Implementation of a GNSS-R Payload Based on Software-Defined Radio for the ^{3}CAT-2 Mission

Roger Olivé, Adrià Amézaga, Hugo Carreno-Luengo, Hyuk Park and Adriano Camps,

Abstract—The ^{3}CAT-2 nanosatellite aims at demonstrating global navigation satellite system reflectometry (GNSS-R) techniques for spaceborne applications in the small form of a six-unit CubeSat. There are many challenges involved from a size, processing, and power perspectives. The proposed solution for the payload uses a software-defined radio (SDR) connected to a nadir looking array of dual-band and dual-frequency and dual-polarization antennas to capture the reflected GNSS signals and to a zenith looking patch antenna to capture the direct ones. The SDR is controlled by the payload computer, which retrieves the binary samples and processes the raw data to obtain delay-doppler maps (DDMs) via various techniques. DDMs are then compressed using the fully adaptive prediction error coder algorithm, producing an output more suitable for the limited downlink capabilities of these small platforms.

Index Terms—Conventional global navigation satellite system reflectometry (GNSS-R), CubeSat, delay-doppler map (DDM), interferometric GNSS-R, software-defined radio (SDR).

I. INTRODUCTION

^{3}CAT-2 is a six-unit CubeSat scheduled for launch in July 2016 [1]. The main objective of this nanosatellite is the demonstration of its payload PYCARO: the P(Y) and C/A ReflectOmeter. PYCARO implements a novel dual-frequency (L1 and L2) and dual-polarization (left- and right-hand circular polarizations, LHCP and RHCP) multiconstellation (GPS, GLONASS, Galileo, and BeiDou) reflectometer [2], [3] [(global navigation satellite system reflectometry (GNSS-R)] whose scientific output consists of delay-Doppler maps (DDM).

DDMs are the most complete form of GNSS-R observable, and consist of the time-domain correlation between two sequences of GNSS signals[1] for different Doppler frequency shifts. While DDMs have a wide range of applications, this receiver focuses on altimetry and scatterometry for sea state, wind speed, and vegetation biomass.

Multiple GNSS-R instruments have been implemented for current or future space-based missions. For instance, TechDemoSat-1 [5] was launched in July 2014 with a GNSS remote-sensing instrument following the success of the UK- DMC experiment one decade earlier. That mission can track for the first time L2 signals from space in addition to L1 C/A ones. It will provide the baseline instrument to be used in the Cyclon Global Navigation Satellite System (CYGNSS) mission [6], which consists of a constellation of eight satellites, each one equipped with a delay-Doppler mapping receiver, in a low incli- nation orbit. The main objective of CYGNSS is sensing ocean wind and wave conditions in tropical cyclones for weather fore- cast and emergency management. It is scheduled for launch in October 2016. PYCARO is the evolution of other GNSS-R instruments developed at the Universitat Politècnica de Catalunya (UPC), such as Passive Advanced Unit (PAU) [7], a hybrid total power radiometer and GNSS-R receiver that features two redundant receivers, and it is based on a Virtex-4 FPGA setup, that was meant to be a secondary payload of the INTA Microsat-1. Its precursor griPAU [8], [9] was the first GNSS-R instrument to compute DDMs in true real time.

Now PYCARO introduces UPC GNSS-R instruments to the CubeSat form factor. Standard CubeSats are "inexpensive" research-oriented pico/nanosatellites with the elementary size of a 10-cm side cube (one-unit CubeSat) made of commercial off-the-shelf electronic components. Those satellites are putting space research at reach of companies and universities that would otherwise not be able to afford a regular space launch for their projects. ^{3}CAT-2 is one of the first six-unit CubeSats (3 × 2 10-cm side cubes), and the GNSS-R instrument has had to be adapted to those conditions. This situation imposes very restrictive requirements in terms of space and power, as well as the amount of data that can be downloaded. The orbit is expected to be sun-synchronous at 500 Km.

One half of the satellite is allocated from the start to critical subsystems: the main computer of the satellite also known as the on-board computer (OBC), the electrical power subsystem (EPS), the magnetotorquer-based attitude determination

---

[1]If the two signals are the direct and the reflected ones, then it is called interferometric GNSS-R or iGNSS-R. If the two signals are the reflected one and a locally generated replica of the transmitted one, then it is called conventional GNSS-R or cGNSS-R. If the P code is estimated from the L1 and L2 signals, and it is then cross correlated with the reflected signal, then it is called reconstructed GNSS-R or rGNSS-R [4].

Fig. 1. ³CAT-2 interior 3-D model. The white patch is the L1/L2 dual-polarization antenna to receive the direct signal.
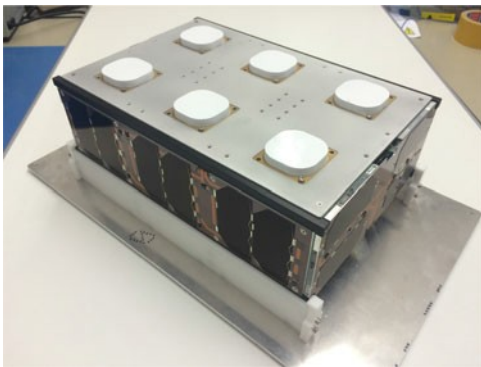

Fig. 2. ³CAT-2 engineering model. The side on top is the antenna array used by the GNSS-R payload to capture the reflected GNSS signals.

and control system, or the communications subsystem (COMS, based on a redundant UHF/VHF transceiver, and a S-band transmitter for the scientific data downlink). The remaining half, corresponding to the black box visible in Fig. 1, is the payload. Some of the critical subsystems have important power budgets, especially the attitude control and communications sub subsystems, which can cause power peaks of several Watts. Inaddition, since an important part of the satellite surface is required to place the nadir-looking payload antennas instead of solar cells (see Fig. 2), the power available for the payload is further decreased.

