POLITECNICO DI TORINO
Repository ISTITUZIONALE

An Open Architecture for Signal Monitoring and Recording Based on SDR and Docker Containers: A GNSS Use Case

Original
An Open Architecture for Signal Monitoring and Recording Based on SDR and Docker Containers: A GNSS Use Case / Mehr, I.E., Minetto, A., Dovis, F., Pica, E., Cesaroni, C., Romano, V.. - ELETTRONICO. - (2023), pp. 66-71. (IEEE EUROCON 2023 - 20th International Conference on Smart Technologies Turin (Italy) 6-8 July 2023) [10.1109/EUROCON56442.2023.10199078].

Availability:
This version is available at: 11583/2981046 since: 2023-08-11T08:54:17Z

Publisher:
IEEE

Published
DOI:10.1109/EUROCON56442.2023.10199078

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

17 November 2026

An Open Architecture for Signal Monitoring and Recording Based on SDR and Docker Containers: A GNSS Use Case

Iman Ebrahimi Mehr
Politecnico di Torino
Department of Electronics and Telecommunications
Turin, Italy
Iman.ebrahimi@polito.it

Alex Minetto
Politecnico di Torino
Department of Electronics and Telecommunications
Turin, Italy
alex.minetto@polito.it

Fabio Dovis
Politecnico di Torino
Department of Electronics and Telecommunications
Turin, Italy
fabio.dovis@polito.it

Emanuele Pica
Istituto Nazionale di Geofisica e Vulcanologia (INGV)
Rome, Italy
emanuele.pica@ingv.it

Claudio Cesaroni
Istituto Nazionale di Geofisica e Vulcanologia (INGV)
Rome, Italy
claudio.cesaroni@ingv.it

Vincenzo Romano
[1]*SpacEarth Technology*
[2]*Istituto Nazionale di Geofisica e Vulcanologia (INGV)*
Rome, Italy
vincenzo.romano@ingv.it

***Abstract*—Signal monitoring and recording station architectures based on software-defined radio (SDR) have been proposed and implemented since several years. However, the large amount of data to be transferred, stored, and managed when high sampling frequency and high quantization depth are required, poses a limit to the performance, mostly because of the data losses during the data transfer between the front-end and the storage unit. To overcome these limitations, thus allowing a reliable, high-fidelity recording of the signals as required by some applications, a novel architecture named SMART (Signal Monitoring, Analysis and Recording Tool) based on the implementation of Docker containers directly on a Network Attached Storage (NAS) unit is presented. Such paradigms allow for a fully open-source system being more affordable and flexible than previous prototypes. The proposed architecture reduces computational complexity, increases efficiency, and provides a compact, cost-effective system that is easy to move and deploy. As a case study, this architecture is implemented to monitor RFIs on Global Navigation Satellite System (GNSS) L1/E1 and L5/E5 bands. The sample results show the benefits of a stable, long-term capture at a high sampling frequency to characterize the RFIs spectral signature effectively.**

***Index Terms*—Software Defined Radio (SDR), Docker Containers, Radio Frequency Interference (RFI), Digital signal Processing, Global Navigation Satellite System (GNSS), System engineering.**

I. Introduction

Monitoring the quality of wireless signals at the Radio Frequency (RF) level is a relevant aspect of ensuring reliable and efficient wireless communications and correct processing of signals to a large extent. RF signals are subject to interference and unexpected degradation, and monitoring them is critical to ensure they remain reliable and consistent. Quality monitoring can be pursued in various ways, including the use of specialized software and hardware tools. These tools can measure the strength and direction of arrival of a signal, the interference levels, and other factors that may affect the performance of wireless communication or may lead to wrong inferred conclusions from signal observations in remote sensing applications. By measuring these factors, it is possible to determine whether a signal is working properly or whether there are interferences that needs to be addressed, for example. For this reason, several kinds of monitoring stations have been proposed, and one of the main threats addressed is indeed the presence of intentional or unintentional interference over the frequency bandwidths of interest such as communication systems [1], Global Navigation Satellite System (GNSS) [2], [3], or others wireless transmission devoted to different aims [4], [5]. These stations can implement advanced equipment and techniques to continuously monitor RF signals and identify any anomalies that affect their performance. The most interesting modern approach to the implementation of such monitoring stations is based on the use of Software Defined Radio (SDR) techniques, which allow the recording of signal samples that can be processed in real-time, but also post-processed to perform deeper analysis and, in some cases, the replay of the scenario [6] [7]. In particular, in recent decades, GNSS monitoring stations have become increasingly popular in scientific research [8], [9]. Historically, GNSS signal monitoring gathers a set of consolidated techniques that leverage live signal processing through front-ends and dedicated monitoring receivers. Similar architectures can be found in the literature that present multi-frequency software receivers aimed at monitoring ionospheric scintillation [10] [11].

