61st International Astronautical Congress, Prague, CZ. Copyright ©2010 by the International Astronautical Federation. All rights reserved.

IAC-10-B2.3.5

DESIGN AND VALIDATION OF A SOFTWARE RECEIVER FOR GALILEO

Ramón Martínez
ETSI de Telecomunicación, Universidad Politécnica de Madrid, Spain, ramon@gr.ssr.upm.es

Roberto de Lezaeta
ETSI de Telecomunicación, Universidad Politécnica de Madrid, Spain, rdlezaeta@gmail.com

This paper presents a new concept of software-based GNSS receivers for the Binary Offset Carrier signals of the Galileo testbed satellites, Giove-A and Giove-B. The receiver designed in Matlab must be able to acquire, track and demodulate both signals. Its advantages in terms of flexibility and reconfigurability make it an excellent instrument to test new designs or new prototypes as a previous stage on a hardware implementation.

Different techniques have been tried in the tracking and acquisition modules to determine which ones were more efficient and which had the best performance. In the acquisition process 3 techniques were studied: Serial Search, Parallel Frequency Space Search and Parallel Code Phase Search. Finally the Parallel Code Phase Search technique was selected because it performs the acquisition process with the smallest number of iterations with a consequent saving on the processing time required, and also because of his good parameters estimation resolution.

For the tracking module, due to the particularities of the BOC signals of Galileo, the BOC-PRN scheme was selected for his ability to avoid ambiguity on the signal tracking proper of BOC signals.

Also an offline signal generator able to generate the E1 signal from both satellites has been implemented for performance testing of the system. The generator is configurable so that the signal generated meets the characteristics required, such as: IF, Doppler shift and code offset for each channel and SNR.

Different performance test were carried out to ensure that the receiver worked as it should. The first stage of this test was done with locally generated signals and the second and final stage was carried out with signal captures performed with a GNSS RF Front-end during different satellite passes from both Giove-A and Giove-B.

The test results show that these signals can be successfully acquired with a narrowband front-end commonly used for GNSS systems utilizing conventional algorithms and it is proven that the data channel can be tracked with the implemented algorithms.

Finally, a simulator with a graphical user interface named GAEDUNAV (Galileo Educational Navigation Tool) has been implemented to integrate and show in a friendly and simple way the software receiver developed.

I. INTRODUCTION

Global Satellite Navigation Systems known as GNSS are today one of the most active areas for space industry. One of the most important reasons of its boom is the great number of applications that can be developed making use of the navigation information obtained by a receiver. The use of GNSS is not only limited to individual users but also for professional services such as maritime, aeronautical, emergencies, environmental surveillance, farming and transport[1].

Today, the most used GNSS system is the American GPS (Global Positioning System). In 2013, the European Galileo will be fully operative. Space segment of Galileo will be formed by 30 satellites in three planes at 56º of a MEO orbit of 23222 km height[2]. The preliminary steps prior to final deployment have been the launch of two technology demonstration satellites in 2005 and 2008, known as Giove A and Giove B, respectively, and the launch of IOV (In Orbit Validation), four satellites quasi identical to the final satellites of the Galileo constellation. The deployment of IOV is expected to be completed in 2011.

Software receiver technology is based on the processing of digital samples in intermediate frequency so that signal processing tasks (downconversion, filtering, demodulation) can be done in the digital domain. Ideally, a software radio receiver will digitize the RF signal at the output of the antenna at the expense of using very high sampling rates and complex analog-to-digital converters.

The advantages of an implementation based on software can be expressed in terms of flexibility, reconfigurability, size and power consumption. As well, it is an excellent way to test new designs and prototypes as a previous step prior to hardware implementation and manufacturing.

With the development of this project, we aimed at contributing to the dissemination of Galileo. Under this framework, we focused on the development of software receivers as a contribution to design more efficient user terminals. As part of our research, we developed a software tool named GAEDUNAV that shall be used in teaching activities to illustrate how a GNSS receiver works and the particularities of different receiver algorithms.

II. THEORY

A software receiver for GNSS is formed by several hardware and software modules. The hardware parts are the antenna and the RF front-end, whose tasks are the reception of satellite signals, conversion to intermediate frequency, analog-to-digital conversion, signal conditioning and delivery of samples to a computer in IF or baseband.

Signal processing is entirely done in a PC with a software program organized in different routines each with a different task. The input of the software modules are packets of IF samples. Therefore, any front-end that provides IF samples can be easily integrated to the software part using e.g. a USB port. In this work, we focus on the software and make use of a commercial GNSS front-end as it will be explained after.

II.I Block diagram of the software receiver

The structure of the software receiver can be divided in three main blocks: acquisition, tracking and post-processing (Fig. 1).

In order to obtain the position of the receiver it is necessary to extract navigation data from the signals received from the satellites. Therefore, the receiver tasks can be separated into the extraction of navigation bits in the acquisition and tracking modules, and the postprocessing of navigation data in order to calculate the position. The mission of acquisition and tracking modules is to acquire and track the frequency and phase variations of the received signals. These modules are essential to successfully demodulate the navigation bits.

