# STUDY OF A SOFTWARE GALILEO RECEIVER

Bart Scheers, Stijn Ceyssens, Alain Muls
Royal Military Academy
Department CISS
Renaissancelaan 30
B1000 Brussels, Belgium
E-mail: {bart.scheers, alain.muls}@rma.ac.be, stijn.ceyssens@mil.be

## ABSTRACT

*In this paper, the development of a software Galileo receiver is presented. The Galileo receiver consist of a Real-Time Spectrum Analyser, serving as an RF front-end to down convert the signal from RF to IF, and a software implemented in Matlab, running on a standard PC, to extract the navigation bits out of the captured signal. The concept of a Galileo software receiver is equivalent to a traditional GPS receiver, containing software routines for signal acquisition, carrier and code tracking, and data decoding. In order to test the software receiver, the code of the Galileo test satellite GIOVE-A is pre-generated and stored in memory. This code is modulated on a locally generated carrier and allows to test the acquisition, the tracking and data demodulation operations. In a next step, the software receiver was tested, with success, on a signal coming from the GIOVE-A satellite, captured with a Real-Time Spectrum Analyser. In the paper, the different parts of the software receiver are described in detail, and the choices made are motivated. Where possible, a comparison with the GPS system is made and advantages of the Galileo signals are highlighted.*

## INTRODUCTION

The success of the US Department of Defence funded GPS program has led to an increased interest and dependence on Global Navigation Satellite System (GNSS) services. Political, technological, economic and social considerations have led to the Galileo program. This initiative of the European Union will offer users global positioning and timing services, similar and interoperable with the GPS system. The core Galileo constellation of 30 satellites, supported by a world wide ground segment, will offer several navigation services aimed at the mass market and professional or safety-of-life applications. The Galileo systems uses higher chipping codes than the current GPS system, resulting in larger bandwidths and higher positioning accuracies. The Open Service (OS) of the Galileo system, aimed at the mass market users, is interoperable with the GPS Standard Positioning Service (SPS). The Galileo OS uses Binary Offset Codes (BOC) to spectral separate its signals from the (current and planned) GPS signals. The development of the software Galileo receiver focuses on the OS code transmitted on the L1 (1575.42 MHz) central frequency.

## THE GALILEO L1 OS SIGNAL [3]

In general, GNSS-signals are composed of three parts: a navigation message, a spreading code and a carrier. In order to avoid interference with the GPS C/A (Coarse/Acquisition) signal, transmitted at the same carrier frequency, a Binary Offset Code BOC(1,1) is applied to the Galileo OS signal to introduce an offset relative to the central L1 carrier frequency, as represented on figure 1.

In the receiver, a replica of the spreading code and the carrier are needed. Therefore we studied the L1 OS signal used by the GIOVE-A test satellite in detail, pre-generated it at the same sample frequency as the digitised received satellite signal after the IF stage.

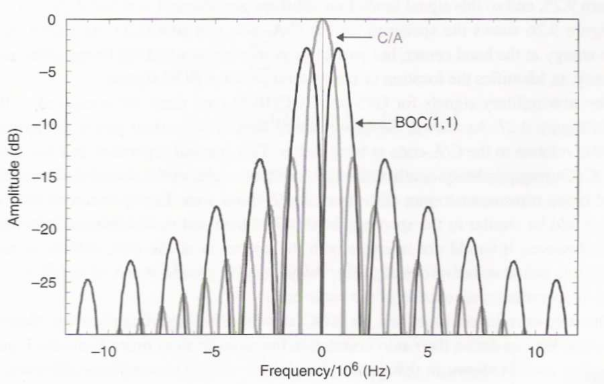


**Figure** 1: The spectrum of the GPS C/A signal and the Galileo OS signal.

In baseband, the L1 signal is composed out of three codes: the $c_{L1-A}$, the $c_{L1-B}$ and the $c_{L1-C}$ code. The $c_{L1-A}$ code is used for the Public Regulated Service and is out of the scope of this paper. The $c_{L1-B}$ is the actual L1 OS code that will be used to spread the navigation data. The period of the code is 4 ms, with a chip rate of 1.023 Mchips/s and a code length of 4092 chips. The $c_{L1-B}$ code is the result of a summation of 2 linear shift registers with length 13. The feedbacks and the initial seeds are shown in figure 2.