Relying on the SDR approach, Cristodaro et al. [2] present the design of a configurable grabbing station to reproduce altered GNSS signals in the laboratory. The proposed architec-

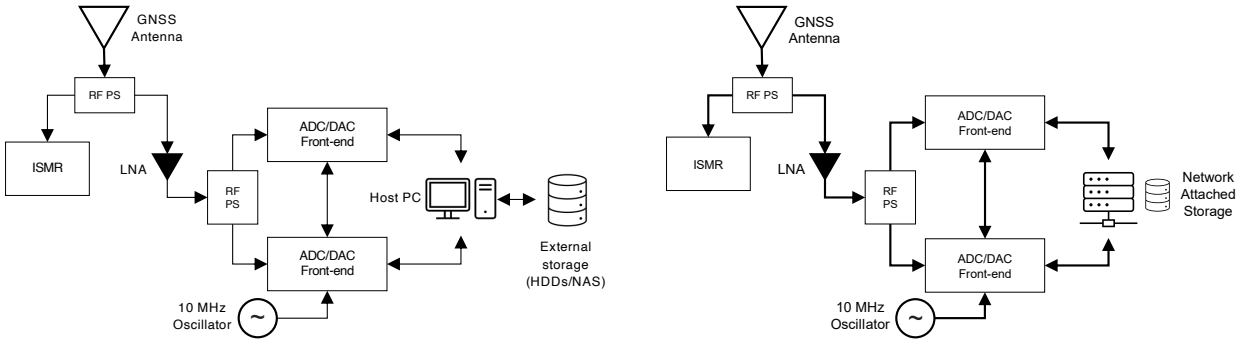


Fig. 1: Evolution of multi-band monitoring and recording stations applied to GNSS signals. State-of-the-art architecture (a) and proposed open architecture (b).

ture, shown in Figure 1a, allowed GNSS signal samples to be stored on external hard drives using Universal Software Radio Peripheral (USRP). In this configuration, the recorded data, in form of binary files (.bin), could be post-processed offline through GNSS fully-software receivers [12]. A MATLAB routine, hosted on the host-PC, was used to post-process the collected samples and decide whether to store or discard the data. Despite its effectiveness [13], this methodology does not easily scale with the number of signals (and bandwidths) that can be recorded, and the need for a dedicated host PC and external storage constitutes a non-negligible limitation. Furthermore, MATLAB-based recording and post-processing require concurring instances that may induce an overload of the host-PC, leading to system instability. Based on such architecture, Pica et al. [14] prototyped a novel system architecture for the triggered grabbing of the GNSS signals by an external GNSS Ionospheric Scintillation and Total Electron Content Monitoring (GISTM) receiver, in case of ionospheric scintillation. At the time of writing, the system is operational in Lampedusa (Italy) and has collaterally detected several unexpected RFI phenomena [14]. Unlike the previous solution, the data storage is managed through a Network Attached Storage (NAS) but still requires a host PC and is mainly based on MATLAB recording and analysis routines. The architecture is still shown in Figure 1a.

