibit degraded data quality.

Finally, the observation files were analysed. Not all simulations produced these files, and even when available, the relevant C/N_0 values may be missing. Interestingly, timestamps missing in Table II could be extracted from the observation files, although they were never used by the receiver. Simulations that produced C/N_0 data are listed in Table III. Each figure in this section shows one simulation, with all satellites and their corresponding C/N_0 values.

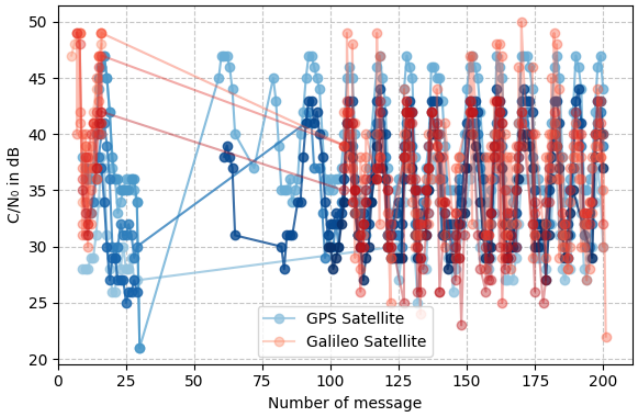
The first notable result is that the simulation can generate data similar to the test recording. The test recording is stable overall, with outliers for satellites barely visible (Figure 4a). gnss_sdr also produces stable data but occasionally exhibits artefacts where a satellite ID is permanently switched (Figure 4b).

A second observation is that the combined GPS and Galileo signal input produces more reliable and abundant data, as reflected in Table III. This behaviour was expected but is nevertheless important to confirm.

The final analysis concerns simulations where spoofed signals were mixed in. Three cases were examined: one using gnss_sdr



(a) C/N_0 of spoofed and base GPS and Galileo Signal Input.



(b) C/N_0 of spoofed and base GPS and Galileo Signal Input, where spoof and base were transmitted with separate SDR.

Figure 5. Comparison between spoofed signal simulation in a ublox receiver, where the signals are mixed via GNU radio or transmitted via different SDRs.

and two using the hardware GNSS receiver with spoofed GPS and Galileo signals. For the SDRs, one simulation transmitted all signals through a single device, while the other separated base and spoofed signals across two different SDRs.

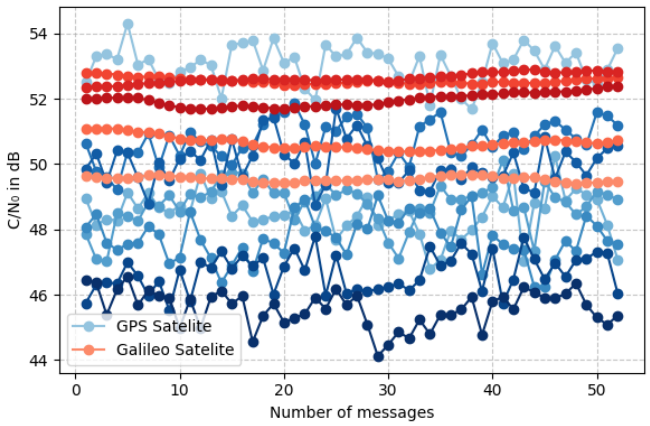


Figure 6. C/N_0 of GPS and Galileo spoofed simulation with gnss_sdr.

The spoofed signal (Figure 6) is significantly more irregular and exhibits stronger fluctuations than the baseline signal (Figure 4b). Comparing it to the real receiver (Figure 5a), the software simulation produced fewer satellites, but similar data points. The number of satellites acknowledged by the receiver drops when the base and spoofed signals are transmitted through separate SDRs (Figure 5b), but the C/N_0 exhibits significantly larger fluctuations likely due to the receiver switching between two independent sources. gnss_sdr handles this situation more gracefully (Figure 6), although the artefacts remain visible. It should be mentioned that the overall values of the C/N_0 in the spoofed simulation with gnss_sdr are much higher than those of the receiver. This is probably caused by interference in the air and the distance between the SDR and the receiver.

V. Discussion

A further issue is that some data could not be produced. In the case of the hardware GNSS receiver, this was likely due to built-in anti-spoofing mechanisms, even though raw values could still be extracted from the observation file. Conversely, gnss_sdr only generates files when it has a valid fixed position. This becomes problematic when analysing spoofing behaviour, because no data are available during the initial phase of the attack. Future work should address this limitation. Additionally, several data types were not analysed but could provide valuable insights. For example, other NMEA message types or navigation files, rather than observation files, may contain useful information.

Some limitations are also given by the signal generators [7] and [8], which we chose to use, caused by the fact that they only generate data for the L1 or E1 bands. Therefore, interesting cross validations between the bands cannot be done. Finally, the spoofed data were less stable and exhibited more fluctuations than the non-spoofed data. This is likely due to the use of identical ephemeris files, which produce identical satellite identifiers and signal characteristics. It should be investigated how the receiver behaves when slightly different ephemeris files are used.

VI. Conclusion and Future Work

This work demonstrates a method to simulate GNSS spoofing attacks without relying on expensive hardware, providing a useful tool for developing and evaluating mitigation strategies. Three major findings stand out. First, combining multiple GNSS signals significantly strengthens the spoofing effect and can deceive receivers more easily. Second, analysing only the coordinates is insufficient to determine whether a spoofing attack has occurred. Third, executing a real-world spoofing attack is considerably more challenging due to the widespread implementation of anti-spoofing mechanisms.

