Scientific Journals
of the Maritime University of Szczecin

Zeszyty Naukowe
Akademii Morskiej w Szczecinie

2020, 63 (135), 113–120
ISSN 1733-8670 (Printed)
ISSN 2392-0378 (Online)
DOI: 10.17402/446

Received: *17.02.2020*
Accepted: *30.09.2020*
Published: *30.09.2020*

Initial analysis of the tracking performance of the GOOSE GNSS Software-Defined Receiver

Roman Galas[1]✉, Florian Kunzi[2], Marjolijn Adolfs[1], Matthias Overbeck[2], Wolfgang Felber[2]

[1] Technische Universität Berlin, Institute for Geodesy and Geoinformation Science
e-mail: roman.galas@tu-berlin.de; marjolijn.adolfs@campus.tu-berlin.de
[2] Fraunhofer Institute for Integrated Circuits IIS, Erlangen/Nürnberg, Germany
{florian.kunzi; matthias.overbeck; wolfgang.felber}@iis.fraunhofer.de
✉ corresponding author

Key words: local ionospheric perturbations, scintillations, single autonomous GNSS monitoring station, continuously monitoring GNSS ground systems, autonomous power management, real-time processing, software-defined receiver

Abstract

The GOOSE (GNSS Receiver with open software interface) Software-Defined Receiver has been developed at the Fraunhofer Institute for Integrated Circuits (IIS) in Nürnberg, Germany. The main motivation for the development of this platform was to control the receiver at all stages, from digital signal processing to the PVT domain, and to enable controlled feedback to the hardware. Besides having access to all raw data including correlation values, the GOOSE receiver also enables for example tight- or ultra-tight integration with an inertial navigation system or other dead reckoning systems, as these kinds of architectures require access to the acquisition and tracking loops.

In this paper, the tracking performance of the GOOSE platform was evaluated and compared to a reference receiver (Septentrio PolaRx5S). Several long data sessions were recorded on a "zero baseline" in which both receivers used the same precise geodetic antenna that was also developed at Fraunhofer IIS. The measurements were performed in a harsh environment (obstructions, multipath, possible interferences), as well as on a site with an unobstructed sky view.

Quality and performance analyses were performed using raw measurements (in the domain of primary observables) of three civil GPS signals: L1CA, L2CM, and L5. The data were processed using the "zeroEdit" module of the TUB-NavSolutions academic software for education and research. The quality of the raw observables and tracking performance were described by the following parameters: number of cycle slips detected, number of un-correctable cycle slips, number of loss of locks of the signals, number of single epoch data gaps, and the length of carrier phase arcs. The presentation is illustrated with some numerical examples.

Introduction: GNSS Software-Defined Radio

There are several ways to categorize GNSS receivers using different criteria. For example, geodetic receivers are used for very precise geodetic surveys, or GNSS receivers integrated with onboard field computers and communication terminals for real-time relative positioning or navigation, or any other dedicated configuration. Receiver descriptions sometimes use confusing definitions, such as kinematic or static receivers, for which the survey method does not imply the type of receiver. The typical architecture of a GPS/GNSS receiver is shown in Figure 1.

A *Radio-Frequency Front-End* consisting of a multi-band antenna, down-converter to an intermediate frequency (IF = 9.548 MHz), a chain of filters and amplifiers, and an analog-to-digital (A/D) converter. As an input, there are broadband radio frequency waves from all navigation satellites in view.

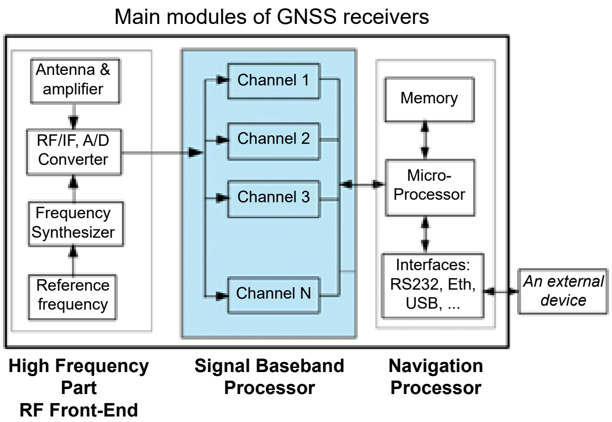

Figure 1. The architecture of GPS/GNSS standard receivers

Samples of those signals (+1/−1, or in higher-resolution, e.g., four bits) in the format signed character (8 bits long) are included in the output.