In such systems, however, some aspects limiting the overall performance and monitoring potential are present. Autonomous and reliable data collection with high quantization depth and sampling frequencies is threatened by architectural bottlenecks leading to system instability and sample loss. These weaknesses adversely affect the post-processing of the data and may cause a lack of integrity in the recorded datasets. As also reported in [2], the data transfer bottleneck is a non-negligible issue since high sampling frequencies and quantization depth require significant throughput between the front-end and the storage units. In this chain, the host PC of Figure 1a, despite optimization of the operating system and functions, still remains the main point of failure for which high-speed data transfer is concerned.

In order to overcome these limitations, this paper proposes an open architecture named SMART (Signal Monitoring, Analysis, and Recording Tool) and depicted in Figure 1b. This solution is based on SDR and Docker containers directly implemented on the NAS unit, aiming to overcome the limitations mentioned above. The adopted paradigms allow for a fully open-source environment for the monitoring, analyzing, and recording of RF signals. Last but not least, the compact implementation of the data processing in the NAS makes this architecture more suited for in-field implementation and monitoring operations to be performed in remote regions of the Earth lacking or with very limited power supply and connectivity resources. Such a modernized system architecture is suitable for recent advances in the signal analysis since Docker can also host specialized on-site monitoring algorithms without the need to transfer a large amount of typical SDR-grabbed data to remote processing centers. On-site processing allows for the implementation of smart strategies for data storage or limiting the capacity needed for the transfer of the results instead of the complete datasets. As a use case, the significant interest in the usage of Machine Learning (ML) methods to detect and classify interference in GNSS by processing raw signal samples [15] [16] can benefit from the high-fidelity datasets, reliably provided by the proposed system. In this framework, the proposed architecture can provide intermediate-frequency (IF) high-fidelity, In-phase/Quadrature (IQ) signal samples that can be used as input to ML training algorithms hosted in a docker of the system.

II. Architecture Design

In the following, the main components of the proposed solution are presented. Then, the design of the system architecture is described in detail.

A. Main system Components

1) Docker Container: A container is a compact, encapsulated environment created through containerization, which

packages an application, its required dependencies, and the necessary system libraries into a single unit. Docker is an open-source platform that automates the deployment of applications within containers and ensures that the container application operates in any environment [17]. Docker is designed to provide a fast and lightweight environment in which programs can run effectively [18]. In other words, containerization is a type of Operating-System-level virtualization that creates virtual environments for applications through process and network isolation. Container virtualization differs from traditional virtualization, which typically involves creating an entire Virtual Machine (VM) with its own operating system. As shown in Figure 2, unlike VMs, containers share the host's kernel and do not simulate the whole operating system [19].

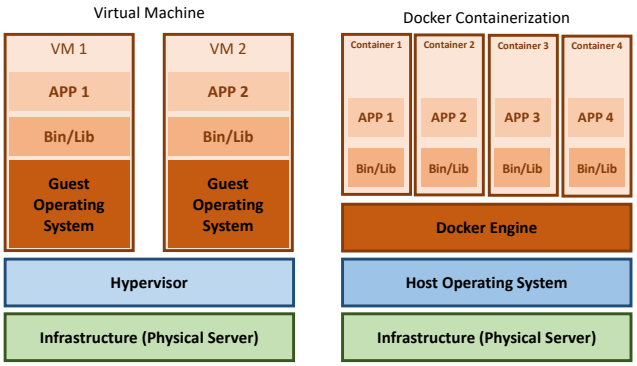


Fig. 2: Docker Container vs. Virtual Machine

The main advantage of Docker is *density*, which implies the container uses the available resources of the host, as opposed to VM, where the machine occupies the dedicated resources, even if it does not use all of them. Docker containers are isolated from each other and the host system, so multiple containers can run on the same host without interfering with each other. Other benefits include speed (time required to create a container), portability, rapid delivery, and an easy way to scale and manage applications [18].