The software is based on the routines provided in[3]. The receiver design has been done in Matlab following two criterias: 1) software modules shall be clearly separated and independent, and 2) software routines shall be reconfigurable.

For the first feature, each routine generates a results file with results that can be saved and used by the following routines anytime without the need of executing the routine again. These results file clearly identify the interfaces between modules.

For the second criteria, a configuration file is available that manages all the parameters of the receiver. Its functions are the initialization and storage of the simulation parameters. Simulation parameters can be modified either in the configuration file or in the GUI (Graphical User Interface) and are available to the user. After program execution, a structure is created and saved with information of simulation parameters.

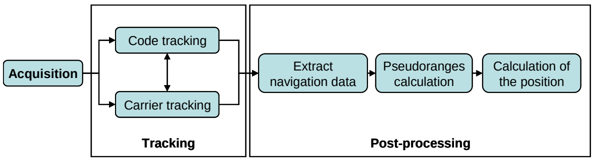


Fig. 1. Structure of the software receiver.

II.II Signal structure

Signals of Galileo are of the Binary Offset Carrier (m,n) (or BOC(m,n)) family. In particular, the modulation of the Open Service (OS) signals is MBOC (Multiplex BOC) and is formed by combination of a BOC(1,1) and a BOC(6,1). Its advantages over conventional BPSK used in GPS lies in their robustness against multipath and narrowband interference. It also allows an adequate RF compatibility in the L1 band (1159 – 1610 MHz)[5].

Signal E1 (OS) is composed of three channels A, B and C. Channel B (eE1-B(t)) and C (eE1-C(t)) carry the navigation data and pilot channel, respectively, whereas channel A (eE1-A(t)) is reserved for governmental use and is encrypted. Primary codes are generated using a LFSR (Linear Feedback Shift Register) structure whose seeds and taps are described in the Giove's ICD[4] (Interface Control Document). The combination of the three channels is done with a CASM (Coherent Adaptative Subcarrier Modulation) in order to maintain the power envelope constant.

II.III Acquisition module

The purpose of the acquisition module in a GNSS receiver is to identify all visible satellites. If a satellite is effectively visible, the acquisition process ends with the determination of its frequency and code offsets (code phase) that is used in the processing and calculation of the receiver position.

In order to search for the signal frequency and code offset, the acquisition algorithm must try with different frequency-code offset pairs. There are three main techniques that differ in complexity and duration: Serial Search, Parallel Frequency Space Search and Parallel Code Phase Search. In this work, we have used Parallel Code Phase Search as it minimizes the number of iterations. This method is more complex and requires the use of FFT and IFFT to perform the cross-correlations in the frequency domain and come back to the time domain representation[3]. The advantage of this method is that all code phases (4092 for E1B and 8184 for E1C) are evaluated for each frequency step which was fixed to 62.5 Hz as a trade-off between resolution and computational complexity. The search interval was fixed to ±14 KHz around the carrier frequency, as a worst case for GNSS systems.

The diagram of the acquisition module is shown in Fig. 2.

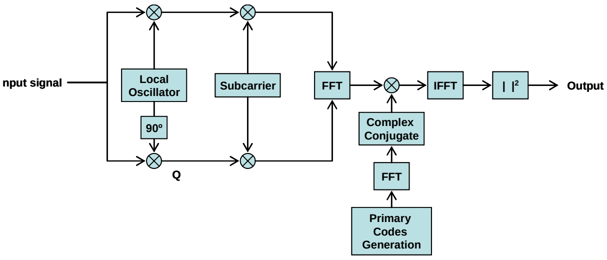


Fig. 2. Scheme of the acquisition module based on Parallel Code Phase Search.

After, in order to refine the frequency estimation ten periods of the signal per code are analyzed using a FFT processing with eight times the number of points in the frequency domain. This procedure is the coarse acquisition and provides an initial rough estimation.

II.IV Tracking module

Tracking is performed after acquisition and has two main functions. First, a fine tracking of the acquired signal in frequency and code domains is performed to avoid loss of the acquired signals. As the satellite is moving, both offsets changes with time due to orbit perturbations and phase noise of the oscillators and in case of signal loss, the acquisition should be carried out again.

Secondly, demodulation of the received signal must be done simultaneously to the fine tracking. Demodulation is done multiplying the signal with a replica of the carrier and then with replicas of the primary code and subcarrier. These replicas are estimated by the tracking algorithm.

The tracking module is formed by a Costas loop for carrier tracking and a DLL (Delay Locked Loop) for code tracking[4]. The use of a Costas loop is motivated by its insensibility against phase changes of 180 degrees that appear once every code period. The discriminator for the E1B signal is the arctan expressed as

$$\phi_{error} = \arctan\left(\frac{Q}{I}\right) \qquad [1]$$

Where I and Q are the in-phase and quadrature components of the signal, respectively.

It is the most adequate discriminator for the Costas loop, but leads to the longest tracking period. The advantage is that its output is the phase error.