The GNSS-R payload has its own demanding requirements that need to be optimized as well. Computing a DDM requires a lot of repetitive mathematical operations such as fast Fourier transforms (FFT), which can easily amount to up to thousands of computations for a single DDM. Also, those operations are applied to long sequences of data (4 096 samples for 0.81 ms of data sampled at 5 MS/s), which further increases the time needed to complete them. In this study, the implementation of the conventional and interferometric GNSS-R, with configurable resolution in both time and Doppler axes is explained. Related to the time resolution, the sampling frequency has to at least match the Nyquist frequency of the GNSS signals of interest. In addition, an aggressive compression strategy is needed to

avoid saturation of the satellite downlink, which is expected to be operational for 10-min periods and with a data bandwidth of 115 kB/s at best, when there is contact with the ground station. Since the requirements are too demanding for the OBC (a Nanomind [10], based on a 40-MHz ARM7 processor), all the data capture and processing is handled by a separate computer, the payload computer, that communicates with the OBC. In this study, the scientific software scheme used by the ³CAT-2 payload computer to sample GNSS data and compute the DDM observables is described. Section II shows a summary of the conventional and interferometric GNSS-R techniques implemented. Next, Sections III and IV detail the hardware platform and the implemented scientific software, respectively. Finally, preliminary results and conclusions are presented in Sections V and VI to evaluate the current status of the instrument.

II. IMPLEMENTED GNSS-R TECHNOLOGIES

According to the mission objectives, the main goal of the payload is to generate DDMs from which geophysical parameters can be extracted. The DDM is a 2-D representation of the correlation between two sequences as a function of relative time delay and frequency offset [4].

Like most navigation receivers, reflectometers correlate during a short period of time, usually around 1 ms, but then, they perform incoherent averaging so as to reduce the noise. DDMs are computed by performing several cross correlations of two signals for a range of frequency shifts, which is also used to estimate and compensate the Doppler frequency between both signals and a frequency offset. In GNSS-R, one of the signals is the GNSS signal reflected on the Earth's surface. The other signal, can be either another real GNSS signal (i.e., the direct one) or a locally generated replica of the transmitted one.

There are several techniques depending on the choice of signals to correlate, and this paper covers two of them. If the received reflected signal S_R is correlated with a locally generated replica copy of a GNSS constellation satellite code a, it is called conventional GNSS-R (1). If the correlation of S_R is with the directly received signal S_D, it is interferometric GNSS-R (2)

$$Y^c(t,\tau,f_d) = \int_t^{t+T_i} S_R(t')a^*(t-\tau)e^{-j2\pi(f_C+f_d)t'}\mathrm{d}t', \qquad (1)$$

$$Y^i(t,\tau,f_d) = \int_t^{t+T_i} S_R(t')S_D^*(t-\tau)e^{-j2\pi(f_C+f_d)t'}\mathrm{d}t', \qquad (2)$$

where τ is the delay, f_d the Doppler frequency shift, t the initial coherent integration time, f_C the carrier frequency, and T_i is the coherent integration time.

In the payload computer, this correlation is calculated by computing the Fourier transform of both signals and multiplying a channel with the Doppler estimation compensation by the complex conjugate of the remaining one. The result can be reverse converted to time domain again in order to obtain the correlation. The entire capture and processing flow is sketched in Figs. 3 and 4 for the conventional and interferometric modes, respectively. If the process is repeated for different Doppler frequency shifts (for the same data stream), and those iterations are stacked, a 2-D plot is obtained: the DDM.
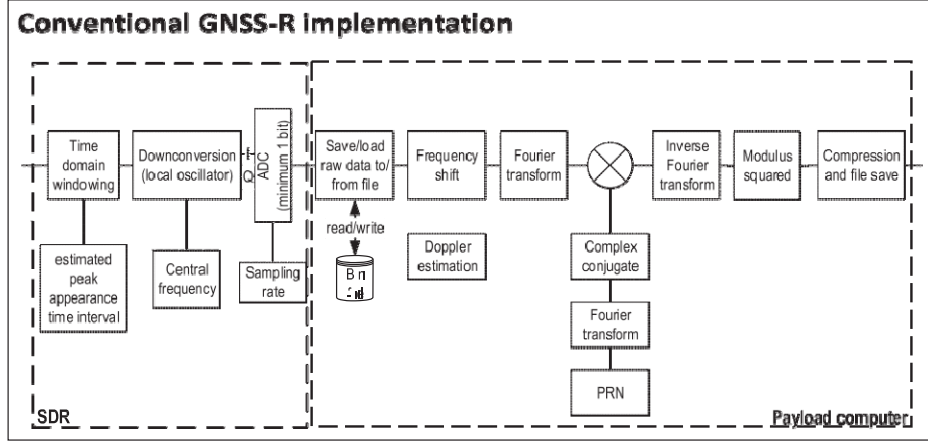
Fig. 3. Acquisition and processing flow in the conventional GNSS-R mode. ADC stands for analog-to-digital converter, and PRN stands for pseudorandom noise.
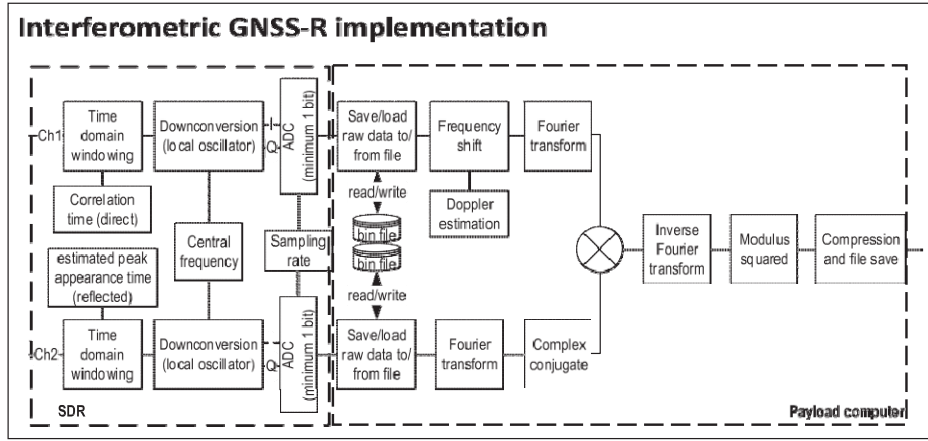