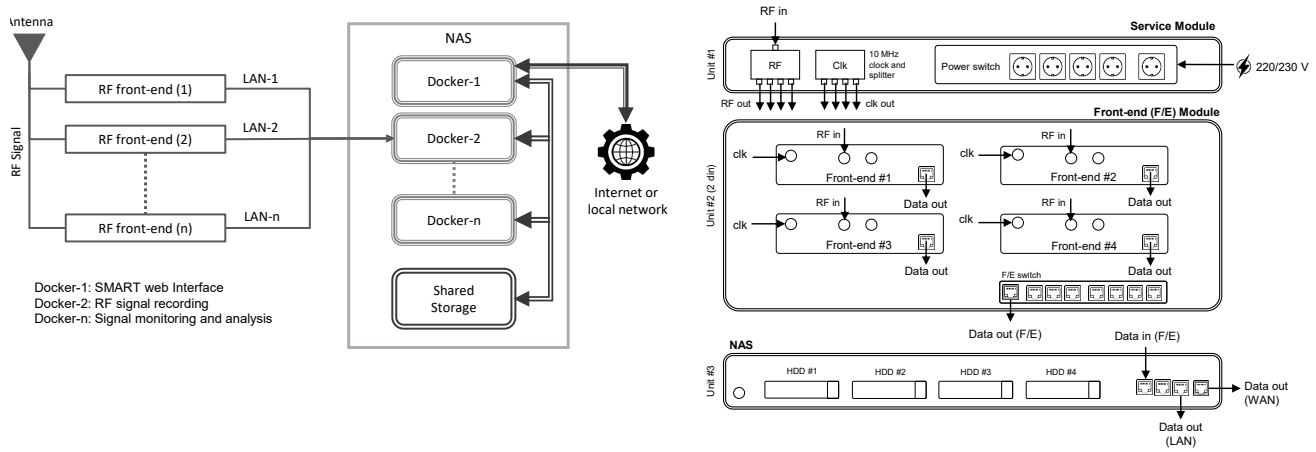
2) Software-Defined Radio: SDR is a wireless communication technology that runs software on a general-purpose processor to emulate hardware components and enable the development of radio communication functions [20]. Consequently, SDRs are more flexible and adaptable to changing the wireless communication architecture by just modifying the software to support new communication services or protocols. SDR technology offers several advantages, such as the rapid evolution of communication standards, performing real-time signal processing, and opening up a wide range of possibilities by simplifying the use of existing radio application types [20]. In GNSS, SDR platforms are used to implement receiver functions in general-purpose software and processors and are divided into real-time receivers, educational/research tools, and snapshot GNSS receivers [21]. GNU radio is one of the most popular open-source frameworks that provides a software toolkit for building and operating SDR systems. It also provides a software development kit (SDK) for developing SDR flowcharts in C++ or Python programming languages [22]. In this study, the SDK of GNU radio is used to develop the system's front-end in Python.

3) Network-Attached Storage: A Network Attached Storage (NAS) device is a standalone data storage system connected to a network and designed for sharing files among multiple users. It acts as a central data repository, and users can access files stored on the NAS from any device connected to the network [23]. The manufacturer typically designs its own Operating System (OS) for NAS units, and it is strongly recommended not to modify its OS or run applications on it. There are a few reasons for this, such as security risks, compatibility issues, and complexity. Instead, they offer tools and services to customize the device, such as a Docker host or virtual machine. Other ranges of features and services include file-sharing protocols (e.g., SMB, NFS, FTP), file synchronization, and backup and recovery options. NAS devices are a convenient and cost-effective alternative to traditional file servers. They offer a flexible and scalable solution for managing and storing data and are designed to provide high reliability, security, and performance for networked data storage needs. The Synology brand is the chosen NAS in this research. Its OS is DiskStation Manager (DSM), which provides a responsive web-based and user-friendly interface for managing and organizing data and various applications and services.

B. System Architecture

Figure 3a depicts a high-level block diagram of the proposed architecture, outlining both the hardware and software components. At the center of the hardware architecture of the proposed system is a NAS device that serves as the core of the system. The NAS is capable of running a Docker container and can host the containers of the required applications and plays the role of host PC. The NAS also stores the collected data, making it the central repository for all raw IF, I/Q signal samples. One of the main advantages of using a NAS as the core of the system is that it eliminates the usual data transfer bottleneck often caused by a host PC and external hard drives. This is because USRP devices can be connected directly to the NAS, bypassing the need for a separate host PC and eliminating the need to transfer data from the USRP device to the host PC and from the host PC to the storage unit.

