Implementation of a Low Earth Orbit Satellite Chirped Preamble Detector

Matthieu Magnant, Bertrand Le Gal, Guillaume Ferré, Florian Collard

Implementation of a Low Earth Orbit Satellite Chirped Preamble Detector

Matthieu MAGNANT[1], Bertrand LE GAL[2], Guillaume FERRÉ[1] and Florian COLLARD[3]
[1]Univ. Bordeaux, CNRS, Bordeaux INP, IMS, UMR 5218, 33400, Talence, France
[2]Univ. Rennes, CNRS, ENSSAT, IRISA/INRIA, UMR 6074, 22300, Lannion, France
[3]EUTELSAT, 92130, Issy-les-Moulineaux, France

***Abstract*—The Internet of Things (IoT) has undergone a significant transformation with the introduction of low-Earth orbit (LEO) satellites as gateways. This transformation fulfills a long-standing promise of IoT, which is to enable the connectivity of objects regardless of their geographical location on Earth. Various physical communication layers have exhibited the sensitivity required for such connections. In this paper, we present a solution designed for IoT network operators employing LEO satellites to detect various uplink IoT communication technologies that share a common preamble. Our proposed approach has been implemented on three ARM cores, specifically the Cortex-A9 and Cortex-A53, which are integrated into AMD Zynq and Zynq UltraScale+ based platforms designed to meet spatial constraints. Our experiments confirm that the proposed approach exhibits real-time capabilities, even when executed on these lower-end processor targets, consuming only approximately 10% of the CPU time. Experiments were performed on both synthetic data and real traffic recordings from Eutelsat's ELO2 payload hosted in the Loft Orbital's Yam-3 LEO nanosatellite, and showed promising results.**

***Index Terms*—IoT, LPWAN, satellite communications, Low-Earth Orbit, chirp signals, ARM, ELO2**

I. INTRODUCTION

THE Internet of Things (IoT) is experiencing a revolution thanks to the democratization of access to space. Numerous projects aim to connect an object to the Internet whatever its position on Earth through satellites [1]–[4]. These low-Earth orbit (LEO) satellites, used as gateways, often take the form of nanosatellites carrying payloads. Whatever the embedded payloads, these satellites are equipped with digital communication systems to send/receive data to/from the ground. From a data communication point of view, it is challenging to connect IoT in this way [5], [6]. The two major issues are related to:

1) the relative speed between the satellite and the Earth,
2) the satellite field of view (FoV) [7].

Indeed, terrestrial Low Power Wide Area Network (LPWAN) technologies, such as Semtech's LoRa [8] and LR-FHSS [9], or NB-IoT [10], are being deployed to support LEO satellite communication. These long-range, low-energy wireless networks exhibit remarkable link budgets, with some capable of accommodating path losses of up to 150 dB. For instance, LoRa technology in Europe offers extended-range modes with sensitivities as low as $S = -140$ dBm and an Effective Isotropic Radiated Power (EIRP) of up to $P_{EIRP} = 14$ dBm in the 868 MHz Industrial, Scientific and Medical (ISM) band. Adapting these technologies for satellite communication necessitates the design of receivers capable of demodulating their signals.

However, before performing signal demodulation, detecting the signals of interest is an essential preliminary step, often accomplished through a signal preamble. While the Doppler effect stemming from the satellite's velocity is independent of the communication technology used, the impact of the FoV is technology-dependent. Communications within unlicensed frequency bands, such as the ISM bands, are susceptible to interference due to the coexistence of multiple technologies sharing the same spectrum without coordination. Strategies to mitigate collisions have been proposed [11]–[14], but the fundamental step is the detection of signals.

In the literature, several effective preamble detection strategies for current LPWAN technologies used on the ground have been developed and evaluated in simulation [15]–[18]. The authors in [19] and [20] also proposed implementation and measurement results for their algorithms. However all these strategies are designed for terrestrial LPWA communications. Research on LPWAN adaptability to the specific context of LEO satellite communication has been conducted in [21] and [2]. However no simulation or implementation result is provided in terms of good detection and false alarm probability for the detection of uplink signals. To the best of our knowledge, algorithm in [1] is the closest related work. Authors in [1] design a new receiver aiming at detecting LoRa signals from a LEO satellite. A LoRaWAN network simulator is implemented and simulation results in terms of percentage of successful decoding are given. However, no real hardware implementation is conducted in [1].

In this paper, we introduce an algorithm for detecting heterogeneous chirped preambles in the context of uplink communications to LEO satellites. In this specific application, our detector is required to deliver robust performance under both low and high-connectivity conditions. Thus the proposed algorithm is evaluated in simulation in terms of good detection and false alarm probabilities. Furthermore, the proposed algorithm must exhibit low computational complexity to comply with space-constrained embedded systems. As such, we present an evaluation of the real-time performance of the proposed algorithm on programmable platforms suitable for space applications, with synthetic data but also with real

This paper was produced by the Univ. Bordeaux, CNRS, Bordeaux INP, IMS, UMR 5218, F-33400 Talence, France.

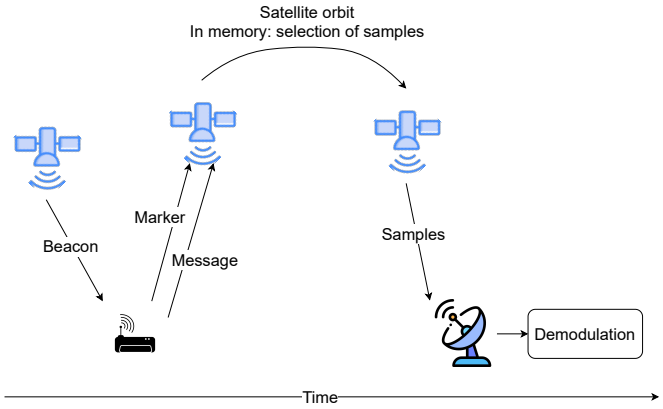

Fig. 1: Store and forward operating mode.

satellite data acquired by our industrial partner, Eutelsat.