Fig. 4. Acquisition and processing flow in interferometric GNSS-R mode. ADC stands for analog-to-digital converter, and PRN stands for pseudorandom noise.

However, the reflected signal is expected to be very weak, resulting in a very poor signal-to-noise ratio. To increase it, incoherent averaging is performed

$$\left\langle \left| Y^{c,i} \right|^2 \right\rangle (\tau, f_d) \triangleq \frac{1}{N_i} \sum_{n=1}^{N_i} \left| Y^{c,i}(t_n, \tau, f_d) \right|^2, \qquad (3)$$

where t_n is the delay (t) adjustment applied when correlating segment n.

Another source that contributes to the need of incoherent averaging is the presence of multiple scatterers over the surface with different phases, that creates the so-called speckle noise (multiplicative noise), which is of special prominence over the ocean [4].

Similarly to the conventional GNSS-R, a third technique known as reconstructed GNSS-R can take advantage of $P(Y)$ codes being transmitted at L1 and L2 [11], [12]. While it is one of the goals for ^{3}CAT-2, the description and results presented in the next sections focus on cGNSS-R and iGNSS-R.

III. HARDWARE IMPLEMENTATION OF THE GNSS-R PAYLOAD

The final GNSS-R instrument is contained within a small box with a size roughly equivalent to the interior of three CubeSat units, forming an L shape. That box, shown in Fig. 6, integrates almost all the necessary hardware, including the zenith antenna structured as shown in Fig. 5, which is used to capture the reflected GNSS signals and has been built into the separate panel covering one of the wide sides of the CubeSat seen in Fig. 2.

Those antennas are custom-made noncommercially available parts. They are dual-band (L1+L2) active antennas (RHCP and LHCP) with a noise figure of $NF = 1.8$, fed at $V = 3.3$ V, and with an approximate current consumption of $I = 30$ mA.

The payload box specifically includes the payload computer, a dual-radio receiver board, its own payload electrical power supply board (different from the satellite EPS, from which it draws the power for the payload), and a radiofrequency (RF) switching matrix.

The payload computer is used as a computing platform and executes the software to manage all the scientific tasks, as well
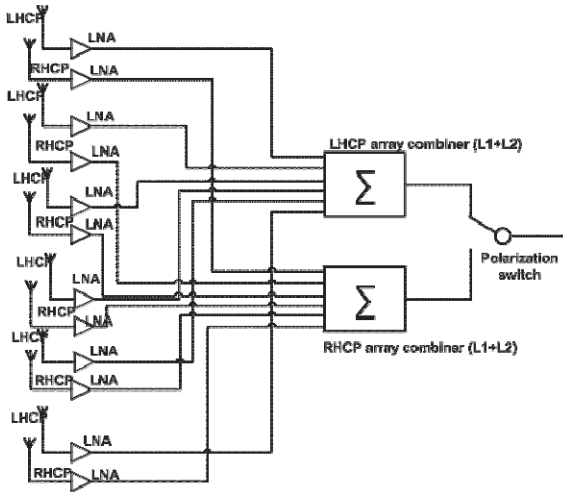
Fig. 5. Diagram of ³CAT-2's nadir-looking dual-frequency and dual-polarization antenna array.
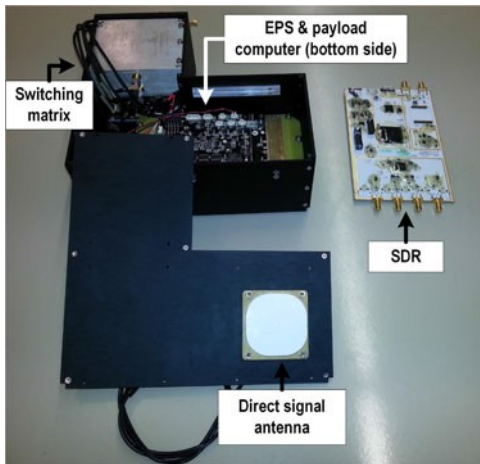

Fig. 6. ³CAT-2 GNSS-R payload: the switching matrix routes the signals from the LHCP and RHCP down-looking antenna array, at different polarizations, to the inputs of the SDR.

as the communication with the OBC through a serial connection. The dual-receiver board is a software-defined radio (SDR), and it is used to sample the direct and reflected GNSS signals. The payload EPS board is fed by the satellite EPS and adapts the incoming power to the requirements of each device of the payload. It is controlled by the payload computer through an I^2C interface to switch ON and OFF circuits and save power. The RF switching matrix is used to select between various combinations of incoming direct and reflected signals with different polarizations in order to retrieve vegetation biomass information [2].

The two components that have the most important influence on the scientific software design are the payload computer and the radio receiver. The scientific software is very demanding in terms of computational power to meet the requirements of real-time sampling, and postprocessing. For this setup, in addition, the CubeSat restrictions must be taken into account: low power,

very limited space, robustness, and extreme thermal and vacuum conditions. Balance between those requirements is a key design consideration in the implementation of the scientific software.