The software architecture of the proposed system is designed to make the collection and processing of radio signals as efficient as possible. To achieve this, a Docker container has been created to host the GNU radio and UHD libraries. These libraries provide support for general-purpose front-ends such as the Universal Software Radio Peripheral (USRP) and make it easier for users to acquire and process signals from multiple sources. In addition, the scripts used for the grabbing routine and data collection process have been coded in Python. This



(a) (b)

Fig. 3: High-level hardware and software block diagram (a), and SMART modular concept for the on-field deployment of the proposed solutions (b).

programming language makes configuration more manageable, eliminates the need for any licensed software, helps reduce costs, and makes the system more accessible. These scripts run within the same Docker container as the GNU radio and UHD libraries. Furthermore, a multi-purpose web application is developed in the system using Flask and Ninja libraries. The web application

- enables monitoring of system status, including the health and performance of hardware and software components.
- provides a dedicated graphical user interface for monitoring output logs, including detailed records of system activities such as data transfers, processing events, and system updates
- provides the ability to configure local station settings, such as USRP device installation, sampling frequency, bit depth, and scheduling of recording sessions (i.e., duration and time intervals).

A modular hardware solution is depicted in Figure 3b that is conceived for the on-field deployment.

III. Experimental Setup

The proposed architecture has been implemented as a case study in GNSS L-band, i.e., L1/E1 (1575.42 MHz) and L5/E5 (1176 MHZ), for further analysis of the collected data to observe and monitor RFI phenomena. In this experiment, two USRP are utilized as RF front-ends for each band to acquire multi-band GNSS signals simultaneously. Figure 4 shows the hardware installation of the SMART system and its configuration in operation at the NavSAS laboratories. A brief description of each part is described below.

- Antenna : It is a choke ring multi-band antenna that supports multi-constellation GPS, GLONASS, and Galileo. (AeroAntenna AT 1675-382).
- Low Noise Amplifier (LNA): This is a line amplifier 20dB with a DC filter from GPS Networking that amplifies the GNSS signal without significantly degrading the signal-to-noise ratio.

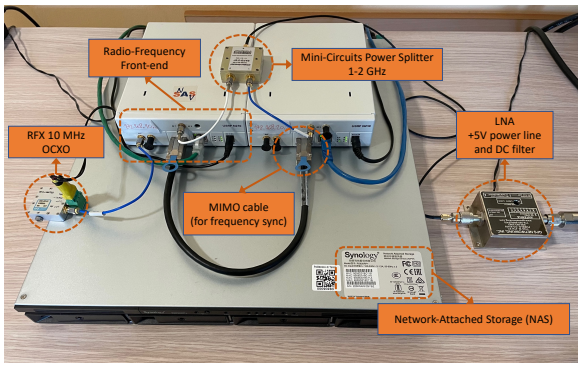


Fig. 4: Prototype of an operational GNSS Monitoring and Grabbing Station at the NavSAS laboratories.

- Mini-circuits power splitter: used to feed the USRP front-end with the RF signals collected from the antenna.
- RFX OCXO: This is an oven-controlled crystal oscillator that provides a more reliable and stable 10 MHz clock source for the USRP devices.
- NAS : The Synology RS820RP+ is a chosen NAS in this experiment, which has a 2.1 GHz Intel Atom CPU with four cores and 6 GB RAM. Its storage is created with RAID 5 (Redundant Array of Inexpensive Disks) and four hard disks of 4 TB, providing a capacity of 10.5 TB.
- ER USRP front-end : The USRP N210 is the RF front-end and performs the ADC conversion of the input signal and the recording of IF signal samples into binary (.bin) files. The USRP devices are connected through Multiple-Input Multiple-Output (MIMO) cable, which is required for frequency synchronization between the devices. Such a shrewdness is particularly suitable to avoid splitting the external clock source, i.e., the OCXO.