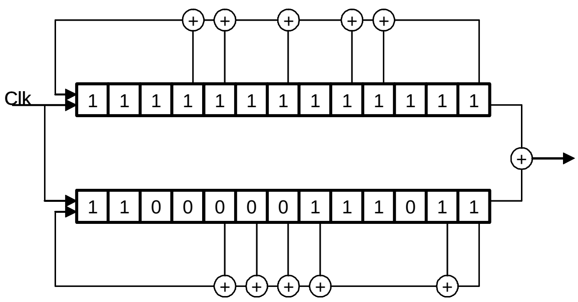


**Figure** 2: Code generation of the $c_{L1-B}$ code

The $c_{L1-C}$ code serves as pilot code. The code will not be modulated by navigation data and has a length of 200 ms, which is quite long. In practice, the pilot code can be used by a receiver during the acquisition phase. Because the code length is very long, this could result in a very noise resistant acquisition. However, in our implementation, the acquisition phase is based on the 4 ms long $c_{L1-B}$ code.

In a next step, the three codes $c_{L1-A}$, $c_{L1-B}$ and $c_{L1-C}$ are multiplied with the navigation data bits $D_{L1-A}$ and $D_{L1-B}$ and a BOC-code to obtain three signals $e_{L1-A}(t)$, $e_{L1-B}(t)$ and $e_{L1-C}(t)$ :

$$e_{L1-A}(t) = \sum_{i=-\infty}^{+\infty} c_{L1-A}.D_{L1-A,[i]}.\text{sign}(\cos(2\pi f_{BOC(15,2.5)}t))$$

$$e_{L1-B}(t) = \sum_{i=-\infty}^{+\infty} c_{L1-B}.D_{L1-B,[i]}.\text{sign}(\sin(2\pi f_{BOC(1,1)}t))$$

$$e_{L1-C}(t) = \sum_{i=-\infty}^{+\infty} c_{L1-C}.\text{sign}(\sin(2\pi f_{BOC(1,1)}t)).$$

The BOC(1,1) code is a block signal with a period of 1 chip. The multiplication with this BOC code has two consequences. The first is that the spectrum of the L1 OS signal will have an offset relative to the L1 carrier as shown in figure 1. The second is that the autocorrelation of the code will have a steeper peak, which will result in a more precise code tracking. In the GPS C/A signal, this correlation peak will be, relative to the length of one chip, twice as large.

After the BOC sub-carrier modulation, the three signals $e_{L1-A}(t)$, $e_{L1-B}(t)$ and $e_{L1-C}(t)$ are multiplexed into one complex baseband signal with constant envelope:

$$\begin{aligned} s_{L1}(t) &= s_{L1-I}(t) + j.s_{L1-Q}(t) \\ &= \frac{1}{3}\{\left[\sqrt{2}e_{L1-B}(t) - \sqrt{2}e_{L1-C}(t)\right] \\ &\quad + j.\left[2.e_{L1-A}(t) + e_{L1-A}(t)e_{L1-B}(t)e_{L1-C}(t)\right]\} \end{aligned}$$

The constellation of $S_{L1}(t)$ is represented in figure 3.

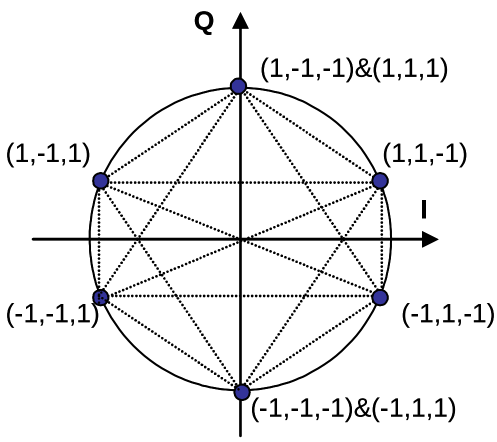


**Figure** 3: Hexagonal constant envelope constellation.

## THE RECEIVER

### The structure of the receiver.

The structure of the Galileo software receiver is equivalent to the structure of a traditional GPS receiver [1],[2]: an acquisition block for the estimation of the Doppler frequency and the code delay, a PLL for the carrier tracking and a DLL for the code tracking. Once the correct carrier replica and code delay are obtained, the navigation bits can be extracted from the incoming signal.