Last, but not least, flight heritage is desirable, but not always an option. To face that the team qualifies in-house at UPC Nanosatlab [13] the components without known flight heritage, as well as the system as a whole. The in-house validation [14] includes vacuum, thermal cycling and vibration tests to ensure that the system can survive in the expected launch and orbital conditions [15]. In addition, a radiation analysis [16] has been conducted to predict the radiation environment and adapt the device shielding accordingly.

A. Payload Computer

Among many embedded ARM-based platforms, the Gumstix Overo Ironstorm module [17] with a Tobi [18] expansion board was selected for several reasons. First, Gumstix Overo boards have flight heritage [19], and they are a good compromise in terms of power (under 2 W) and size (the Tobi board has a size of 40 × 105 mm), while being computationally performant. In addition, the Overo, with a DM3730 [20] system on chip, is compatible with Linux, which is necessary to use the SDR drivers. Its large pool of RAM is also very welcome for memory intensive applications, and the high-speed USB 2.0 is needed to stream data from the SDR.

B. Dual-Radio Receiver

Since the GNSS-R payload is a technology demonstrator, it is desirable to have as many adjustable parameters as possible in all the stages, including sampling. That allows for a deeper testing and reconfiguration capabilities to take better advantage of the mission. SDRs provide this kind of flexibility as they offer a general purpose RF interface that can be programmed to tweak many parameters such as the central frequency, sampling rate, gain, or amplitude resolution. The sampling process is handled by the SDR and any processing work is off-loaded to the payload computer.

A considerable amount of options have flourished in the SDR market lately, with cost-effective alternatives suitable for both professional and amateur users. Many are enough from a spectral point of view to capture L-band signals. Sample size is not a concern either, since the resolution of amplitude information is not an important factor as long as frequency information can be decoded correctly. Bandwidth and sample rate ideally should be as large as possible to have the possibility to capture the signals with a wide spectrum. However, the Overo CPU, and its USB 2.0 interface [17] turn to be the bottleneck for the highest sampling rates. Those parameters become secondary when choosing ³CAT-2's SDR.

While full-duplex capabilities are not necessary (the payload only needs the reception mode), the selected SDR [21] MIMO capabilities make it very interesting for its dual-receiver mode, so all capture hardware is concentrated in a single board. Even though it is technically possible to use separate receivers for direct and reflected signals, that imposes added programming difficulty, capture synchronization challenges, and integration

| Parameter	Default Value
Sampling frequency	5 MS/s
Master clock frequency	10 MHz
Doppler resolution	500 Hz/pixel
Doppler range	$+-$ 5000 Hz
Coherent integration time	1 ms
Incoherent integration time	blocks of 50 coherent periods

Fig. 7. Example of an averaged DDM. A region akin to the part highlighted in red is preserved, along with its coordinates within the entire DDM. The rest can be discarded safely since there is no meaningful information.

hurdles such as an increased power consumption, or a larger instrument size making it harder to fit inside the satellite. Also, the product comes in a board-only fashion (without case), which eases the integration.

In [3]CAT-2, the SDR is used in dual-receiver mode (two simultaneous receiving channels) to enable the desired simultaneous conventional GNSS-R in both channels, as well as interferometric GNSS-R. However, in practice, the maximum sampling rate is limited by the USB 2.0 to 8 MS/s from both channels combined at 16-bit samples, or 16 MS/s at 8-bit samples.

With this SDR setup, the hardware can support both L1 and L2 bands by adjusting the central frequency configuration appropriately. While it could potentially capture other bands with GNSS signals (i.e., L5), the antennas are optimized specifically for L1 and L2 only.

IV. SOFTWARE IMPLEMENTATION

The scientific software is able to stream the input samples from the SDR, generate the DDMs, and compress them so that the payload orchestrator[2] can manage them later. It also accepts input parameters from the orchestrator to tweak the acquisition and creation of the DDM data, as summarized in Table I. It should be noted that incoherent integration is limited to blocks of 50 coherent integration periods to avoid blurring. Retracking is performed on ground.

All the software is designed to run on a custom Linux image with Python, mathematical libraries such as FFTW [22], as well as the USRP Hardware Driver (UHD) to control the SDR [23]. The main program of the scientific software set is implemented in Python for the sake of flexibility and quick development time, as it facilitates high-level modifications according to the changing scientific objectives of the mission. It enables easy manipulation of arrays and matrices of data, and there is a wide selection of mathematical libraries. As an interpreted language it is slower, but it allows for quick prototyping. Plus, the program bottlenecks can be addressed using alternate programming languages such as C++. Specifically, capture and performance-critical parts of the DDM computation are implemented in C++, and packaged so they can be called from Python.

The program reads the input arguments specifying the types of DDM to compute, and it decides whether it has to capture new raw data files or load existing ones. If new captures are needed,

---

[2]Scheduler of the scientific tasks. Program that controls the experiment, including the execution of the scientific software described here.
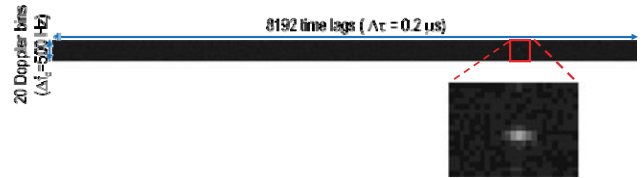
the SDR function is used to capture new GNSS data. Finally, the newly captured or already existing binary files are correlated as many times as desired in different modes following the same sequence, which is based on Figs. 3 and 4.