IV. Result

Hereafter, a sample application is exploited to show the benefits of the proposed architecture in terms of system performance. Furthermore, the advantages of recording signals at high frequency and direct transfer to the NAS storage unit without loss of samples are highlighted.

A. System resources usage

Figure 5 presents the resource usage (CPU and memory) of the SMART system during the process of grabbing the GNSS signals at different sampling frequencies. The sampling frequency on the x-axis refers to the sum of the sampling frequency for both USRPs (L1/E1 and L5/E5). Figure 5

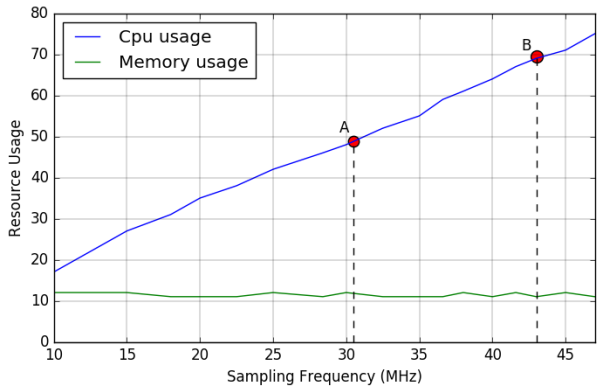


Fig. 5: Resource usage of SMART system

provides insight into how the usage of CPU increases with increasing sampling frequency and how memory utilization does not change during signal grabbing. Points A (30.5 MHz) and B (43 MHz) in Figure 5 refer to the critical sampling frequency of the implemented system in the case of 32-bit and 16-bit quantization, respectively. When the sampling frequency exceeds these points, the system starts to lose sampling. Nonetheless, the result shows long-term reliability in grabbing GNSS signals without sampling loss at a high quantization depth and sampling frequency below the critical points. It is quite evident that the system can achieve a higher sampling frequency if the resources of the NAS device are upgraded. It is interesting to note that in the idle state (no grabbing) with ten active users, the station occupies 4 % of CPU and 9 % of RAM resources.

B. RFI characterization in GNSS L-Band

In this experiment, an interference scenario is designed to show the advantages of grabbing GNSS signals with high sampling frequency. In this scenario, the GNSS signal is interfered with by a linear chirp signal (known as the most disruptive jammer in the GNSS band [24]). Figure 6 shows the implementation of this scenario, where the interfering signal from the multi-band jammer is attenuated and mixed with the GNSS signal.

Figure 7 shows the time-frequency representation (spectrogram) [25] of the GNSS L1/E1 signal, where the jamming

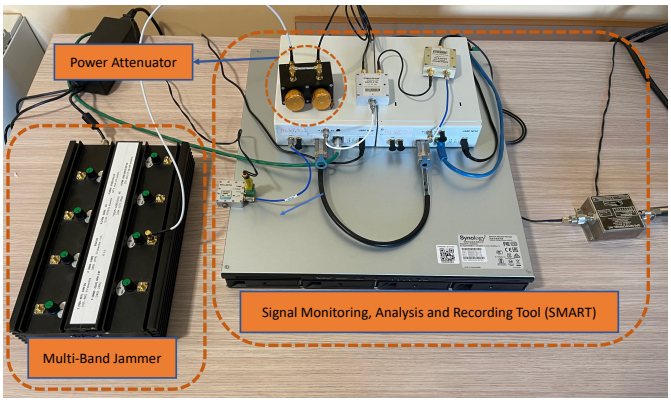
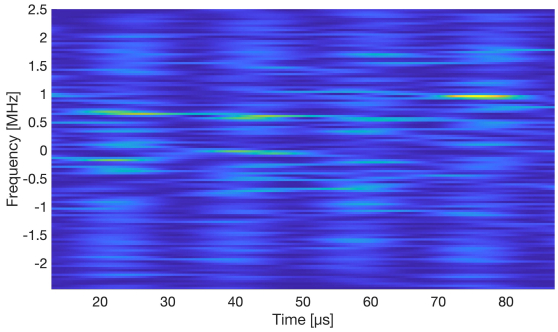
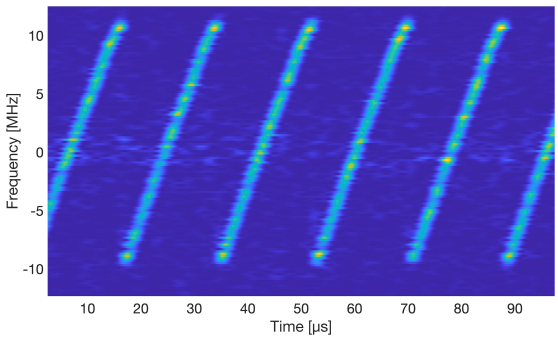