For the E1C, we use the extended arctan discriminator (arctan2) given by

$$\phi_{error}^{ext} = \arctan 2\left(\frac{Q}{I}\right) \qquad [2]$$

The reasons for using the extended discriminator for the pilot channel is the largest lineal margin for trackng, so that it entirely covers the [-π,π] range[6]. It thus supports a larger tolerance to phase errors without tracking loss. Moreover, it does not require normalization so that complexity is reduced.

Regarding the DLL, a correlation scheme with three replicas is used: prompt, early and late. Contrary to GPS, BOC(1,1) signal presents two additional problems: bandwidth occupied by the main lobe is doubled and it presents secondary peaks in the autocorrelation function that may lead to ambiguities[7]. If the DLL gets tracked to an erroneous time instant, it may lead to errors in the pseudorange calculation.

To solve these problems, several algorithms were studied and we final opted to use the BOC-PRN (E+L) discriminator[7]. Its architecture is based on the BOC PRN used for mixing the input signal (BOC) with the same replicas of the primary code (PRN). Therefore, the effect of the subcarrier modulation in early and late branches is avoided, while in the prompt branch this effect is eliminated to estimate the Doppler deviation of the carrier.

Phase errors in frequency and code are feedback as in GPS receivers, but with the additional correction of the locally generated subcarrier. The advantages are the elimination of ambiguous points, the steep slope of the discriminator function and the robustness against multipath making the early-late spacing reconfigurable. The expression of this discriminator is

$$D(x) = \frac{\left(I_E + Q_E + I_L + Q_L\right)}{K} \qquad [3]$$

Where {I_E , Q_E} and {I_L , Q_L} are the in-phase and quadrature components of the early and late branches, respectively, and K is an empirical factor that depends on the sampling frequency and the signal power to maintain the phase error in a limited range. The scheme is shown in Fig. 3. Other authors solve the above mentioned problems using a normalized “Early-Late” discriminator with lower spacing between replicas lower (0.2 chips) and a BOC-BOC scheme[8].

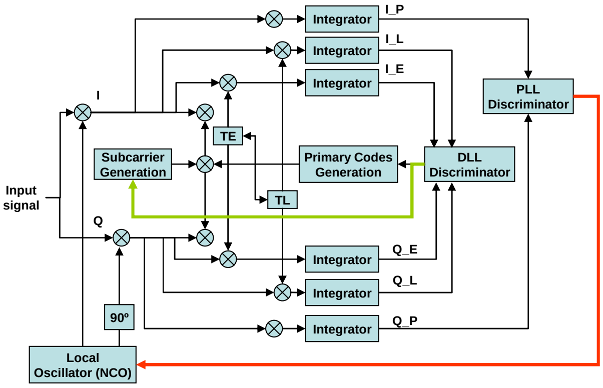


Fig. 3. Scheme of the tracking module.

III. SIMULATION

III.I Signal generator

As a previous stage on the design of the software receiver, a signal generator was implemented to be used as a testing performance tool that could easily check the functioning of the different modules involved. The high value of this tool resides on its own configurability, so in the case that the original parameters of the signals from Giove-A/B are changed, the generator could be adapted in order to meet the new characteristics. Moreover, almost every parameter can be configured to generate a specific signal to test each module according to the issues desired. The generator implemented in Matlab was designed following the last ICD[4].

The tool produces I-Q samples of the E1 signal's complex envelope at an intermediate frequency, along with all its channels (A, B and C). It is possible to generate the two E1 signals now flying (Giove-A and Giove-B) with their own modulations scheme. All samples are created according to a fixed sampling frequency of 61.38 MHz. The reason for this common characteristic is that this value is 4 times the higher subcarrier frequency (15.345 MHz), and therefore it allows performing the tests with different intermediate frequencies. It is important to remind that the E1 signal comprises an approximate total bandwidth of 32.736 MHz.

The intermediate frequency of the samples as well as the Doppler shift, the code offset for each channel and the SNR are variable, so that the characteristics of the generated data are closer to those that could be obtained from a GNSS RF front-end device. The power spectral density (PSD) of the generated signals are shown in Fig. 4 (Giove-A) and Fig. 5 (Giove-B).

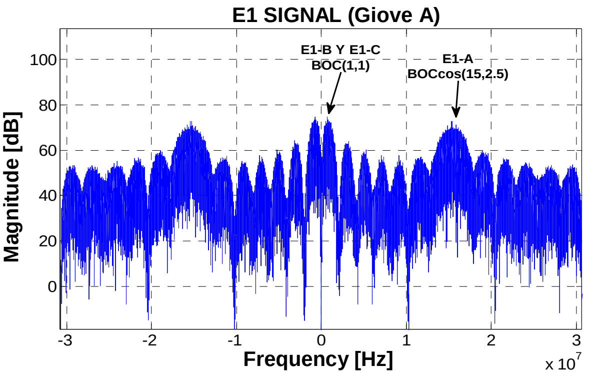


Fig. 4. Spectrum of E1 Signal from Giove A (baseband).