The paper is organized as follows. In section II, the study context is presented. Then, the proposed detection algorithm is described in section III. Its performances evaluation in simulation is detailed in section IV. Then section V presents implementation results, in terms of execution time and energy consumption, of the proposed detection algorithm on a selected set of ARM cores that are currently used in nanosatellites. Finally, section VI details experimental results coming from real-world satellite signals acquired by our industrial partner, whereas section VII exposes the conclusion as well as the research perspectives.

II. Context

This research work is a collaborative project in partnership with the French satellite operator Eutelsat [22], aimed at enhancing their LEO IoT satellite constellation called ELO. Eutelsat proposes to provide long-distance, low-bit-rate communication services for IoT using its satellites in low Earth orbit. Eutelsat aims to cover several modulations used in IoT, thus offering multi-modulation connectivity services, regardless of the waveform used (such as LoRa CSS [23], LoRa DCSS [7], LR-FHSS [9], ex-Sigfox [24], Ternwaves Golden Modulation [25], etc.).

There are two possible operating modes for ELO satellites to achieve multi-modulation connectivity. The first one is the implementation of one or several on-board demodulators. The second one is a ”store and forward” mode. This involves recording the spectrum onboard the satellite, storing it and bringing it down when a ground station is in view. This second operating mode, which is of interest to us here, is represented in Figure 1. To achieve this, Eutelsat suggests using a unique marker transmitted by the ground device according to the beacon signaling, so that the receiver, located at the LEO satellite level, can identify this marker and start the spectrum recording.

Here, we propose using preambles with a chirp waveform as the marker (figure 2). This implies only a firmware update of devices already deployed in IoT networks, as the chirp waveform is already available in the deployed chips, including SX1261 and SX1262 for LoRa and LR-FHSS, as well as

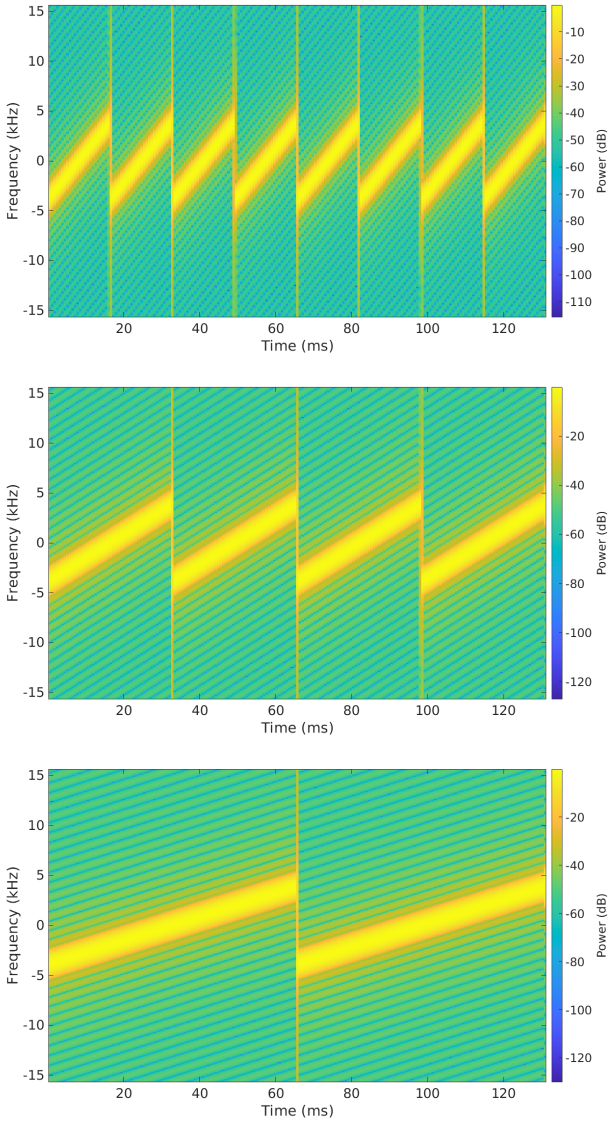

Fig. 2: Proposed markers: chirped preambles placed before standard IoT frames. The bandwidth is $B_p = 7.8$ kHz and preambles duration is $T_p = 131.3$ ms with $SF \in \{7, 8, 9\}$ and $N_p \in \{8, 4, 2\}$ chirps respectively.

S2-LPQTR and S2-LPCBQTR for Sigfox [24]. Therefore, in this article, we present an algorithm for detecting our preambles, which is intended to be implemented in Eutelsat's ELO satellites. The goal is for the ELO satellites to collect messages transmitted by ground devices by detecting our preambles, recording the spectral portion containing these messages, and then transmitting this spectral recording back to Earth. Message demodulation is then performed on Earth, depending on the waveform used (such as LoRa CSS, LoRa DCSS, LR-FHSS, ex-Sigfox, Ternwaves Golden Modulation, etc.). The idea is to shift the computational complexity of demodulation to the ground, keeping at the satellite level only the relatively low computational complexity of chirped preambles detection.