Fig. 6: Implementation of the interference scenario

signal is present at two different sampling frequencies of 5 and 25 MHz. From the results shown here and our preliminary work [15], it appears that a high sampling frequency provides more detail for in-depth analysis by ML or other techniques.



(a) Sampling frequency: 5 MHZ



(b) Sampling frequency: 25 MHZ

Fig. 7: Spectrogram of linearly-chirped jamming signals

V. Conclusion

The paper presents a novel open architecture for signal monitoring, analysis and recording. The proposed solutions overcome the limitations of previous non-commercial solutions and is prone to definition of standard-de-facto for the RF signal monitoring, analysis and recording. Furthermore, the use case presented in the paper about RFI characterization on GNSS L-band justify the need for a reliable and high-fidelity recording of the incoming signals.

ACKNOWLEDGMENT

The authors acknowledge the support from INGV researchers and their contribution to its conceptualization. The Ph.D. work of I. Ebrahimi Mehr is supported by the grant DOT1332092 CUP E11B21006430005 funded within the Italian Programma Operativo Nazionale (PON) Ricerca e Innovazione 2014-2020, Asse IV "Istruzione e ricerca per il recupero" con riferimento all'Azione IV.4 "Dottorati e contratti di ricerca su tematiche dell'innovazione" e all'Azione IV.5 "Dottorati su tematiche green" DM 1061/2021. A. Minetto acknowledges funding from the research contract no. 32-G-13427-5 DM 1062/2021 funded within the PON Ricerca e Innovazione of the Italian Ministry of University and Research.

REFERENCES

[1] S. Rayanchu, A. Patro, and S. Banerjee, "Airshark: Detecting non-WiFi RF devices using commodity WiFi hardware," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 137–154. [Online]. Available: https://doi.org/10.1145/2068816.2068830

[2] C. Cristodaro, F. Dovis, N. Linty, and R. Romero, "Design of a configurable monitoring station for scintillations by means of a GNSS software radio receiver," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 3, pp. 325–329, 2018.

[3] A. Morrison, N. Sokolova, and J. Curran, "Design of a multiband global navigation satellite system radio frequency interference monitoring front-end with synchronized secondary sensors," *Sensors*, vol. 18, no. 8, p. 2594, Aug 2018. [Online]. Available: http://dx.doi.org/10.3390/s18082594

[4] B. Naticchia, M. Vaccarini, and A. Carbonari, "A monitoring system for real-time interference control on large construction sites," *Automation in Construction*, vol. 29, pp. 148–160, 2013.

[5] M. B. Perotoni and K. M. dos Santos, "Sdr-based spectrum analyzer based in open-source gnu radio," *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, vol. 20, pp. 542–555, 2021.

[6] C. Cristodaro, L. Ruotsalainen, and F. Dovis, "Benefits and limitations of the record and replay approach for GNSS receiver performance assessment in harsh scenarios," *Sensors (Switzerland)*, vol. 18, no. 7, 2018.

[7] N. Linty, F. Dovis, and L. Alfonsi, "Software-defined radio technology for GNSS scintillation analysis: bring Antarctica to the lab," *GPS Solutions*, vol. 22, no. 4, 2018.