In the conventional mode, the first step is creating a GNSS code sampled at the same frequency as the input signal with the appropriate modulation. The main limiting factor in terms of software is the public availability of the employed code. Coupled with the antenna limitations outlined before, the list of possible codes can be narrowed down to GPS L1C/A, L2C, GLONASS L1 C/A, L2 C/A, Galileo E1B/C, and BeiDou B1I. Private codes such as L1 and L2 P codes can only be used in interferometric mode, with limited bandwidth. The second step is loading to RAM the data sequences that have to be used to compute a DDM (samples from one channel and the recreated GNSS code, or samples from both channels). Applying the techniques mentioned in Section II, the sequences are correlated using different Doppler frequency shifts with the specified frequency resolution. The correlations with different Doppler frequency shifts are stacked to form the rows of an output "picture" like the one shown in Fig. 7.

Different repetitions are accumulated for averaging and the peak position/pixel of the resulting 2-D array is detected: the algorithm finds the highest power point and a subpicture around it is preserved, along with the coordinates of that region in the original matrix. Depending on the level of trimming, the file size can be reduced by a factor up to 100. The final result is normalized to fully use the available dynamic range (16 bits) and the maximum and minimum values are preserved separately to restore the original range of values later. Last, but not least, the normalized matrix is compressed as a 16-bit gray scale Portable Network Graphics (PNG), or using the fully adaptive prediction error coder (FAPEC) algorithm [24].

A. SDR Capture Software

The raw data acquisition uses the C++ API provided by UHD (USRP Hardware Driver) [23], the open source driver for USRP devices. The bit rate is configured to 8 bits per phase/quadrature value instead of standard 16 bits because the extra precision is not necessary to extract the frequency information of the signal. Actually, most navigation receivers use just 1 bit. This bit rate allows higher sampling frequencies and better usage of the available USB bandwidth.

Samples are all always stored in RAM first because writing to the SD card is slower than the rate at which the samples

can be transferred through the USB interface.[3] Thanks to its improved transfer speed, the sampling rate can be raised to levels that would cause buffer overflows when using SD card storage. DM3730s LPDDR RAM can reach 200 MT/s,[4] which is enough to handle the incoming (8+8) bit 16 MS/s. This approach puts a hard limit on how much data can be captured in a single run (unused RAM space). However, it is more than enough to capture several seconds of contiguous RF data from each channel.

It is also very important to have the processor completely free from other tasks during streaming because of the amount of data being transferred. Otherwise, the driver buffer may overflow similarly to when using the SD card. In addition, the CPU clock must be kept as high as possible (1 GHz) during the whole process to ensure stability. If those conditions are met, internal tests have shown that 5 MS/s per channel is the highest sampling rate at which the payload computer can reliably stream the data using the current software.

B. DDM Computing Library

To improve performance, a custom Python library with a C++ implementation of the DDM processing algorithm has been created, taking advantage of FFTW to accelerate Fourier transforms, and *Boost.Python* [25] to integrate the methods with the Python software. Packaging computations within the custom C++ methods helps reducing the code overhead compared to implementing the same operations using Python and general purpose libraries.

C. FAPEC Compression

The FAPEC algorithm [24] has been implemented in the payload computer. FAPEC is a compression software developed by DAPCOM data services designed for space applications that claims better performance than the standard currently used in space communications (CCSDS 121.0) and better handling of outliers or noise in the data [24]. For instance, as compared to the popular GZIP compression (which is used extensively in Linux systems and has a nice compression/performance ratio), FAPEC produces 3.5 compression ratio, while GZIP is 2.5, and compresses three time faster on average [24]. In the case of DDMs, it can produce files around 20% smaller than the equivalent PNG when applied to DDMs in lossless mode.

The FAPEC algorithm is included in [3]CAT-2's version of the payload software as an external binary that is called from the DDM computing software to compress the raw DDM. Also, DAPCOM has optimized the project's version specifically for 16 bit files.

The many options provided by the compression and decompression programs include a toggle between little and big Endian files, a selector of the precompression algorithm filter order, the possibility of lossy compression, and an option to preserve some bytes uncompressed at the beginning of the file (useful

---

[3]The maximum speed of a high-end UHS-II bus is 300 MB/s, but minimum guaranteed speeds for an UHS Speed Class 3 SD card falls to 30 MB/s.
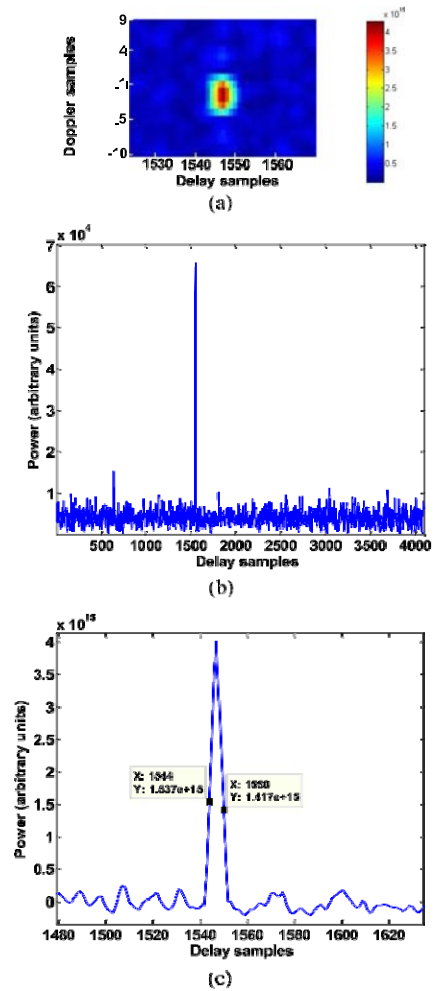[4]Megatransfers per second.