During the validation of the hardware-free GNSS spoofing simulator, several aspects were identified that warrant further investigation, as discussed in the previous section. Future work should focus on supporting additional satellite systems, improving data acquisition, and enhancing data generation by using varied ephemeris files.

Another promising direction would be to investigate how multi-band receivers behave when only a single band is simulated in detail. This would help determine whether a one-band model is sufficient for meaningful system-level evaluation or whether cross-band interactions require a more comprehensive multi-band simulation approach. It would also be valuable to conduct controlled tests on a certified GNSS test range or laboratory environment where such experiments are legally permitted. Finally, future work could include a brief comparison between the hardware-free simulator and a commercial GNSS spoofing system to better understand the differences in signal characteristics, system behaviour, and practical limitations under safe and authorized test conditions.

Acknowledgments

This work was supported by the project 'Digital Forensics in IT Systems (DiForIT)', funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK).

References

[1] T. Reichel et al., "A forensic analysis of GNSS spoofing attacks on autonomous vehicles", in *Proceedings of the Sixteenth International Conference on Cloud Computing, GRIDs, and Virtualization*, 2025, pp. 32–39.

[2] K. Muthineni, A. Artemenko, J. Vidal, and M. Nájar, "A survey of 5G-based positioning for industry 4.0: State of the art and enhanced techniques", in *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2023, pp. 120–125. DOI: 10.1109/EuCNC/6GSummit58263.2023.10188352.

[3] Y. Song, *Positioning and navigation methods based on starlink signals: A comprehensive review*, 2024, https://www.researchgate.net/publication/388122630, Accessed: Feb. 23, 2026.

[4] M. Bartock et al., "Foundational PNT profile: Applying the cybersecurity framework for the responsible use of positioning, navigation, and timing (PNT) services", National Institute of Standards and Technology, Tech. Rep. NIST IR 8323 Rev. 1, Jan. 2023, Accessed: Feb. 23, 2026.

[5] Federal Republic of Germany, *Telecommunications act (tkg) § 149 – administrative offenses*, 2021, https://www.gesetze-im-internet.de/tkg_2021/__149.html, Accessed: Feb. 23, 2026.

[6] T. Ebinuma, *Gps-sdr-sim*, 2015, https://github.com/osqzss/gps-sdr-sim, Accessed: Feb. 23, 2026.

[7] gym487, *Gps-sdr-sim-realtime*, 2017, https://github.com/gym487/gps-sdr-sim-realtime, Accessed: Feb. 23, 2026.

[8] H. Sathaye, M. Motallebighomi, and A. Ranganathan, "Galileo-SDR-SIM: An open-source tool for generating galileo satellite signals", in *Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2023)*, 2023, pp. 3470–3480.

[9] yangfan852219770, *Beidou-sdr-sim*, 2021, https://github.com/yangfan852219770/beidou-sdr-sim, Accessed: Feb. 23, 2026.

[10] A. A. Maksutov, D. A. Valter, G. V. Borisenko, and K. A. Ovchinnikov, "Real-time simulation of the GLONASS system signals using SDR", in *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 2019, pp. 26–28. DOI: 10.1109/EIConRus.2019.8657287.

[11] P. Berglez et al., "Development of a dual frequency software-based GNSS receiver", in *Proceedings of the ION GNSS 2010*, 2010, pp. 1967–1974.

[12] C. Fernández-Prades, J. I. Arribas, and P. Closas, *GNSS-SDR: An open source tool for researchers and developers*, pp. 780–789, 2011, https://gnss-sdr.org, Accessed: Feb. 23, 2026.

[13] W. Feng, J.-M. Friedt, G. Goavec-Merou, and F. Meyer, "Software-defined radio implemented GPS spoofing and its computationally efficient detection and suppression", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 3, pp. 36–52, 2021. DOI: 10.1109/MAES.2020.3040491.

[14] M. Ali, F. Zahra, and A. A. Raja, "Robust spoofing detection in gnss-sdr systems: A two-stage method for real-time signal integrity", in *2024 3rd International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE)*, 2024, pp. 1–6. DOI: 10.1109/ETECTE63967.2024.10823726.

[15] N. Stenberg, E. Axell, J. Rantakokko, and G. Hendeby, "Results on GNSS spoofing mitigation using multiple receivers", *NAVIGATION: Journal of the Institute of Navigation*, vol. 69, no. 1, Mar. 2022. DOI: 10.33012/navi.510.

[16] C. Fernandez-Prades, *GNSS-SDR documentation: Acquisition blocks*, 2021, https://gnss-sdr.org/docs/sp-blocks/acquisition/, Accessed: Feb. 23, 2026.

[17] C. E. Noll, "The crustal dynamics data information system: A resource to support scientific analysis using space geodesy", *Advances in Space Research*, vol. 45, no. 12, pp. 1421–1440, 2010, ISSN: 0273-1177. DOI: 10.1016/j.asr.2010.01.018.

[18] International GNSS Service (IGS), *BRDC navigation files, day 332, 2025*, https://igs.bkg.bund.de/root_ftp/IGS/BRDC/2025/332/, Accessed: Feb. 23, 2026.