Available spectrum is divided into two sub-channels: one

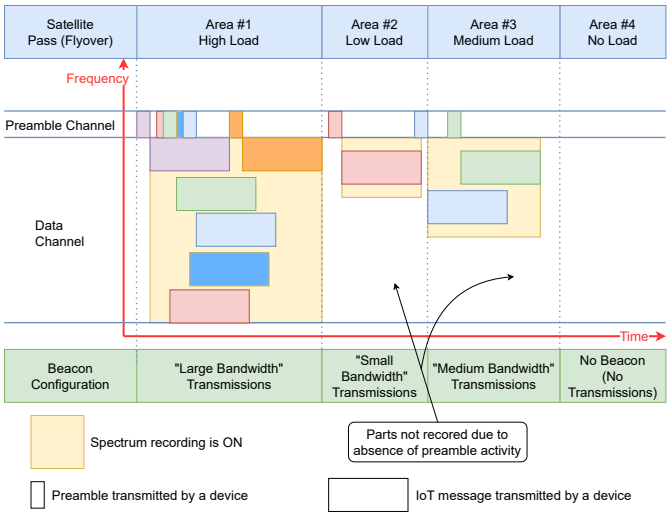

Fig. 3: Spectrum division and spectrum recording process.

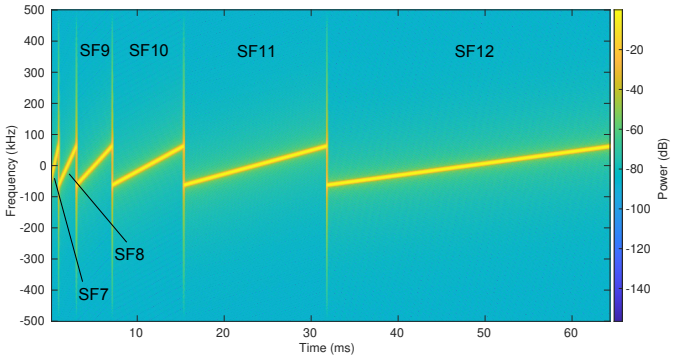

Fig. 4: Typical LoRa chirps represented in time-frequency domain with $SF \in \{7, ..., 12\}$ and a bandwidth $B = 125$ kHz.

preamble channel containing the detection information (i.e. our preambles) and one data channel containing the actual messages transmitted by objects. Furthermore, the bandwidth of the preamble channel is limited to a maximum of 10 kHz, and the maximum allowed duration of the preamble is 150 ms, for spectral congestion reasons. This division into two channels allows for not limiting the data rate of the messages by the maximum bandwidth of the preamble channel. This spectrum division is represented in figure 3. The preamble is a corner stone. Preamble presence triggers the spectrum recording based on Tx bandwidth (signaled by the beacon) and message duration. Preamble detector could be switched off in very high load and no load area. Generic preamble detector provides thus an efficient solution for an implementation of a selective and intelligent spectrum recording to drastically reduce the amount of spectral data recorded by the satellite while supporting generic waveforms. The following section provides a description of chirp-based communication systems and also the detection algorithm developed.

III. Chirp-based preambles detection algorithm

Let us start with an explanation of Chirp Spread Spectrum (CSS) modulation technique [23], [26], [27]. CSS modulation relies on sine waves whose frequency evolves linearly with time across a specified bandwidth, denoted as B. These particular waveforms are referred to as chirps. A raw chirp initiates at an initial frequency, denoted as f_i, and progresses linearly to a final frequency, f_f, throughout the symbol time, T, with the bandwidth B calculated as $B = |f_f - f_i|$. When f_i exceeds f_f, it is classified as a down chirp, whereas it is considered an up chirp when the reverse is true. It is noteworthy that the CSS waveform maintains a constant envelope, thereby mitigating transmitter energy consumption.

The transmission of binary information is initially subdivided into subsequences, each of length SF. A symbol comprises a set of SF consecutive bits. The total number of possible symbols is thus $M = 2^{SF}$. When the symbol equals to 0, the chirp obtained is called raw chirp. In LoRa, SF falls mainly within the range of $\{7, 8, 9, 10, 11, 12\}$, representing the spreading factor. The example below illustrates the bit-symbol association with $SF = 7$, a symbol being obtained simply by converting the binary subsequence in its decimal value:

$$\underbrace{(0110100)_2}_{\text{7-bit subsequence of binary information}} = \underbrace{(52)_{10}}_{\text{associated symbol}} \quad (1)$$

In conventional digital communication systems without spreading spectrum, the signal's bandwidth is directly proportional to the symbol rate, with the proportionality determined by the shaping filter, which commonly adopts a half-Nyquist profile. In the context of CSS, the signal's bandwidth is predetermined by B, and its relation to the symbol time T can be expressed as follows:

$$T = \frac{M}{B}. \quad (2)$$

This relation ensures phase continuity between successive modulated chirps. Consequently, as illustrated in figure 4, for a fixed bandwidth, it becomes evident that an increase in SF results in an extended symbol duration (specifically, augmenting SF by 1 effectively doubles the symbol duration T). Note that $\frac{1}{T}$ is equal to $\frac{B}{M}$. If M is large (greater than a factor of 100, which is the case here with $SF \geq 7 \longrightarrow M \geq 128$), $\frac{B}{M}$ is small compared to B. This is why we do not include the term $\frac{1}{T}$ in the bandwidth occupied by the chirp.

A. Preamble construction

The preambles of different SF are orthogonal as soon as received power difference is about 6 dB. For satellite communication, such a power loss (resp. gain) corresponds to a division (resp. multiplication) by $2\times$ of the communication range. Given that the satellite constellation orbits at an altitude of 550 km, such a power difference corresponds to a FoV of at least 2.8 million km^2 i.e. $5.2\times$ France area. Thus, it is necessary to develop an algorithm allowing the detection of non-orthogonal preambles, even if the SF are different.