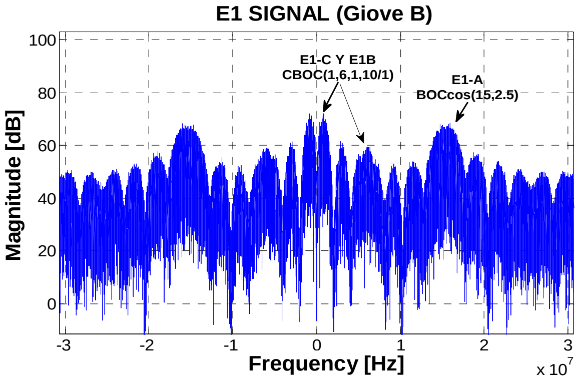


Fig. 5. Spectrum of E1 Signal from Giove B (baseband).

III.II Acquisition results

The performance of the acquisition module was tested with both locally generated signal and signals coming from capture files. In Fig. 6, the correlation matrices on the frequency and code domain for a locally generated signal are shown. The intermediate frequency (IF) is 1 MHz, the Doppler shift introduced is 2000 Hz, the SNR = -30 dB and the code phases are 1.135 Chips for E1B and 4.657 Chips for E1C.

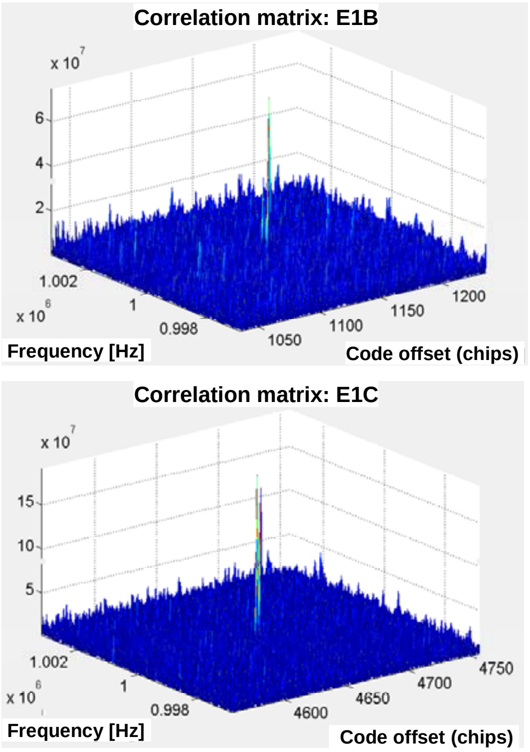


Fig. 6. Correlation matrices of E1B and E1C signals (Giove B): IF = 1 MHz, Doppler shift = 2000 Hz, SNR = -30 dB, E1B Code Offset = 1135 chips and E1C Code Offset = 4657 chips.

The results of the Doppler estimation are 2020 Hz for E1B and 1992 Hz for E1C, while the code offset estimations are 1135 chips for E1B and 4657 chips for E1C. The reason for the Doppler estimations not meeting the value introduced are that the main objective is having a coarse approach to the carrier frequency, a first estimation. The accurate (fine) estimation is carried out within the next module (tracking). On the other hand, the fact that the obtained values are not exactly the same is due to the different number of points on the FFTs used, which is a consequence of the different code lengths.

The tests with measured signals were performed with a captured file obtained using the NordNav R30 on the Politecnico of Torino. The intermediate frequency is 4.1304 MHz, the sampling frequency is 16.3676 MHz and the format of the samples is an 8 bit integer. The results are shown in Fig. 7 and Fig. 8.

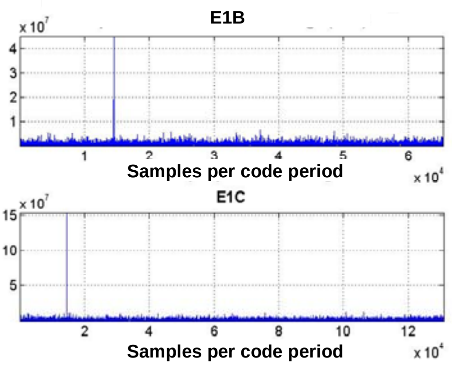


Fig. 7. Correlation results in the code domain for the selected frequency bin.

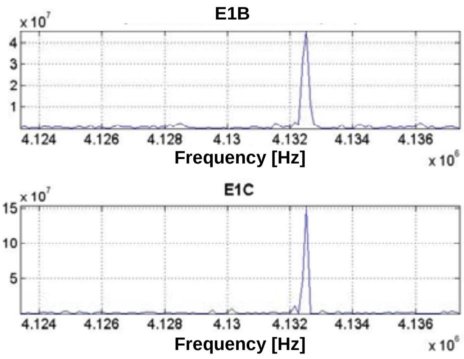


Fig. 8. Correlation results in the frequency domain for the selected code offset.

The Doppler estimation on the E1B channel was 2055 Hz while on the E1C it was 2052 Hz. The code offsets were 14.250 and 14.519, respectively. These results are among the normal values that can be obtained during a satellite pass, but still, it is important to remark that the estimations are very similar. This is due to the fact that the oscillators and the clocks used to generate the signals on board are the same, which means that the signal disturbances will be also similar.