In standard commercial GNSS receivers, the front-ends are always hardware solutions, the navigation processors are always software implementations, and the baseband processors are/were usually implemented as a hardware solution. The "ideal" software-defined GNSS is based on a hardware front-end and a "pure" software implementation of the baseband processor on a microprocessor (or other platforms like FPGA) using assembler language or a high-level language. However, the performance of highly-flexible pure software solutions is very low compared with low-flexibility hardware-only baseband processors. Due to this, ideal software GNSS receivers have some limitations.

Enabling this control and augmentation (deep coupled integration of the navigation processor and baseband processor (Pany, Kaniuth & Eissfeller, 2005), and tight- and ultra-tight coupled integration with inertial navigation sensors are the main motivations for developing software-based GNSS receivers (GOOSE).

Baseband Signal Processor. Incoming signals from the A/D converter are demodulated, and the 50 Hz "navigation data stream" (message) is decoded, and after successful acquisition, they are continuously tracked. Having a signal in track allows the receiver to demodulate the 50 Hz "navigation data stream" (message) and further process it to track the satellite's PRN code and carrier wave. The tracking procedures are based on lock-loops, such as delay lock-loops (DLL) and phase locked-loops (PLL).

Baseband signal processors are usually implemented as a **hardware solution**, but some, or all processes can be implemented as software solutions in a programmable digital chip like a microprocessor, field-programmable gate array (FPGA), or on a single-board computer. The highest flexibility is available using a "pure" software solution implemented on a PC-like computer; however, a very high performance is provided by hardware-only implementations of signal correlators (Figure 2).

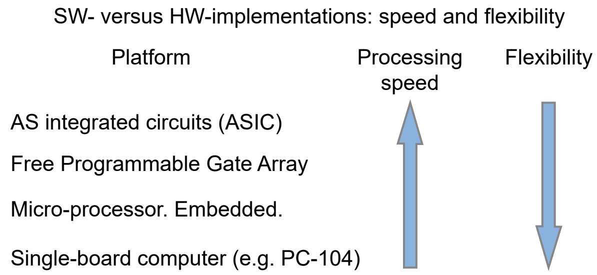

Figure 2. Comparison of the processing speed and flexibility of software- and hardware implementations

There are also some mixed implementations, which are various combinations of hardware modules managed and controlled by software modules. Sometimes various confusing terminology is used in publications to describe or define "Software-Defined Receivers/Radios." Here, the definition widely used in the field of GNSS positioning and navigation is: "a software-defined receiver is a receiver in which all internal digital processing is carried out in a programmable processor by software techniques" (Won, Pany & Hein, 2006).

On the output of the baseband signal processor are primary observables: Doppler shift, code- and carrier-phase observables, and signal-to-noise ratio. These observables appear in the input to the receiver's navigation processor.

Navigation Processor always describes the software implementation. Among the tasks to be fulfilled are, for example, predicting a satellite's visibility. For this, the navigation message and current coordinates of the antenna are required inputs using most actual coordinates of the receiver's antenna and the most recent set of ephemeris collected during signal acquisition and tracking. If the receiver has just been switched-on and no SV's are tracked, the satellite's visibility can be predicted using the most recent station coordinates and almanac, which are usually stored in the receiver's inviolable memory. Another example process is the estimation of an antenna's coordinates, receiver's clock correction, and provision of these pieces of information through user interfaces.

A *user interface* (Figure 1) supports the two-way exchange of information between the receiver and user devices. The interfaces are standard RS232 serial ports, USB ports, or Ethernet port. Users can send requests through these interfaces, e.g., the configuration of receiver tracking parameters, downloading or receiving streams of recorded measurements, status reports, and many others.