Considering the project constraint on the preamble bandwidth $B_p = f_f - f_i$ (< 10 kHz), the latter is fixed to

TABLE I: Chirps duration with $B = 7.8$ kHz

SF	7	8	9	10	11	12
T (ms)	16.4	32.8	65.6	131.3	262.6	525.1

7.8 kHz, the minimal bandwidth supported by off-the-shelf LPWAN chips which respects the constraint [28]. Another project constraint is having a preamble duration T_p of less than 150 ms. According to (2), fixing the bandwidth and the SF values determines the chirps duration. Thus table I presents symbols duration with typical LoRa SF values. The last two symbol durations exceed the 150 ms constraint and thus the associated SF values are not suitable for our project.

In order to include some chirp repetition in the preambles and to enhance preamble detection, $SF \in \{7, 8, 9\}$ with respectively $N_p \in \{8, 4, 2\}$ raw up chirps in the preamble are selected. Consequently, the preamble duration T_p is the same for all our preambles:

$$\forall SF \in \{7, 8, 9\}, \quad T_p = N_p \times \frac{2^{SF}}{B_p} = 131.3 \text{ ms}. \quad (3)$$

Figure 2 represents the preambles thus created.

Let us remember that the CSS modulation consists in associating to each SF-uplet of bits a unique phase trajectory $\phi_k(t)$ of duration T among a set ω_S of M different trajectories. If $s_p(t)$ denotes the preamble complex envelope, we obtain:

$$s_p(t) = \sum_{m=1}^{N_p} \exp\left(j\phi_0(t - (m-1)T)\right) \mathbb{1}_{[(m-1)T, mT)}(t). \quad (4)$$

Without loss of generality, the signal instantaneous phase $\phi_0(t)$ is obtained by integrating the instantaneous frequency $f(t) = B_p\left(\frac{t}{T} - 0.5\right)$ and corresponds to the raw up chirp.

B. Preamble detection algorithm

In order to explain the preamble detection algorithm, the following equation provides the complex envelope of the waveform $y_r(t)$ received by the satellite:

$$y_r(t) = \sum_{i=1}^{N_R} \sqrt{P_i} x_i(t - t_{s,i}) \exp\left(j(2\pi(\Delta f_i + \frac{c_{d,i}t}{2})t + \varphi_{0,i})\right) + w(t), \quad (5)$$

with:

- $w(t)$ the complex additive white Gaussian noise whose variance is σ_w^2,
- N_R the number of preambles captured in the sampled bandwidth,
- P_i, $t_{s,i}$ Δf_i, $c_{d,i}$ and $\varphi_{0,i}$ are respectively the power, the starting time, the frequency offset, the Doppler rate and the initial phase of the i-th received signal,
- $x_i(t)$ the complex envelope of the i-th received signal.

The latter can be expressed as the concatenation of our custom preamble and a typical IoT packet:

$$x_i(t) = s_{p,i}(t)\mathbb{1}_{[0,T_p]}(t) + r_i(t - T_p)\mathbb{1}_{[T_p, T_p + T^i_{pkt})}(t). \quad (6)$$

Here $r_i(t)$ represents the packet complex envelope of the i-th received signal of duration T^i_{pkt}, coming from a IoT object on the ground using a typical LPWAN protocol. The other signal $s_{p,i}(t)$ represents the complex envelope of the preamble.

1) Preamble detection without interference

A first scenario without preamble collision is now considered. This preliminary approach will be extended further in this paper. Therefore, the received signal, sampled at $T'_s = \frac{1}{\alpha \times B_p}$, where α stands for the oversampling factor, is expressed as follows:

$$y_r(n) = \sum_{i=1}^{N_R} \sqrt{P_i} s_{p,i}(n - n_{s,i}) \exp\left(j(2\pi(\Delta f_i + \frac{c_{d,i}.nT'_s}{2})nT'_s + \varphi_{0,i})\right) + w(n), \quad (7)$$

with $n_{s,i} = \lfloor \frac{t_{s,i}}{T'_s} \rfloor = K_{s,i}\alpha M + \tau_{s,i}$. Here $\tau_{s,i}$ represents the i-th received signal time offset and $K_{s,i}$ the index of the T-long initial preamble sequence. Considering the substantial range of values Doppler rates can take for these communications, the operations are executed in an oversampled mode compared to the Nyquist standard ($\alpha > 1$).

The algorithm thus performs the following steps, for each SF value:

1) *Dechirping* of the entire received signal $y_r(n)$ for each SF value. In the context of this paper, the term *dechirping* will refer to the operation involving the multiplication of a T-long segment of the CSS signal by the conjugate of the raw up chirp, $e^{-j\phi_0(t)}$ (also called raw down chirp).
2) FFT on the dechirping over $\frac{T}{T'_s}$ samples.
3) Accumulation of FFT bin energy with computation of the following function:

$$T(k, p) = \sum_{j=p}^{p+N_p-1} \left| \frac{Y(k, j)}{\sigma_w} \right|^2. \quad (8)$$

Here $Y(k, j)$ ($k \in [\![0, M-1]\!]$) corresponds to the k-th FFT bin of the *dechirping* of the j-th slice of $y_r(n)$ of duration T. Furthermore, $p \in \{1, \ldots, N_B\}$ represents the index of the T-long sequence being processed and $N_B \times T$ represents the buffer duration, i.e. the number of recorded samples is $N_B \times M$.