In order to observe the effect of noise in the acquisition process, which sometimes may mask the signal making impossible its detection, the algorithm performance was tested with different values of SNR. The Doppler shift introduced on each test was constant and equal to 2000 Hz and the intermediate frequency was 1 MHz. Fig. 9 shows the difference (in Hz) of the estimated and the real values, and its variation with the SNR (dB) for both satellites. The final results were obtained averaging 100 iterations of the algorithm for each SNR. Each iteration takes around 8 minutes, which means 13 hours per SNR and 10 days per plot. In normal circumstances the acquisition time of a generated signal or a captured one is of 2 minutes in average, and the higher level of noise tolerable is 30 dB over the signal power.

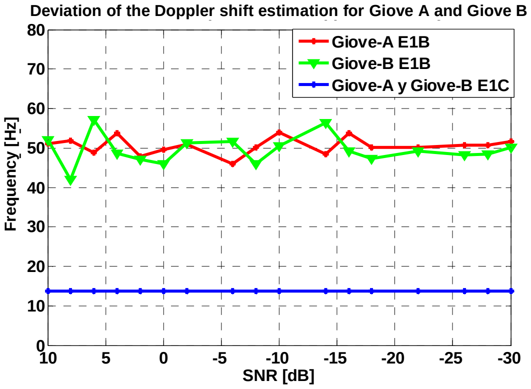


Fig. 9. Deviation of the Doppler shift estimation for different SNR.

III.III Tracking results

The tracking results of the data channel (E1B) were obtained from captured files performed over different satellite passes of both spacecrafts, Giove-A and Giove-B. Fig. 10 shows the navigation bits of Giove-B extracted from a captured file taken at Madrid with an elevation angle of 58.4º on August 25th, 2009.

In Fig. 11 the correlation levels of the three code replicas involved in the process are shown, being the green coloured the prompt, the blue one the early and the red one the late. By taking a close look on these values it is possible to determine if the tracking is correct or not. A higher correlation level on the prompt replica and a similar and lower correlation level on both the late and the early replicas indicate that the prompt replica is actually hitting on the right place and that the tracking system is working correctly.

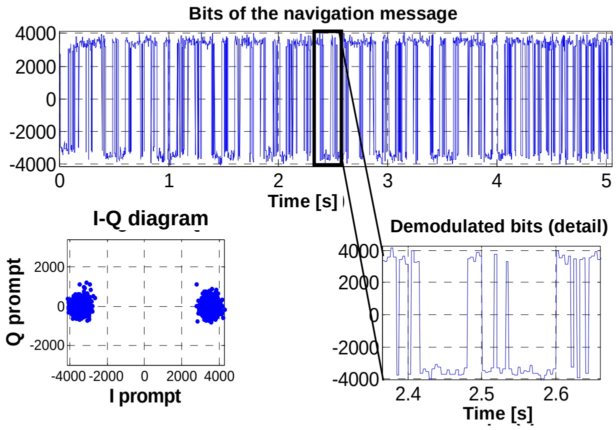


Fig. 10. Bits of the navigation recovered from the capture of Giove B signal on 25/08/09 in Madrid with an elevation of 58.4º.

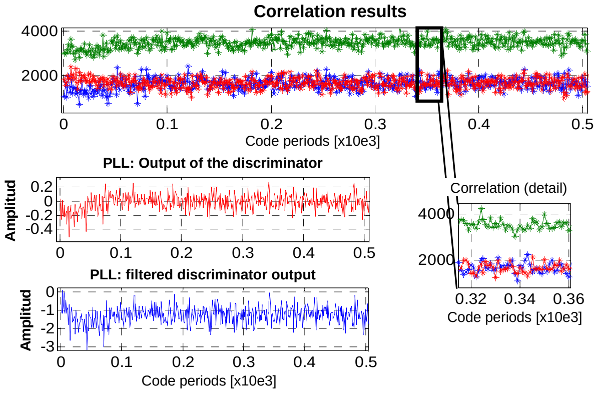


Fig. 11. DLL performance with the capture of Giove B signal on 25/08/09 in Madrid with an elevation of 58.4º.

Similar results for a signal capture from Giove-A performed in Madrid with an elevation angle of 52º the 30th of June of 2009 are shown in Fig. 12 and Fig. 13. The length of the process depends on the number of second that are to be processed. As a normal example, in order to process 5 seconds of signal it is necessary to wait 111 seconds. The parameters used in the tracking process are detailed on Table I.