User's external devices (Figure 1) are platforms (e.g., embedded computer) with a user's own navigation/positioning software installed, e.g., real-time precise relative positioning. A user's computer that communicates over these interfaces can also be used to integrate communication terminals and additional auxiliary sensors (e.g., loosely integrating an inertial measurement unit (IMU)). To avoid possible misunderstandings, we would like to stress that these "devices" are not "GPS receivers." The interfaces can be used for uncoupled, loosely-coupled, or tightly-coupled integration with IMUs. The term "coupling" indicates if there is data feedback. In the tightly-coupled implementations, internal inputs may also be used within both sensors.

Architecture of the GOOSE GNSS SDR receiver

The GOOSE platform (GNSS Receiver with an Open Software Interface) has been developed at Fraunhofer IIS, Nürnberg, Germany (Overbeck et al., 2015). It consists of a full receiver chain, including a geodetic antenna, front-end, software baseband processor, and a software navigation processor. It is based on a numerically-controlled oscillator (NCO). The overall receiver architecture follows a modular approach, which allows the use of different modules, such as front-end or baseband, within the same receiver setup.

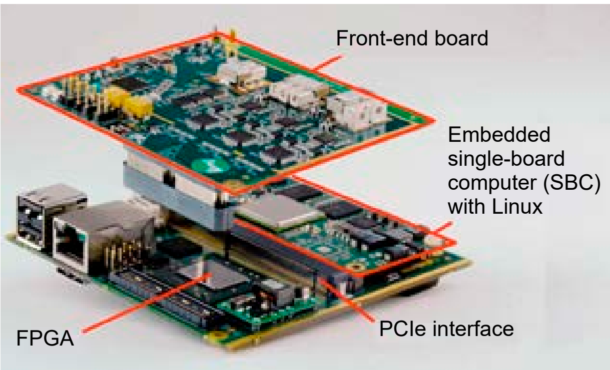

Figure 3. GOOSE Receiver in Single-Board Computer (SBC) configuration

Radio-frequency front-end

The front-end of the GOOSE receiver is designed to sample GNSS signals in the L1, L2, and L5 bands with a bandwidth between 40 and 68 MHz. Therefore, it can support the processing of most signals from the four large constellations including GPS L1/L2/L5, Galileo E1/E5, GLONASS G1/G2, and BeiDou B1/B2, as well as those from Satellite-Based Augmentation System (SBAS) services such as EGNOS.

Two front-end versions are offered: GOOFI and GOOFEX. GOOFI supports three reception channels (bands), while GOOFEX supports four channels. With four bands, the GOOFEX front-end can also receive next-generation GNSS signals. Users can also use their own developed front-end boards instead, but they should have a suitable digital Signal-In-Space (SIS) interface (Garzia et al., 2016).

Baseband signal processor

The baseband processor of the GOOSE receiver consists of a Xilinx 7-series FPGA with a PCIe interface. The processor provides 60 channels in total, split into three groups with 20 channels each, dedicated to L1, L2 and L5 bands. With this configuration, triple-band signal tracking of up to 20 satellites is possible.

The PCIe interface allows the baseband board to be connected to an embedded Linux single-board computer (SBC) with a dual-core ARM Cortex A9, the standard configuration of the GOOSE receiver that has also been used for the evaluation in this paper. However, the board can instead be plugged into a free PCIe slot on the motherboard of a more powerful PC to exploit its processing power, e.g., for development purposes. Furthermore, the baseband provides access to the correlator control interface, code and carrier phase measurements, as well as integrated and dump values, which is a requirement for integrated architectures such as deeply-coupled INS/GNSS.

Navigation processor

The navigation processor of the GOOSE receiver can control the baseband signal processor (Garzia et al., 2016) running on the FPGA (Figure 3). Processing of the data provided by the baseband processor is done using the software running on the SBC or PC. In the Figure 3 the SBC version is displayed. Communication between the different software applications running on the single-board computer is performed with the Open GNSS Receiver Protocol (OGRP™) (GitHub, 2019), a JSON-based vendor-neutral protocol. It is used for real-time communication within receiver software processes and can be adapted and extended for specific purposes and requirements. The software provides an external tracking interface, which has access to all raw measurements and baseband hardware, e.g., so it can steer the numerically-controlled oscillators (NCO) within a closed-loop architecture. By using this interface, custom software can be developed to test complex algorithms or architectures, such as deep coupling in real-time directly in the receiver.