Figure 5 illustrates the outcomes derived for the function $T(k, p)$ concerning the detection process with parameters set to $SF = 7$ and $N_p = 8$. Figure 5 illustrates that the detection associated with a specific SF manifests a distinct *pattern*. The detection algorithm involves performing a particular intercorrelation. Given the signal's phase continuity over the entire preamble duration, we can afford not to perform a sample-by-sample intercorrelation, but rather every symbol duration, which is αM. Consequently, we observe a classic triangle autocorrelation pattern. In a Nyquist scenario (i.e. $\alpha = 1$ where the sampling frequency is the signal's bandwidth), discovering $2N_p - 1$ peaks in the *pattern* is anticipated, considering that the interval between these peaks is M. Moreover, the maximum energy is attained for $p = K_{s,i}$ because all the FFTs include the preamble. However, within an oversampling framework,

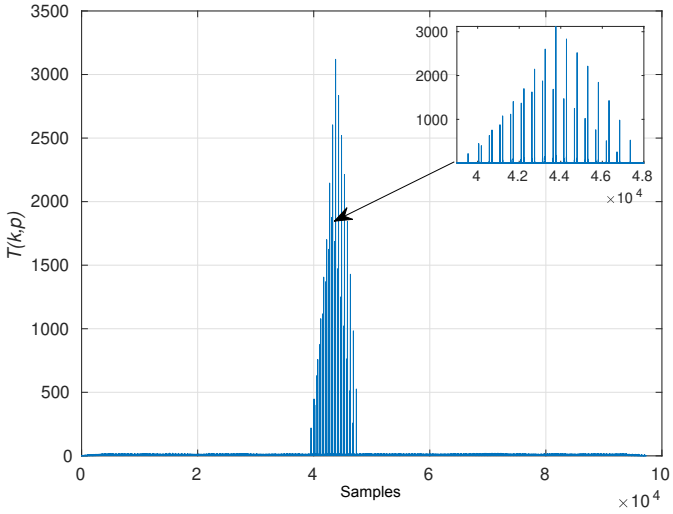

Fig. 5: $T(k,p)$ without interference, $SF = 7$, $B_p = 7.8$ kHz and $N_p = 8$.

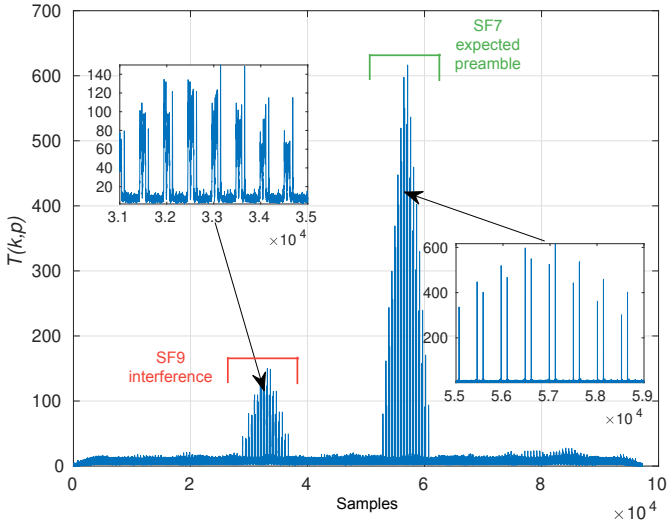

Fig. 6: Shape of $T(k,p)$ for $SF = 7$, $B_p = 7.8$ kHz and $N_p = 8$ with the presence of an $SF = 9$ and $N_p = 2$ preamble.

each FFT reveals two energy peaks (located at a distance from M) due to dechirping. This phenomenon persists even in the case of perfect synchronization [1]. Thus, in order to identify the presence of a valid preamble, this *pattern* should be identified.

To achieve this objective, starting from the FFT bin energy accumulation $T(k,p)$:

1) The function $M(p) = \max_k(T(k,p))$, $p \in \{1,\dots,N_B\}$ is computed to extract the peak of each T-long sequence,
2) The algorithm then scans all energy peaks within $M(p)$ that surpass a threshold Th while maintaining a minimum separation of N_p to ensure the identification of two distinct patterns. In essence, the algorithm endeavors to discover $\hat{K}_{s,i}$, an estimation of $K_{s,i}$ where $\hat{K}_{s,i} = \operatorname{argmax}_p(M(p))$. It should be noted that the calculation of the threshold Th involves conducting a binary hypothesis test and utilizing $T(k,p)$ as the test variable under the assumption of noise,
3) The algorithm confirms whether the energy peaks at the sequence $K_{s,i} - 1$ and $K_{s,i} + 1$ surpass the threshold Th and are positioned at the identical frequency as the energy peak of the sequence $K_{s,i}$.

Thus, the following subsection introduces a necessary evolution of the detection method aimed at incorporating the consideration of interferences from frames having the same SF value as for frames with different ones.

2) Preamble detection when interference occur

To understand the impact of inter-SF interference effects on the initial proposed approach, we depict in figure 6 the profile of $T(k,p)$ when conducting a detection for $SF = 7$ and $Np = 8$. While observing the anticipated *pattern*, an additional *pattern* emerges, corresponding in this instance to the presence of an $SF = 9$ and $Np = 2$ preamble. When performing the SFx algorithm on SFy with x different from y, the energy of the chirps in the preamble spreads across multiple frequencies, leading to both a different pattern and a reduction in the maximum observable energy. This is an expected phenomenon because we are no longer performing autocorrelation but intercorrelation. On closer examination of the latter, it becomes apparent that this $SF = 9$ preamble could potentially trigger a false detection if the previous algorithm is applied without adjustment. To address this limitation and given the *pattern*'s structural variations (notably, the presence of numerous energy peaks), we modify the detection process with a constraint based on peak-to-average power ratio (PAPR). This constraint is implemented as follows:

$$PAPR = \frac{\max_k(T(k,p))}{\operatorname{mean}_k(T(k,p))}, \tag{9}$$

where $p \in \{K_{s,i} - 1, K_{s,i}, K_{s,i} + 1\}$. If the $PAPR$ surpasses a predetermined threshold, the preamble is deemed valid; otherwise, it is rejected. This holds true regardless of the SF affected by the interference. It should be noted that the computing of this threshold is more complex than the previous one. Its value is determined by numerical simulation.