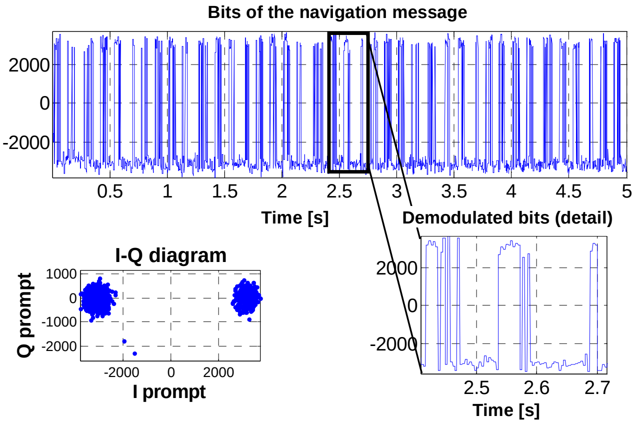


Fig. 12. Bits of the navigation recovered from the capture of Giove A signal on 30/06/09 in Madrid with an elevation of 52º.

Parameter	Value
DLL discriminator	BOC-PRN (E-L)
PLL discriminator	arctan
DLL/PLL order	2/2
Early-Late spacing	0.6 chips
Integration time	4 msec
Scaling factor for the DLL	550

Table I. Simulation parameters.

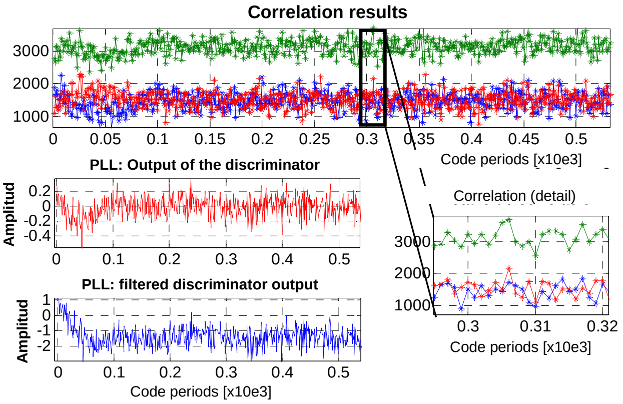


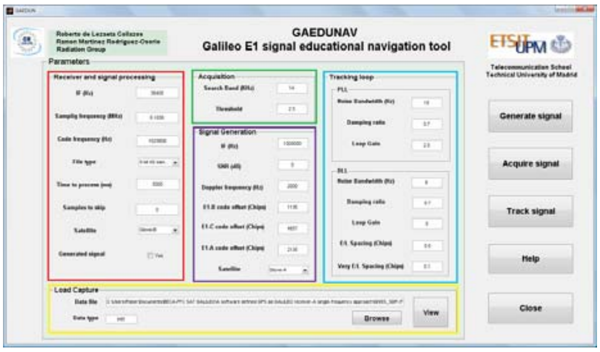
Fig. 13. DLL performance with the capture of Giove A signal on 30/06/09 in Madrid with an elevation of 52º.

III.IV GAEDUNAV

GAEDUNAV (GAlileo EDUcational NAVigation Tool) is an interface designed in Matlab to integrate and show in a friendly and simple way the software receiver developed at the Grupo de Radiación of the Telecommunications Eng. School in the Universidad Politécnica of Madrid. This tool is aimed to be used as a complement in education at different levels and has been built on the pursuit of helping in the understanding the operation of GNSS receivers and in particular Galileo E1 signal receivers.

The interface allows the user to follow and perform the basic tasks of a GNSS receiver, which are acquiring and tracking. The signals to be processed may come either from a capture file made during a satellite pass with a GNSS front-end, or from the signal generator module that is able to generate the E1 Galileo signal from both satellites (Giove-A and Giove-B). The results of each stage are given in different and independent windows.

The data entries occupy the larger part of the interface's main window since they control the performance of the receiver. The organization is simple and intuitive; the groups are separated in different categories depending on the module they control (Fig. 14): General parameters (left and red), acquisition parameters (centre-up and green), tracking parameters (right and blue), capture parameters (down and yellow) and finally generation parameters (centre-down and purple).



Roberto de Lezaeta Collazos
Ramon Martinez Rodriguez-Osorio
Radiation Group

GAEDUNAV
Galileo E1 signal educational navigation tool

Telecommunication School
Technical University of Madrid

Parameters

Receiver and signal processing
- IF (Hz): [illegible]
- Sampling frequency (MHz): [illegible]
- Code frequency (Hz): [illegible]
- File type: [illegible]
- Time to process (ms): [illegible]
- Samples to skip: [illegible]
- Satellite: [illegible]
- Generated signal: □ Yes

Acquisition
- Search Band (KHz): 14
- Threshold: 2.5

Signal Generation
- IF (Hz): 1000000
- SNR (dB): [illegible]
- Doppler frequency (Hz): 2000
- E1.B code offset (Chips): [illegible]
- E1.C code offset (Chips): [illegible]
- E1.A code offset (Chips): [illegible]
- Satellite: [illegible]

Tracking loop

PLL
- Noise Bandwidth (Hz): [illegible]
- Damping ratio: [illegible]
- Loop Gain: [illegible]

DLL
- Noise Bandwidth (Hz): [illegible]
- Damping ratio: [illegible]
- Loop Gain: [illegible]
- E-L Spacing (Chips): [illegible]
- Very E-L Spacing (Chips): [illegible]

Load Capture
- Data file: [illegible]
- Data type: [illegible]

Fig. 14. Graphical User Interface of GAEDUNAV.