The GOOSE also has USB 2.0 and USB 3.0 hardware interfaces. The USB 3.0 is primarily used to archive digitized data that is available on the output of the front-end. The archived raw data can later be replayed over this interface for off-line signal processing to test or develop new signal processing algorithms. The receiver can also be accessed without knowing the IP address using a USB 2.0 interface. More information about GOOSE receiver can be found in (Ayaz et al., 2015; Overbeck et al., 2015; Garzia et al., 2016).

Description of the testbed

Analysis of tracking performance and data quality of the GOOSE software-based radio receiver was done in a zero-baseline configuration, using a commercial off-the-shelf (COTS) receiver Septentrio PolaRx5S (with an ultra-low noise oven-controlled crystal oscillator, OCXO) as a reference. Both receivers were connected to the same multi-frequency antenna. In this configuration, all biases related to the satellites and atmosphere affected the observations in the same way. Figure 4 displays the hardware/software architecture of the GOOSE receiver with an external computer board used in the investigations.

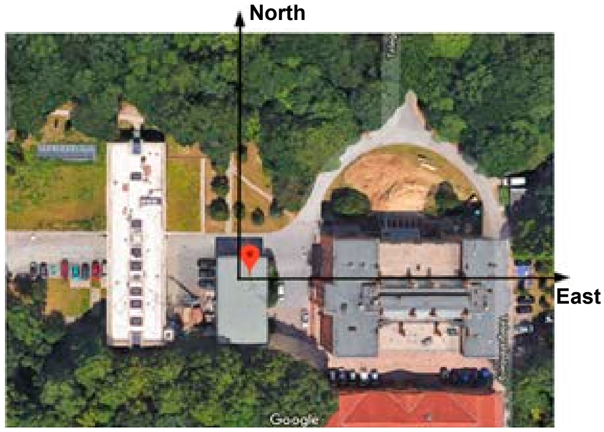

Figure 5. Location of the GNSS antenna on the Telegrafenberg/GFZ in Potsdam

Primary measurements were recorded on the GOOSE receiver's internal platform using the software client OGRP, and downloaded daily onto the external test computer. Septentrio's measurements were logged on an external computer using the serial interfaces.

Field tests were conducted on the Telegrafenberg in Potsdam in a static scenario. Tests were performed in nearly open sky, except for some low-elevation obstructions in the azimuthal directions of 90° and 315° (Figure 5).

During the two test campaigns, which were three-days long each, primary observables from the GOOSE and the Septentrio receivers were continuously recorded with a 1 Hz sampling rate. The Septentrio's configuration parameter "select

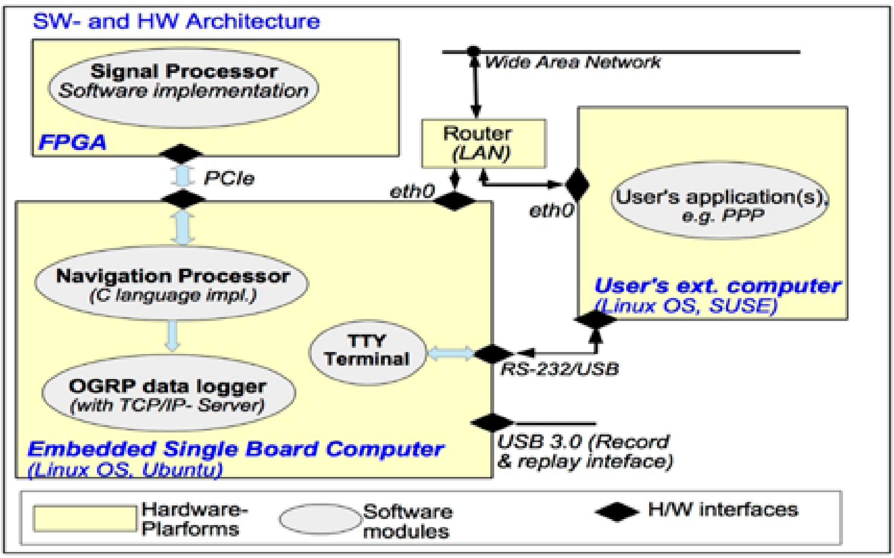

Figure 4. Software and hardware used in the testbed. User applications can also be run on the GOOSE embedded single-board computer

satellite track" was set to track only the GPS constellation, and the GOOSE receiver was also configured to track only GPS. All other configuration parameters of the receivers were left with their default values.