When confronted with multiple preambles of the same SF causing interference, a specialized procedure is necessary. Our proposed approach involves employing the received power of preambles to execute an iterative process akin to the Successive Interference Cancellation (SIC) algorithm [15], [29]. The sub-figure presented on the left side of figure 7 portrays the pattern of the function $T(k,p)$ in an instance where multiple preambles of the same SF are in collision (here $SF = 9$ and $Np = 2$).

As it currently stands, our approach can only detect the preamble exhibiting the strongest received power. To overcome this limitation, the contribution of each valid preamble is sequentially suppressed from the function $T(k,p)$: the energy peaks corresponding to the detected preambles are reset to zero. Consequently, through this iterative process, multiple

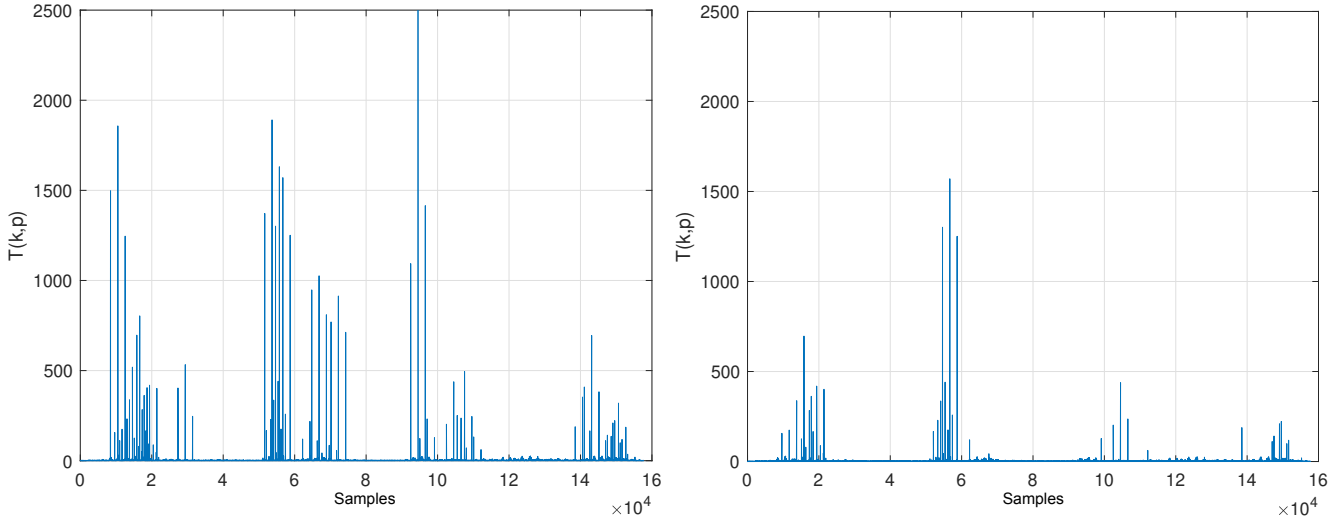

Fig. 7: Shape of $T(k,p)$, iterative process $SF = 9$, $B = 7.8$ kHz and $N_p = 2$ - (left) Initial state - (right) After 1 iteration.

preambles in collision can be detected. In the current algorithm version, a maximum number of iterations N_{itr} is fixed.

Figure 7 on the right showcases the example after a single suppression iteration, where the detected preambles have been eliminated, thereby facilitating the resolution of preambles in collision. This figure is provided for illustrative purposes. It allows for the observation of the suppression in the spectrum of the detected preamble after one iteration, enabling the discovery of patterns that were previously invisible in the first iteration.

Figure 8 summarizes the algorithm thus developed. It represents the different steps to identify the *patterns* and perform a SIC-like procedure to deal with same SF and inter-SF interference. The figure is also accompanied by the pseudo-code Algorithm 1.

Algorithm 1 Preamble Detection Algorithm

Require: Received signal
Ensure: Lists of preambles found
  for $SF = 7$ to $SF = 9$ **do**
    dechirper $\leftarrow$ conjugate(bank_of_chirps(SF));
    for $i = 0$ to number_of_signal_chunks **do**
      dechirped$_i$ $\leftarrow$ sig_chunk$_i$ $\times$ dechirper;
      $modSqr_i(k) \leftarrow |\text{FFT}(\text{dechirped}_i)|^2$; ▷ k = FFT bin
    end for
    for $i = 0$ to number_of_signal_chunks **do**
      $T(k,i) \leftarrow \sum_{p=i}^{i+N_p} modSqr_p(k)$;
      $M(i) = \max_k(T(k,i))$;
    end for
    PreamblesFound $\leftarrow$ Values of M which $> Th$;
    SIC procedure on $T(k,i)$ values;
  end for

In order to evaluate the relevance of the proposed detection algorithm, simulations on synthetic data are first performed. These experiments and their results are described in the following section.

IV. Simulation Results

The purpose of this section is to assess the detection performances of our algorithm through Monte Carlo simulations under varying system loads, corresponding to different levels of communication interference. The system load is defined as $Load = \frac{N_R T_p}{T_B}$, where N_R denotes the number of packets to be detected, and $T_B = N_B T_p$ represents the duration of the processed recording. Essentially, the system load quantifies the number of packets that need to be detected.