Fig. 8. C/A code only conventional DDM with an input power Pin = − 95 dBm. (a) Zoomed DDM peak. (b) 1-D slice of the DDM. (c) Zoom of (b) around the peak. (One delay sample = 200 ns, one Doppler frequency sample = 500 Hz.)

to implement headers). The lossy compression option discards a specified amount of least significant bits from each 16 bit sample.

V. PRELIMINARY RESULTS

A. DDM Characterization

In order to characterize the payload performance, a Rodhe & Schwarz SMU-200A [26] arbitrary signal generator with the option to generate artificial GPS signals in high-dynamic conditions has been used. The same signal was fed to both channels of the SDR during the tests. From those sample captures, conventional and interferometric DDMs were computed.

The Doppler axis maintains the same settings across all the tests in terms of dynamic range and resolution: from +4500 to − 5000 Hz around the chosen central frequency in 500 Hz steps.

The time axis depends on the coherent integration time set for each test, but the sampling frequency is always set to 5 MS/s. Figs. 8–10 show sample conventional and P-code interferometric, and P+C/A interferometric DDMs, respectively.
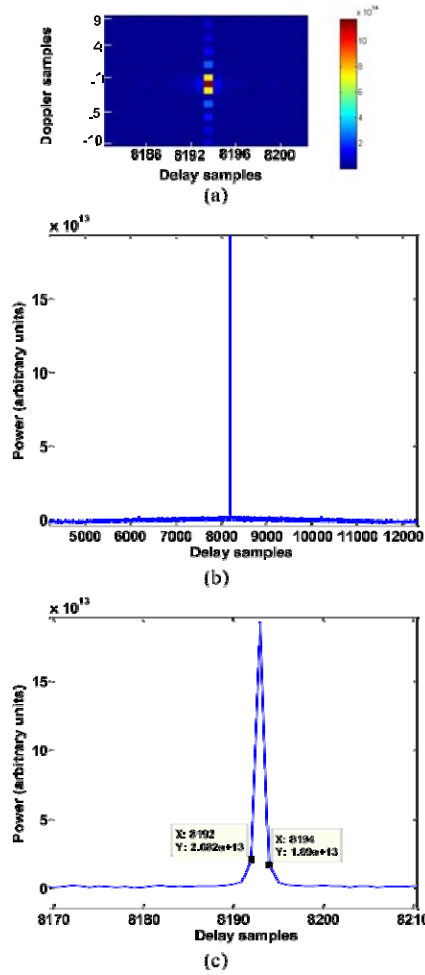
Fig. 9. P-code interferometric DDM with an input power Pin = − 95 dBm. (a) Zoomed DDM peak, with the time axis shifted to show the peak in the central part of the picture instead of the edges. (b) 1-D slice of the same time-shifted DDM. (c) Zoom of (b) around the peak. (One delay sample = 200 ns, One Doppler frequency sample = 500 Hz.)

Fig. 10. P+CA codes interferometric DDM with an input power Pin = − 95 dBm. (a) Zoomed DDM peak, with the time axis shifted to show the peak in the central part of the picture instead of the edges. (b) 1-D slice of the same time-shifted DDM. (c) Zoom of (b) around the peak. (One delay sample = 200 ns, One Doppler frequency sample = 500 Hz.)

For the C/A conventional DDMs (see Fig. 8), the peak shows the wider triangular shape of approximately six samples. When switching to interferometric P-only DDMs (see Fig. 9), the peak width shrinks to approximately one fifth of the former's width. If a more realistic setup is simulated with P plus C/A codes (see Fig. 10), the peak becomes thicker, around four samples, which positions it in a middle ground in terms of resolution.

1) Time and Frequency Deviations: The DDM peak can appear in arbitrary coordinates in both frequency and time axes. In the time axis, it is important to ensure a good synchronization between channels. The interferometric DDMs show the cross correlation of two ideally equal signals minus the phase difference and distortions of the reflected signal. Therefore, by feeding the same signal to both channels (see Fig. 10), synchronization issues can be assessed by checking the peak position in the DDM. Given that the SDR uses the same local oscillator for both channels, the synchronization is very tight: the peak appears in the leftmost or rightmost column of the DDM, so the issue can be considered minimal. Note that in Fig. 10, the peak has been shifted to the centre for better visibility.
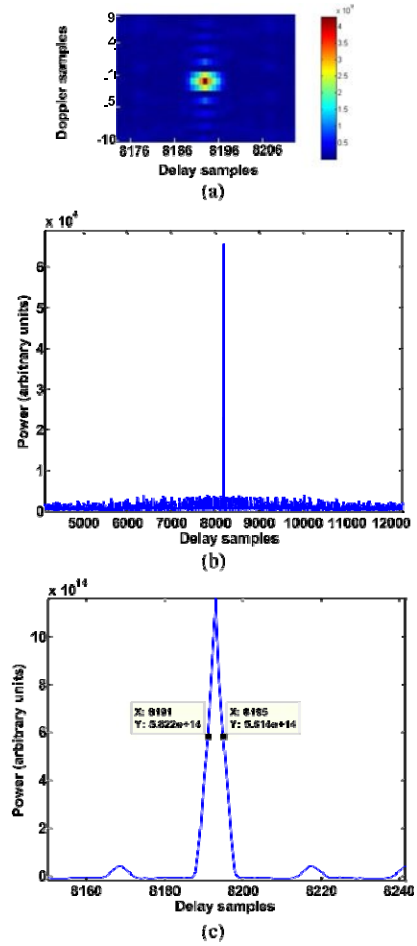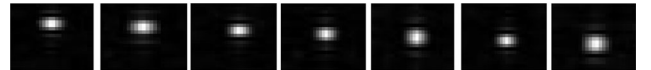

Fig. 11. Captures of different Doppler offsets on a C/A PRN 5 code emulated with the SMU-200A arbitrary signal generator. From left to right, 1500 to − 1500 Hz in 500 Hz steps. − 3000 Hz correction included (0-Hz Doppler is the central picture).