IV. SIGNAL CAPTURES AND MEASUREMENTS

IV.I Planning and scenarios

In order to evaluate the performance of the receiver in a real scenario, a measurement campaign with the signals of Giove-A and Giove-B was carried out. Theses tests contribute to verify the operation of the receiver out of a controlled simulation scenario.

Signal captures were made in different locations of the ETSIT-UPM in order to provide the operation of the receiver in typical scenarios: LoS (Line of Sight), NLoS (Non Line of Sight) and indoor. These scenarios are shown in Fig. 15. ETSIT-UPM is located in (40.3968N, -3.7134E).

In the LoS scenario, the receiver antenna is located in the rooftop of building C. No obstacles in the surroundings alter the reception of the signal. NLoS measurements were done in the staff parking, an outdoor space surrounded by three tall buildings. Finally, indoor tests were carried out in the main hall of Building A and in the Laboratory of the Grupo de Radiación.

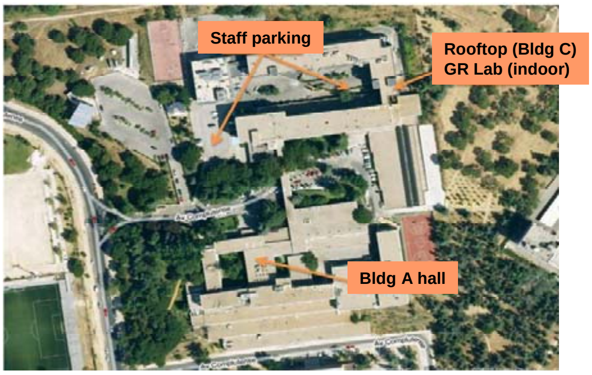


Fig. 15. Measurement location and scenarios.

We started with the measurements in LoS as the most favourable scenario for a signal reception without obstacles. After verifying the performance of the receiver, measurements in the other scenarios were done.

IV.II Hardware and setup

The equipment used to perform the signal captures is shown in Fig. 16. The main device is a single-frequency GNSS front-end (SiGE GN3S v2)[9] commercialized by Sparkfun Electronics which was used with a 26 dB's gain antenna plus low noise amplifier (LNA) to pre-process and sample the signal. Several software tools like the tracking program NOVA[10] were also used to establish the conditions required on each pass and generate a measurement schedule, as well as to perform the captures.

The drivers of the front-end gave some problems at the beginning but after the first tests they proved to be reliable. The rest of the equipment worked properly from the first time allowing the realization of many captures with nice results which were later processed with the receiver with equally satisfactory results.

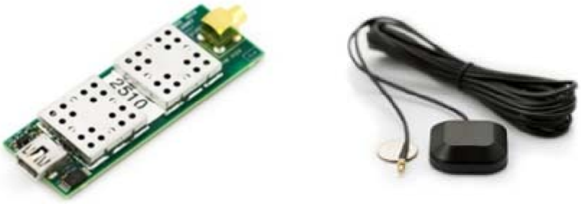


Fig. 16. GNSS RF front-end and antenna.

IV.III Results

The measurement campaign comprised a large series of captures on different scenarios which permitted to verify the correct functioning of the receiver in most cases, except in those with high indoor distortions. Almost all the PSDs showed the same form and amplitude with slight variations from one to another (Fig. 17). This was due to presence of the same navigation signals used on the L1 band (GPS and Giove-A/B). The results of these acquisitions are shown in Fig. 18.

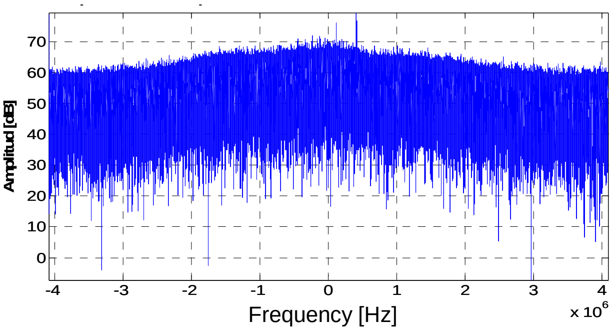


Fig. 17. Power spectrum of the capture carried out on 22/07/09 at 17.53 in Line of Sight conditions.

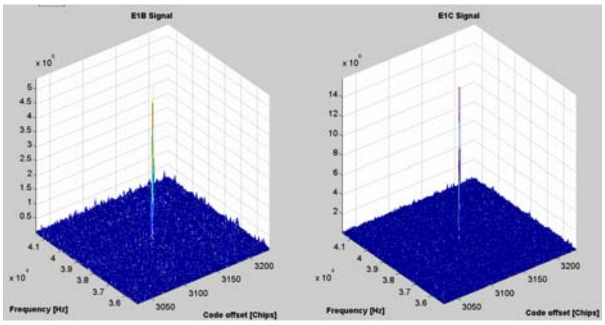


Fig. 18. Correlation matrices of the capture carried out on 22/07/09 at 17.53 in Line of Sight conditions.