For simulation purposes, the number of packets generated

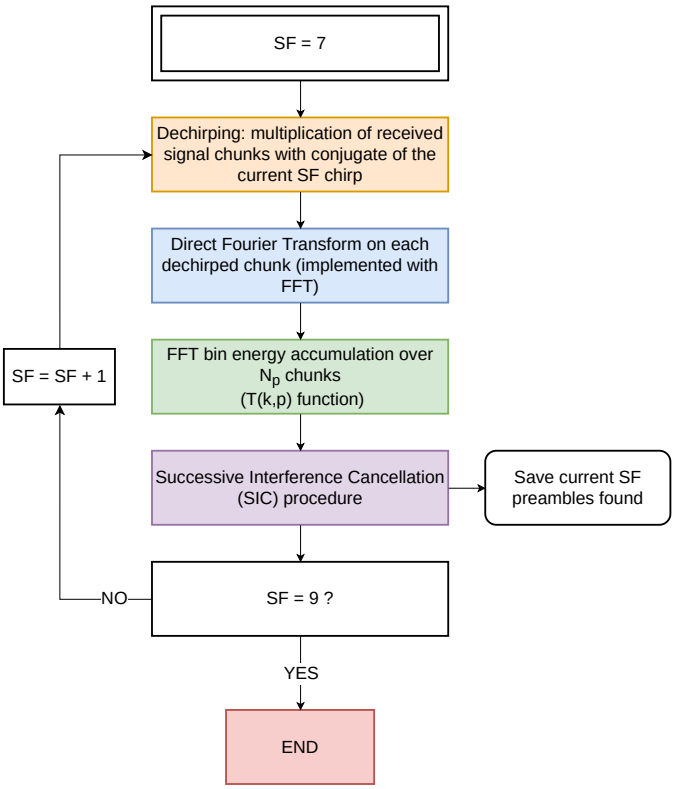

Fig. 8: Diagram of the preamble detection algorithm.

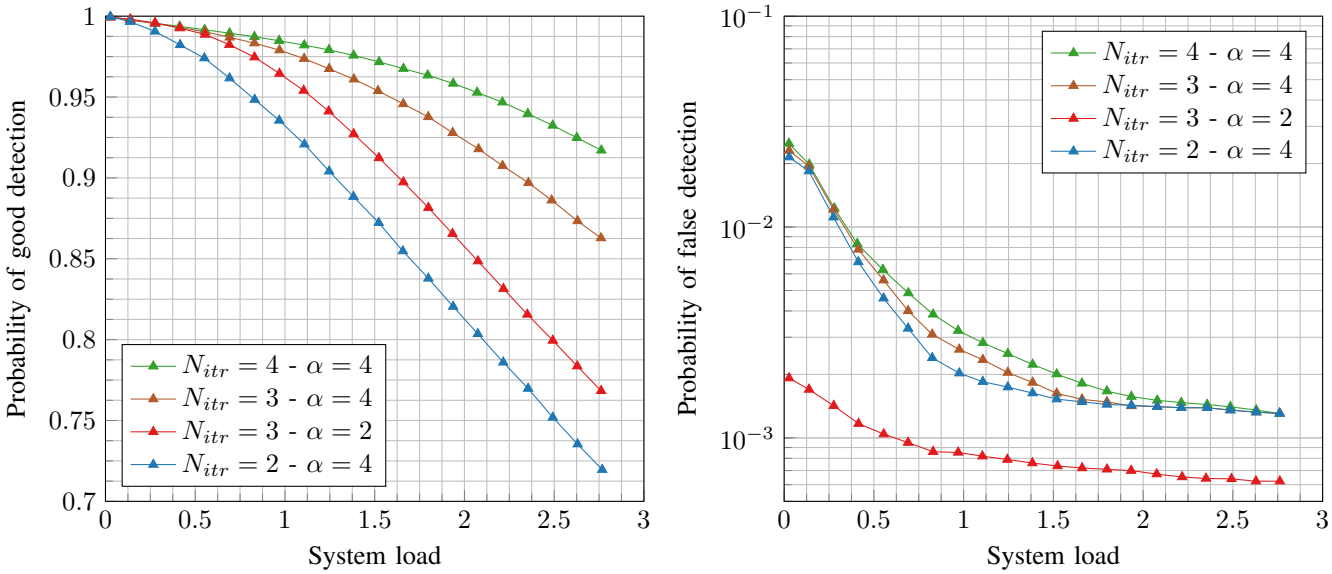

Fig. 9: Good and false detection probabilities for the proposed algorithm depending on system load, α and N_{itr} values ($SNR \in [-5; +5]$ dB).

Carrier f_c (MHz)	868
Preamble bandwidth B_p (kHz)	7.8
CFO max Δf_{max} (kHz)	20
DR max DR_{max} (Hz/s)	300
Power spread PS (dB)	5
Mean SNR (dB)	0
Oversampling factor α	$\{2, 4\}$
Probability of false alarm P_{fa}	10^{-3}
Buffer duration T_B (s)	5
Number of iterations N_{itr}	$\{2, 3, 4\}$
Monte Carlo number	10000