On the other side, errors in the frequency axis are a real concern, since the SDR LO will never tune at the exact desired frequency, and it is important to hit the right spot in order to find the peak successfully. Fortunately, the error is very systematic: peaks always appear with a +3000 Hz offset when testing the flight model SDR board and computing conventional DDMs. Therefore, the error can be compensated by subtracting 3000 Hz by software when defining the Doppler axis. Fig. 11 shows DDM peaks with varying Dopplers in 500 Hz steps, after compensation of the systematic error.

B. Processing Time

The time devoted to acquisition cannot vary much when changing the hardware since it is bound to real-time streaming.
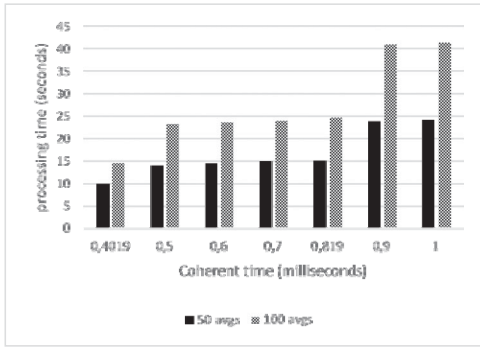
Fig. 12. Capture + processing time comparison for combinations of two incoherent times (50, 100 averages) with a selection of coherent integration times (in Gumstix Overo, [3]CAT-2 version of the instrument software).
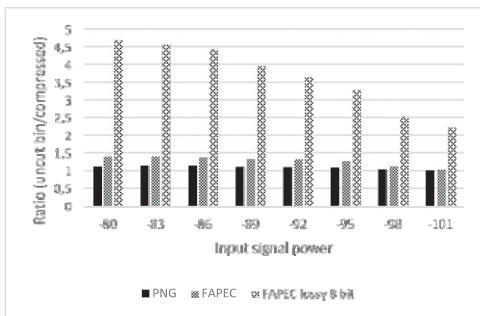

Fig. 13. DDM compression ratio with tc = 0.819, ti = 50.

However, the time spent performing processing tasks differs notably from case to case when switching from desktop grade hardware to the Overos ARM Cortex A8.

The combined processing and computation times scale in a predictable way according to the execution times in Fig. 12. Approximately, processing times double when the FFT algorithm has to change to a higher power of two for efficiency. At 5 MS/s and for the coherent integration interval included in the figure, it happens after the 0.4019 and 0.819 ms boundaries. When doubling the incoherent averaging the processing time doubles as well: it is comparable to computing two separate DDMs with half the incoherent integration time and making the average of both. This limitation in the computation time implies that full DDMs could only be computed for selected targets.

C. Compression Ratio

Employing the same sources, sample compression ratios have been extracted to evaluate the advantages and disadvantages of the different strategies: formats, the need or not to discard noisy fragments, and so on. Fig. 13 presents ratios corresponding to conventional DDMs with different input signal powers (to reflect the degradation as the SNR becomes weaker) and coherent time = 0.819 ms. The trimmed DDMs have been set to a size of 100 × 20 pixels.

Obviously, discarding most of the DDM is the most effective strategy to decrease file size, but the compression algorithms offer quite straightforward conclusions as well: PNG compression does save some bytes, but its effectiveness is very limited and soon fades when the DDM becomes noisy. Lossless FAPEC offers a clearly better compression and the ratio resists much better the decreasing SNR. Additionally, in lossy mode, the precision loss due to discarding the eight least significant bits of each sample makes the stored observable less noisy. That noise reduction helps the compression algorithm, which performs better and reduces the compressed size to less than a half of the lossless version of the compressed DDM. This means that by combining all these techniques, a reduction up to 160x ratios is feasible (1 kB/DDM), which is very convenient due to the stringent downlink limitations.

VI. CONCLUSION

This study presents the GNSS-R instrument designed for the [3]CAT-2 mission, a six-unit CubeSat. The selected hardware has been described: a Gumstix Overo with a Tobi extension board as the payload computer, and an SDR for data acquisition. A software to capture and process raw data has been optimally designed for the hardware components, developed, and successfully tested.

The instrument is able to compute complete DDMs using different GNSS-R techniques for different frequencies and polarizations. Most parameters can be arbitrarily selected from ground: coherent and incoherent integration times, offset in frequency axis, and Doppler frequency and temporal resolutions. Also, in conventional GNSS-R multiple constellations can be used with the possibility to expand to other types of signals by means of a simple software upgrade.

Preliminary laboratory tests of the instrument with simulated and real GPS signals show promising results in conventional and interferometric modes. Field tests have shown good results with real BeiDou signals as well (not shown in this paper because they look like the GPS C/A code ones).

In terms of compression, FAPEC offers a good compression ratio even in noisy files using lossless mode, that is significantly improved if lossy compression is applied, with a negligible degradation of the SNR.

The only limitation of this software implementation is the computation time: 15 s for $T_i = 0.5$ ms and $N = 50$ incoherent averages, and 25 s for $T_i = 1$ ms and $N = 50$ incoherent averages. Total integration time ($T_i N$) can be longer than 50 ms, but it is deliberately limited to this value to minimize blurring of the DDMs. Retracking will be performed on ground. Although despite the Doppler frequency and delay drifts the DDM peak can be tracked and measured continuously, over the whole duration of the acquisition as illustrated in Fig. 7. The relatively long integration time implies that only selected full DDMs can be acquired over selected targets, computed and downloaded.