### The acquisition

The aim of the acquisition block is to estimate the Doppler shift and the code delay in the digitised received satellite signal. This is done based on the ambiguity function. The result of this function is shown in Figure 4. The peak in the two dimensional plot indicates the Doppler shift estimate and the code delay estimate. For the Galileo signals, the ambiguity function is evaluated with a resolution of 250 Hz on the Doppler shift, meaning that a PLL is needed to fine-tune and track the carrier. The correlation to determine the code phase is performed on segments of 4 ms, corresponding to the length of the spreading code $c_{L1-B}$. Compared to GPS system, with a C/A code length of only 1 ms, the Galileo receiver will have a better resistance to noise.

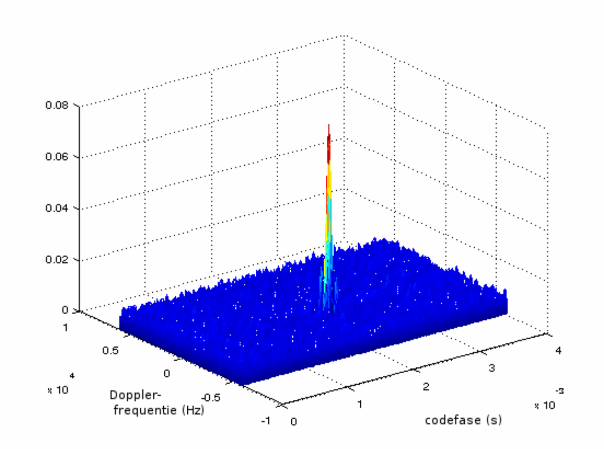


**Figure** 4: Result of the acquisition phase

### The carrier and code tracking

The carrier and code tracking are respectively done by a Phase-Locked Loop (PLL) and a Delay-Locked Loop (DLL). The DLL, represented in figure 5 is based on an Early-Late-Prompt sampler. For the PLL, a Costas loop is used. The particularity of the two loops is that they are entwined, meaning that the output of one loop serves as input of the other and vice-versa.

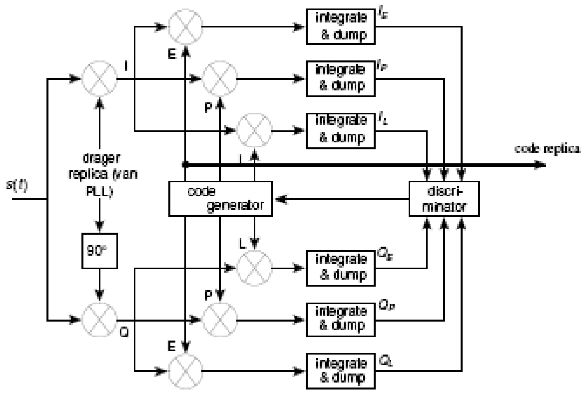


**Figure** 5: The DLL, based on an Early-Late-prompt sampler

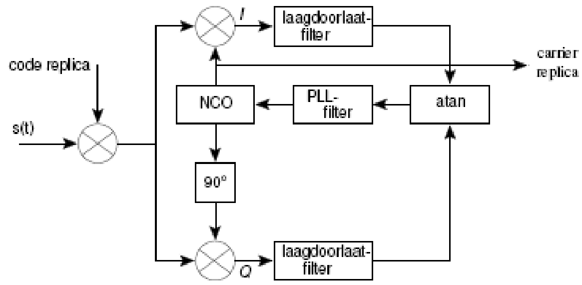


**Figure** 6: The Costas loop.

In the software implementation, the DLL is executed on segments of 4 ms. The Costas Loop is executed on incoming segments of 0.5 ms, to obtain a faster convergence. The quadrature phase detector of the Costas loop generates its error by taking the $\tan^{-1}$ of the ratio Q over I, integrated over the 0.5 ms segments. The loop filter in the Costas loop is the same as for the software GPS receiver in [2]. The frequency response function of the filter is given by

$$F(z) = \frac{(C_1 + C_2) - C_1 z^{-1}}{1 - z^{-1}},$$

where $C_1$ and $C_2$ are given by

$$C_1 = \frac{1}{k_0 k_1} \cdot \frac{8\varsigma\omega_n t_s}{4 + 4\varsigma\omega_n t_s + (\omega_n t_s)^2}$$