Recorded data were processed with UNAVCO's software tool TEQC (UNAVCO, 2019). The GOOSE data, available in OGRP messages, a custom JSON format, was converted to the RINEX format using a custom version of RTKLIB's "CONVBIN" software tool (CONVBIN 2019).

Presentation and discussion of the results

GOOSE receivers can only track civil navigation signals. The performance of the GOOSE was investigated using observations in Table 1 and transmitted by the GPS satellites Block IIR, IIR-M, and IIF.

Table 1. The GPS Signals processed in the GOOSE receivers

SV Block	L1CA		L2C		L5	
	C1C	L1C	C2S	L2S	C5Q	L5Q
IIR	x	x				
IIR-M	x	x	x	x		
IIF	x	x	x	x	x	x

The signal identifiers in the above table are according to the RINEX version 3.03.

The first evaluation parameter was the availability of:

a) single frequency L1, and

b) dual-frequency L1 and L2

code- and carrier-phase GPS observables. GPS measurements provided by the GOOSE receiver and the receiver PolaRx5S.

The GOOSE receiver currently supports the following signals:

- GPS (L1CA, L2C (M+L), L5 (AltBoc));
- Optional SBAS (EGNOS);
- GLONASS (G1, G2);
- BeiDou (B1I, B2I).

However, at the time the tests were performed, the GOOSE test-receiver could only process the following signals:

C1C, L1C, C2S, L2S

and does not support processing of the "Long length code" C2L (which is processed in Septentrio receivers) and semicodeless proprietary techniques for processing of the encrypted P-code signals.

PolaRx5S receivers do not support the civil "Medium lengt code" C2S. Due to this, availability of the observables provided by the GOOSE test-receiver were compared with availability of the observables provided by the reference receiver Septentrio PolaRx5S processing

C1C, L1C, C2L, L2L

signals.

The default elevation-mask (h = 0°) is different from the default one of the GOOSE receiver (10°), and it cannot be changed in the GOOSE receiver. To make our comparison of the tracking performance between both receivers more consistent, we omitted measurements in the Septentrio's data series (arcs) to the satellite's, in which the elevation angle was lower than 10°. Figure 6 presents the availability of the dual-frequency observations to the selected Block IIF satellites, separately for each (connected) satellite arc.

The length of the full observation arcs recorded by the reference receiver is always longer than those recorded by the GOOSE receiver, even though the elevation mask of the Septentrio's was adjusted to 10°. This is a result of the much better signal acquisition and tracking performance of the Septentrio receiver, as we expected. The signal processing speed is faster in hardware implementations of the baseband processor than in the software (or partial hardware) models (Figure 2). This is the price for the flexibility of software solutions. To reduce differences between those observation series over different timespans, the data availability was also investigated over common timespans. An example

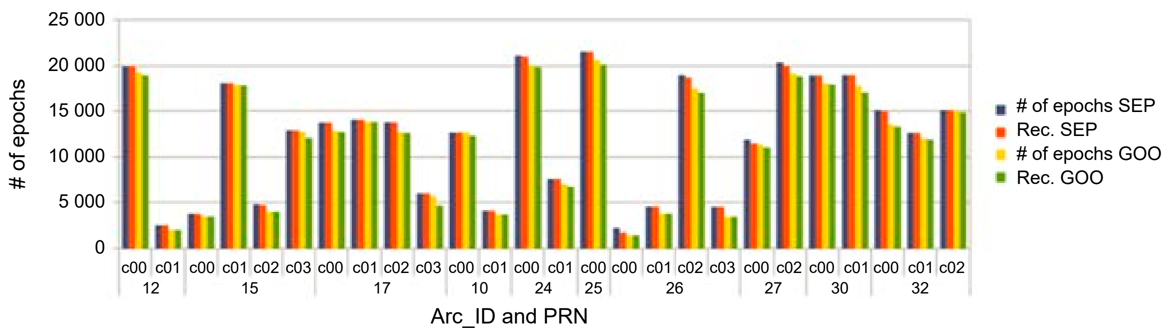

Figure 6. Number of "nominal versus recorded" measurement-epochs for the selected GPS satellites