Nevertheless, some indoor captures and their PSDs showed high power interfering signals which made not possible the detection of the navigation signals in the acquisition process (Fig. 19). Their origins were unknown, but all were located around 410 KHz to the right of the central frequency of the band (1575.42 MHz). It was concluded that such interferences were not proper of the measure equipment since they would be present in every capture made. Therefore, it is conceivable that the source of these components might be located within the environment of the capture place (Laboratory of the Grupo de Radiación of the Universidad Politécnica de Madrid).

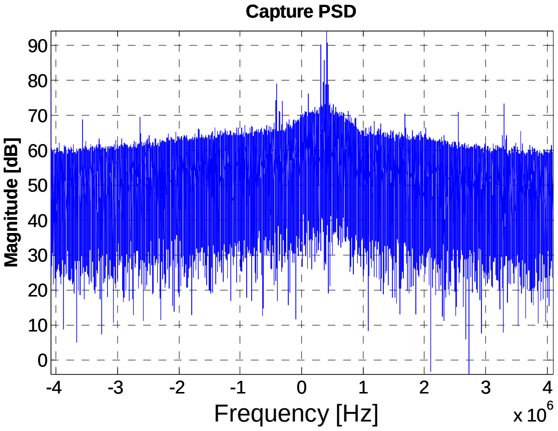


Fig. 19. Power spectrum of the capture carried out on 01/10/09 at 13.55 in indoor scenario.

V. CONCLUSIONS

Results of the measurement campaign in different scenarios (outdoor and indoor) show that the software receiver concept is an interesting and versatile approach towards the implementation of commercial GNSS receivers either analog or software radio. With minimal changes, the designed receiver shall be used to acquire and track signals of IOV satellites and the final Galileo constellation.

Regarding the acquisition, it is important to mention that this module is adequate for E1B and E1C channels, as they have been satisfactorily demodulated in simulation and from measured signals. Similarly, the scheme used for tracking is valid for E1B as navigation data have been obtained from captured signals.

GAEDUNAV, the simulation tool used for receiver design and evaluation represents a feasible approach to learn about GNSS receiver design as it shows the signals in every receiver point and see the impact of parameter values in the demodulation process. We plan to use it as support tool during GNSS lecturers.

As part of the research work done, a set-up for capturing GNSS signals has been implemented. It allows the reception of signals in a wide range of scenarios in order to verify the designed receivers in different conditions. Tests showed that in indoor scenarios with interference demodulation of signals is not possible and further improvements in the front-end and receiver are required.

Finally, it important to mention some of the impairments we dealt with during the project. First, we experienced the switch-off of Giove-A signals from July 7th to Sept 7th, 2009. The service interruption forced us to change the project schedule and produced some delays. As a positive consequence, we were forced to modify the software receiver to process Giove-B signals. Secondly, we had some problems with the front-end drivers and the problem was solved without any support from the supplier.

ACKNOWLEDGMENTS

Authors wish to thank MICINN (Ministerio de Ciencia e Innovación) under the CROCANTE project (ref. TEC2008-06736/TEC), and Cátedra Isdefe in ETSIT-UPM for the funding of this research work.

[1] Elliot D. Kaplan. *Understanding GPS: Principles and applications*, 2nd Edition, Artech House, 2006.

[2] ESA, http://www.esa.int/esaNA/ESAAZZ6708D_galileo_0.html.

[3] Kai Borre, Dennis M. Akos, Nicolaj Bertelsen, Peter Rinder, Søren Holdt Jensen, *A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach*, Birkhäuser, 2007.

[4] "Giove - A + B Navigation Signal-in-Space Interface Control Document", Galileo Project Office, ESTEC 2008.

[5] G. W. Hein,, J-A. Avila-Rodriguez, S. Wallner, A. R. Pratt, J. Owen, J-L. Issler, J. W. Betz, C. J. Hegarty, S. Lenahan, J. J. Rushanan, A. L. Kraay, T.A. Stansell, "MBOC: The New Optimized Spreading Modulation Recommended for GALILEO L1 OS and GPS L1C", IEEE/ION PLANS 2006, April 24-27, 2006, San Diego, California

[6] Olivier Julien, "Design of Galileo L1F Receiver Tracking Loops", PhD Thesis, University of Calgary, Dpt. Of Geomatics Enineering. Canada. July 2005.

[7] Jinghui Wu, Andrew Dempster. "Galileo GIOVE-A Acquisition and Tracking Analysis with a New Unambiguous Discriminator". Proc. IGNSS Symposium 2007. The University of New South Wales, Sydney, Australia, Dec. 2007.

[8] Cheon Sig Sin, Jae Hyun Kim, Sanguk Lee, Jae Hoon Kim. "A Software Receiver Implementation for GPS L1 and Galileo E1 Signal". Proc. ASMS. Bologna, Italy, Aug. 2008.

[9] SiGE Semiconductor, GPS Products, http://www.sige.com/products/gps/details.html.

[10] NOVA, Satellite Tracking Software, http://www.nlsa.com.