[8] J. T. Curran, M. Bavaro, A. Morrison, and J. Fortuny, "Developing a multi-frequency for GNSS-based scintillation monitoring receiver," in *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, 2014, pp. 1142–1152.

[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*, 2012, pp. 470–478.

[10] J. T. Curran, M. Bavaro, A. Morrison, and J. Fortuny, "Developing a multi-frequency GNSS-based scintillation monitoring receiver," in *27th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2014*, vol. 2, 09 2014, pp. 1142–1152.

[11] S. Peng and Y. Morton, "A USRP2-based reconfigurable multi-constellation multi-frequency GNSS software receiver front end," *GPS Solutions*, vol. 17, 01 2013.

[12] D. Akos, J. Arribas, M. Z. H. Bhuiyan, P. Closas, F. Dovis, I. Fernandez-Hernandez, C. Fernández-Prades, S. Gunawardena, T. Humphreys, Z. M. Kassas *et al.*, "Gnss software defined radio: History, current developments, and standardization efforts," in *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*, 2022, pp. 3180–3209.

[13] N. Linty, A. Minetto, F. Dovis, and L. Spogli, "Effects of phase scintillation on the GNSS positioning error during the september 2017 storm at svalbard," *Space Weather*, vol. 16, no. 9, pp. 1317–1329, 2018. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018SW001940

[14] E. Pica, A. Minetto, C. Cesaroni, and F. Dovis, "Unclassified Radio Frequency Interferer Affecting Ionospheric Scintillations Monitoring over the Mediterranean Area," 9 2022. [Online]. Available: https://www.techrxiv.org/articles/preprint/Unclassified_Radio_Frequency_Interferer_Affecting_Ionospheric_Scintillations_Monitoring_over_the_Mediterranean_Area/21133078

[15] I. E. Mehr and F. Dovis, "Detection and classification of GNSS jammers using convolutional neural networks," in *2022 International Conference on Localization and GNSS (ICL-GNSS)*, 2022, pp. 01–06.

[16] W. Qin and F. Dovis, "Situational awareness of chirp jamming threats to GNSS based on supervised machine learning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 3, pp. 1707–1720, 2022.

[17] A. M. Potdar, N. D G, S. Kengond, and M. M. Mulla, "Performance evaluation of Docker container and Virtual Machine," *Procedia Computer Science*, vol. 171, pp. 1419–1428, 2020, third International Conference on Computing and Network Communications (CoCoNet'19). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920311315

[18] B. Bashari Rad, H. Bhatti, and M. Ahmadi, "An introduction to Docker and analysis of its performance," *IJCSNS International Journal of Computer Science and Network Security*, vol. 173, p. 8, 03 2017.

[19] J. Cito, G. Schermann, J. E. Wittern, P. Leitner, S. Zumberi, and H. C. Gall, "An empirical analysis of the docker container ecosystem on GitHub," in *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, 2017, pp. 323–333.

[20] T. Ulversoy, "Software defined radio: Challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 531–550, 2010.

[21] D. Akos, J. Arribas, M. Z. H. Bhuiyan, P. Closas, F. Dovis, I. Fernandez-Hernandez, C. Fernández-Prades, S. Gunawardena, T. Humphreys, Z. Kassas, J. Salcedo, M. Nicola, T. Pany, M. Psiaki, A. Rügamer, Y.-J. Song, and J.-H. Won, "GNSS software defined radio: History, current developments, and standardization efforts," in *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*, 10 2022, pp. 3180–3209.

[22] GNU Radio Website, 2023. [Online]. Available: http://www.gnuradio.org

[23] G. Gibson and R. Van Meter, "Network attached storage architecture," *Communications of the ACM*, vol. 43, pp. 37–45, Nov. 2000.

[24] F. Dovis, *GNSS Interference Threats and Countermeasures*, ser. The GNSS technology and application series. Artech House, 2015.

[25] E. Klejmova and J. Pomenkova, "Identification of a time-varying curve in spectrogram," *Radioengineering*, vol. 26, no. 1, pp. 291–298, 2017.