Table 2a. Data availability comparison: GOOSE versus PolaRx5S. September 2–3, 2019. Dual frequency (L1CA & L2C) epochs. El. Mask = 10°. Two selected arcs, SV PRN26

PRN	Arc	DOY	First obs. epoch	Last obs. epoch	Arc length	First obs. epoch		Max. elev. epoch		Last obs. epoch		Number of meas. epochs			G/S
						El [°]	Az [°]	El [°]	Az [°]	El [°]	Az [°]	Nom.	Rec.	Gaps	
Satellite arcs length as originally registered by the receivers															
26	c00	245	09:02:44	09:27:28	0h24m44s	26.12	184.88	26.12	184.88	15.24	183.16	1 485	1 485	0	G
			09:02:44	09:39:52	0h37m08s	26.12	184.88	26.12	184.88	10.00	182.30	2 229	1 774	455	S
26	c02	246	04:31:37	09:23:52	4h52m15s	16.15	290.60	71.10	245.33	15.02	183.12	17 536	17 066	470	G
			04:15:31	09:31:50	5h16m19s	10.01	287.89	71.10	245.33	11.64	182.58	18 980	18 746	234	S
Satellite arcs length reduced to the common time span															
26	c00	245	09:02:44	09:27:28	$0^h24^m44^s$	26.12	184.88	26.12	184.88	15.24	183.16	1 485	1 485	0	G
													1 485	0	S
26	c02	246	04:31:37	09:23:52	$4^h52^m15^s$	16.15	290.60	71.10	245.33	15.02	183.12	17 536	17 066	470	G
													17 536	0	S

Table 2b. Data availability comparison: GOOSE versus PolaRx5S. September 2–3, 2019. Dual frequency (L1CA & L2C) epochs. El. mask = 10° (SVs which arcs length has been reduced to be the same for both series)

PRN	Arc	DOY	First obs. epoch	Last obs. epoch	Arc length	First obs. epoch		Max. elev. epoch		Last obs. epoch		Number of meas. epochs			G/S
						El [°]	Az [°]	El [°]	Az [°]	El [°]	Az [°]	Nom.	Rec.	Gaps	
15	c00	245	09:04:25	10:02:49	0h58m24s	15.69	56.09	15.98	51.21	11.29	32.66	3 505	3 505	0	G
													3 505	0	S
26	c00	245	09:02:44	09:27:28	0h24m44s	26.12	184.88	26.12	184.88	15.24	183.16	1 485	1 485	0	G
													1 485	0	S
27	c00	245	09:02:44	12:12:13	3h09m29s	69.44	293.76	85.09	226.45	16.76	162.65	11 370	11 048	322	G
													11 370	0	S
32	c00	245	10:32:25	14:18:19	3h45m54s	19.80	131.65	45.40	84.92	10.13	44.73	13 555	13 328	227	G
													13 555	0	S
17	c00	245	12:44:49	16:19:31	3h34m42s	14.40	321.29	39.34	284.57	11.75	237.20	12 883	12 756	127	G
													12 883	0	S
24	c00	245	22:06:25	03:40:01	5h33m35s	13.53	263.43	85.21	218.02	13.93	156.48	20 017	19 869	148	G
		/6											20 017	0	S
26	c02	246	04:31:37	09:23:52	4h52m15s	16.15	290.60	71.10	245.33	15.02	183.12	17 536	17 066	470	G
													17 536	0	S
27	c02	246	06:49:01	12:08:13	5h19m12s	14.16	271.33	85.09	226.42	16.71	162.65	19 153	18 819	334	G
													19 153	0	S
15	c02	246	08:51:25	09:58:44	1h07m19s	15.09	59.93	15.98	51.22	11.30	32.69	4 040	4 040	0	G
													4 040	0	S
17	c02	246	12:39:25	16:11:09	3h31m44s	13.96	321.40	39.35	284.55	13.29	238.30	12 705	12 650	55	G
													12 705	0	S

of the dual-frequency data availability over full and reduced arcs is shown in Table 2a.