$$C_2 = \frac{1}{k_0 k_1} \cdot \frac{4(\omega_n t_s)^2}{4 + 4\varsigma\omega_n t_s + (\omega_n t_s)^2}$$

with $\omega_n$ the natural frequency, $\varsigma$ the damping ratio, $t_s$ the sampling period, $k_0$ the gain of the phase comparator and $k_1$ the gain of the NCO (numerical controlled oscillator). In figure 7, the locking of the Costas loop is shown on a 100ms long captured signal from the GIOVE-A satellite. The figure presents the output of the NCO over time. After about 20 ms, the Costas loop has retrieved the exact carrier.

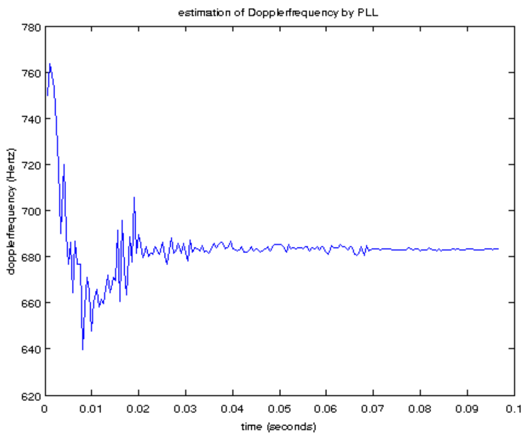


**Figure** 7: Estimation of the Doppler frequency by the Costas loop on a captured signal

## Results

The software receiver was evaluated on synthetic data with different SNR. After successful testing, the software receiver was tested on a real signal sample from the GIOVE-A satellite. The signal was captured and saved on disk using a Real-Time Spectrum Analyser, as an RF front-end, and a normal active GPS antenna. The software Galileo receiver allowed to extract the data bits from the Galileo navigation message as shown on Figure 8, representing 96 ms of a navigation message (24 bits), confirming successful operation of the implemented acquisition and tracking loops.

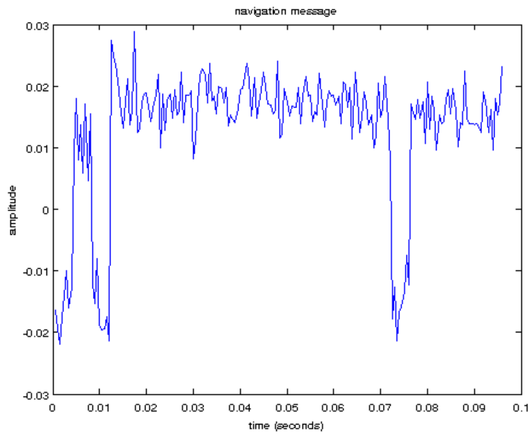


**Figure** 8: The result of the receiver, representing 96 ms of a navigation message

## CONCLUSIONS AND FUTURE WORK

In this work, the development of a software Galileo receiver is presented. The structure of the Galileo software receiver is based on that of a traditional GPS receiver. Most of the operations are done on data segments of 4 ms, except for the PLL. The receiver has been tested with success on synthetic data and on real data, coming from the GIOVE-A test satellite. The concept proves that, from the receiver point of view, the Galileo system and the GPS system are comparable and that the two systems can easily be integrated into the same receiver.

## BIBLIOGRAPHY


[1] Kai Borre, Dennis M. Akos, Nicolaj Bertelsen, Peter Rinder, and Soren Holdt Jensen: *A software defined GPS and Galileo receiver, A single frequency approach*. Birkhäuser Boston, 2007.

[2] Falin Wu, Kefei Zhang and Akio Yasuda: *GPS signal acquisition and tracking using software GPS receiver*. Proceedings of the Eighth International Symposium on Signal Processing and Its Applications, 2005. Volume 2, Issue , August 28-31, 2005 Page(s): 843 - 846

[3] Galileo Project Office. Giove-a navigation signal-in-space, interface control document, March 2007.

[4] Stijn Ceyssens: *Studie en ontwerp van een software Galileo ontvanger*. Thesis, Royal Military Academy, Brussels, 2007