REFERENCES

[1] H. Carreno-Luengo *et al.*, "³CAT-2; An experimental nano-satellite for GNSS-R earth observation: Mission concept and analysis," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, to be published.

[2] H. Carreno-Luengo, A. Amzaga, D. Vidal, R. Oliv, J. F. Munoz, and A. Camps, "First polarimetric GNSS-R measurements from a stratospheric flight over boreal forests," *Remote Sens.*, vol. 7, no. 10, p. 13120, 2015. [Online]. Available: http://www.mdpi.com/2072-4292/7/10/13120

[3] H. Carreno-Luengo, A. Camps, I. Ramos-Perez, and A. Rius, "Experimental evaluation of GNSS-reflectometry altimetric precision using the P (Y) and C/A signals," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 5, pp. 1493–1500, May 2014.

[4] V. Zavorotny, S. Gleason, E. Cardellach, and A. Camps, "Tutorial on remote sensing using GNSS bistatic radar of opportunity," *IEEE Geosci. Remote Sens. Mag.*, vol. 2, no. 4, pp. 8–45, Dec. 2014.

[5] M. Unwin, S. Duncan, P. Jales, P. Blunt, and M. Brenchley, "Implementing GNSS reflectometry in space on the TechDemoSat-1 mission," in *Proc. 27th Int. Tech. Meeting Satellite Division Inst. Navigation*, 2014, pp. 1222–1235.

[6] C. S. Ruf *et al.*, "The CYGNSS nanosatellite constellation hurricane mission," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2012, pp. 214–216.

[7] A. Camps *et al.*, "PAU instrument aboard INTA microsat-1: A GNSS-R demonstration mission for sea state correction in L-band radiometry," in *Proc. IEEE Int. Geosc. Remote Sens. Symp.*, 2011, pp. 4126–4129.

[8] J. Marchan-Hernandez, I. Roures-Perez, X. Lluis-Bosch, A. Camps, N. R. Alvarez, and D. Albiol, "PAU-GNSS/R, a real time GPS-reflectometer for earth observation applications: Architecture insights and preliminary results," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2007, pp. 5113–5116.

[9] E. Valencia, A. Camps, J. Marchan-Hernandez, X. Bosch-Lluis, N. Rodriguez-Alvarez, and I. Ramos-Perez, "Advanced architectures for real time delay-doppler map GNSS-reflectometers: The GPS reflectometer instrument for PAU (griPAU)," *Adv. Space Res.*, vol. 46, no. 2, pp. 196–207, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0273117710000876

[10] Nanomind A712. Gomspace. [Online]. Available: http://gomspace.com/index.php?p=products-a712c, Accessed on Sep. 30, 2015.

[11] H. Carreno-Luengo, A. Camps, I. Ramos-Perez, and A. Rius, "Experimental evaluation of GNSS-Reflectometry altimetric precision using the P(Y) and C/A signals," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 5, pp. 1493–1500, May 2014.

[12] S. T. Lowe, T. Meehan, and L. Young, "Direct signal enhanced semi-codeless processing of GNSS surface-reflected signals," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 5, pp. 1469–1472, May 2014.

[13] Universitat Politècnica de Catalunya - Barcelona Tech. UPC NanoSat Lab. [Online]. Available: http://www.tsc.upc.edu/nanosatlab/home, accessed on Jul. 2, 2016.

[14] J. F. Muoz, *³Cat2 Software Validation Plan*, Doc. E-GEM_UPC_³Cat2_OBSWVTP_v2-0_301015, Ver. 2.0, Oct. 2015.

[15] M. F. van Bolhuis, *Environment levels ISILaunch08 Auxiliary payloads*, Doc. ISL.ISILaunch08.EL, ver. 1.0, Jan. 2016.

[16] A. Mazzoleni, *³Cat2 Radiation Analysis*, Doc. EGEM-DMS-TEC-MEM01-10, Ver. 1.0, Oct. 2014.

[17] Gumstix. Overo computer product page. [Online]. Available: https://store.gumstix.com/index.php/products/268/, accessed on Jul. 20, 2015.

[18] Gumstix. Tobi expansion board product page. [Online]. Available: https://store.gumstix.com/index.php/products/230/, accessed on Aug. 20, 2015.

[19] Gumstix. IPEX—eoPortal Directory—Satellite missions. [Online]. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/i/ipex, accessed on Sep. 23, 2015.

[20] Texas Instruments. DM3730. [Online]. Available: http://www.ti.com/product/dm3730/, accessed on Jul. 15, 2015.

[21] Ettus Research. USRP B210. [Online]. Available: http://www.ettus.com/product/details/UB210-KIT, accessed on Jul. 20, 2015.

[22] M. Frigo and S. G. Johnson. FFTW home page. [Online]. Available: http://www.fftw.org/, accessed on Jul. 20, 2015.

[23] Ettus Research. UHD project. [Online]. Available: http://code.ettus.com/redmine/ettus/projects/uhd/wiki, accessed on Jul. 20, 2015.

[24] FAPEC. DAPCOM Data Services. [Online]. Available: http://www.dapcom.es/fapec.html, accessed on Jul. 30, 2015.

[25] D. Abrahams. Boost python. [Online]. Available: http://www.boost.org/. Accessed on Jul. 30, 2015.

[26] Rohde & Schwarz. R&SSMU200A vector signal generator—overview—Rohde & Schwarz International. [Online]. Available: https://www.rohde-schwarz.com/en/product/smu200a-productstartpage_63493-7555.html, accessed on Sep. 10, 2015.