In each table row, the upper line describes the GOOSE data, and the lower line Septentrio's data. The gaps in the data sets from the reference receiver only occurred at lower elevation angles, between 10 and 16 degrees, due to obstructions. Similar values were seen in all other observation arcs. The availability of the dual-frequency observations for selected satellites and full observation arcs over common timespans between both receivers are presented in Table 2b.

The recorded measurements of the signal-to-noise observables from the reference- and the GOOSE receivers were also compared. Below (Figure 7), one typical SNR for a selected arc is displayed. We evaluated the obtained results as satisfactory.

The multipath has been plotted in Figure 8. In the GOOSE data sets, there are much larger multipath signatures than in the measurements from the PolaRx5S receiver. Those differences can be explained by the implementation of some multipath mitigation algorithms in the PolaRx5S firmware. The GOOSE receiver does not yet support multipath mitigation.

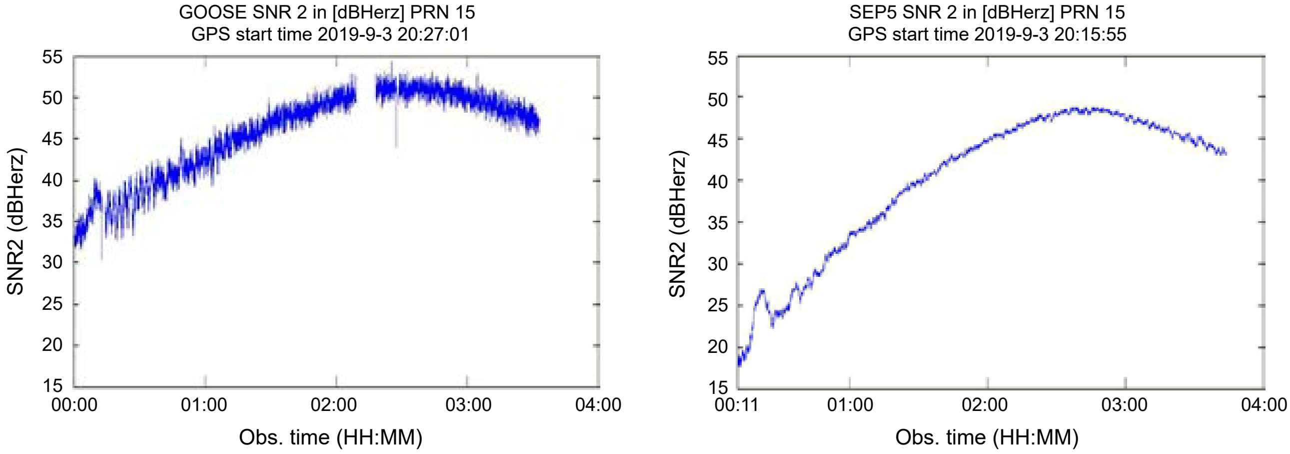

Figure 7. Signal-to-noise plots of the GOOSE receiver (left) and the PolaRx5S receiver (right)

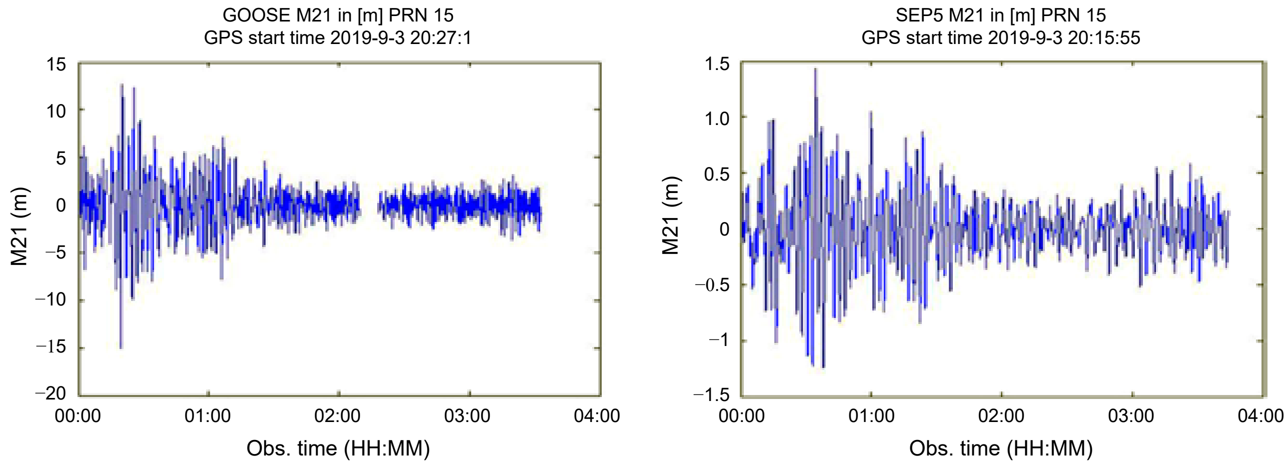

Figure 8. Example multipath plots (M21) of the GOOSE receiver (left) and the PolaRx5S receiver (right)

We also compared the proprietary data formats of both receivers “Septentrio Binary Format” (SBF) and the JSON-based format developed for the GOOSE. The OGRP files are much larger than those from the Septentrio receiver (SBF). All raw GOOSE measurements are displayed as JSON objects. The advantage of the OGRP format is human readability; however, it produces much larger files than SBF files.

Conclusion and next steps

GOOSE is a “pure” software-defined GNSS receiver that can process three constellations: GPS, Galileo, and GLONASS.

The results obtained from the field tests are more satisfying than expected. The GOOSE receiver is a good solution that may be used in several applications, and it fulfills the requirements for the research and development of integrated sensor platforms. The receiver is also a perfect platform for learning and teaching.

We will continue to evaluate GOOSE regarding the quality of its measurements, and TU-Berlin will also develop dedicated software applications for some specific applications.

Acknowledgments

We thank Dr. Tilo Schöne and Mr. Nico Stolarczuk from the GeoForschungsZentrum Potsdam for their help and support with the test measurements on the Telegrafenberg, the site free of radio-interferences.

References

1. Ayaz, A.S., Ko, S.J., Eissfeller, B., Overbeck, M., Kurz, O. & Stahl, M. (2015) *GNSS Receiver Architecture Based on Open Software Interface*. 5th International Conference on Indoor Positioning and Indoor Navigation (IPIN).
2. CONVBIN (2019) [Online] Available from: https://github.com/Fraunhofer-IIS/RTKLIB [Accessed: August 2019].

3. Garzia, F., Strobel, C., Overbeck, M., Kumari, N., Joshi, S., Förster, F. & Felber, W. (2016) *A Multi-Frequency Multi-Constellation GNSS Development Platform with an Open Interface*. *2016 European Navigation Conference (ENC)*, pp. 1–7.
4. GitHub (2019) *OGRP – the open GNSS receiver protocol*. [Online] Available from: https://github.com/Fraunhofer-IIS/ogrp [Accessed: August 2019].
5. Overbeck, M., Garzia, F., Popugaev, A., Kurz, O., Föster, F., Felber, W., Ayaz, A.S., Ko, S.J. & Eissfeller, B. (2015) *GOOSE GNSS Receiver with an Open Software Interface*. Proceedings of the 28th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2015), pp. 3662–3670.
6. Pany, T., Kaniuth, T. & Eissfeller, B. (2005) *Deep Integration of Navigation Solution and Signal Processing*. Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005), pp. 1095–1102.
7. UNAVCO (2019) *TEQC*. [Online] Available from: https://www.unavco.org/software/data-processing/teqc/teqc.html [Accessed: August 2019].
8. Won, J.-H., Pany, T. & Hein, G.W. (2006) GNSS Software Defined Radio. Real Receiver or Just a Tool for Experts? *Inside GNSS* July/August 2006, pp. 48–56. Available from: https://insidegnss.com/wp-content/uploads/2018/01/july-aug06-WP.pdf [Accessed: February 16, 2020].