TABLE II: Simulation parameters

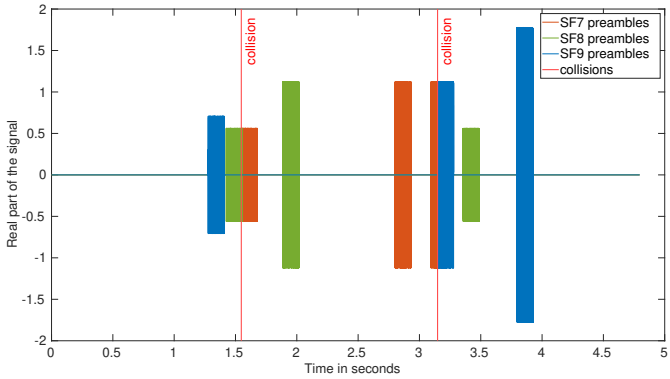

Fig. 10: Example of a simulation input signal. The noise is not represented for reasons of legibility.

in a buffer follows a Poisson distribution. The preamble type ($SF \in \{7, 8, 9\}$), the preamble positions in the reception buffer, and, consequently, the number of packets in collision are also generated randomly. The power of received preambles is uniformly distributed across a set of values consistent with the link budgets specific to our communication use case. Taking into account the satellite's Field of View, the received power is distributed over a range PS of ± 5 dB, with a mean Signal-to-Noise Ratio (SNR) of 0 dB, because the LEO satellite is approximately at 550 km. Figure 10 represents a typical input signal used in simulation. The input signals last for 5 seconds. Preambles are randomly generated within these 5 seconds. Here, figure 10 corresponds to a load of 0.24, i.e., approximately 9 preambles in a 5-second buffer. The carrier frequency offset (CFO) and the Doppler rate (DR) are respectively uniformly distributed in $[-\Delta f_{max}, \Delta f_{max}]$ and $[-DR_{max}, DR_{max}]$. The simulation parameters employed are summarized in Table II.

We aim to assess the influence of the number of iterations N_{itr} and the oversampling factor α on the preamble detection. Figure 9 illustrates the probabilities of accurate and false detection concerning the system load. On the left part, we observe that as the number of iterations increases, the probability of correct detection also increases, with $\alpha = 4$ fixed. With a fixed number of iterations, increasing α leads to better performance because it aligns the preambles more effectively, allowing us to collect more energy through our algorithm. On the right part, as the number of iterations increases, it becomes more likely that false detections will occur at low load levels. This is due to both the removal of the preamble overlaid from the previous iteration and the increased probability of finding a false preamble with more iterations (as more time is spent searching for it when it does not exist). However, we observe that this effect diminishes and becomes negligible at higher loads (typically for a system load of 2.6). At this 2.6 load (which corresponds to an average of 100 preambles received within $T_B = 5$ s), the probability of accurate detection is 0.92 (resp. 0.88) for $N_{itr} = 4$ (resp. $N_{itr} = 3$).

Additionally, the curves in figure 11 demonstrate that the detection performances are inversely proportional to the increase in SF. This outcome can be attributed to the number of chirps N_p in the preamble, which decreases by half when

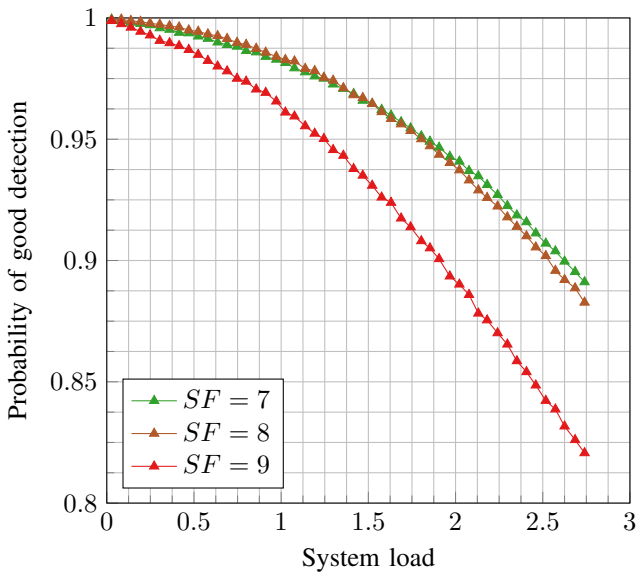

Fig. 11: Probabilities of good detection for the proposed algorithm depending on system load and SF value when $\alpha = 4$ and $N_{itr} = 3$ ($SNR \in [-5; +5]$ dB).

SF increases by one to maintain the same preamble duration. Indeed, the smaller the SF, the better the performance, because for intercorrelation, we have a signal with more chirps available.

Several studies have attempted to evaluate the performance of chirp signals for Direct-to-Satellite IoT. In [30], an analysis of the acquisition of symmetry chirp signals is conducted in terms of missing probability. However, the simulations are not conducted based on the total system load, i.e., the number of collided packets to be detected. In [31] and [32], an analysis of the detection performance of symmetric, asymmetric, and LR-FHSS chirp signals is performed. However, in [31], the same SF is used by all terminals. Furthermore, perfect time and frequency synchronization are assumed, and the number of collided frames is low. Then in [32], the results are not provided as a function of the system load, whereas this metric is of interest in our case study.

To the best of our knowledge, [1] is the closest related work. The authors in [1] design a new receiver aiming at detecting LoRa signals from a LEO satellite. Therefore, the comparison of our results with the literature should be made with the results presented in [1]. For a system load of 1, i.e., approximately 38 transmitting users in 5 seconds, our detection probability is 0.97 compared to 0.94 for [1]. For a load of 2, i.e., approximately 76 transmitting users in 5 seconds, our detection probability is 0.92 compared to 0.83 for [1]. We observe that our algorithm achieves better performance in terms of detection probability regardless of the system load. No results are provided in terms of false detection probability in [1].

Based on these simulation results, a Real-Time implementation of the detection algorithm is developed and evaluated on real data sets to validate simulation results. This work is reported and discussed in the following sections V and VI.

V. Real-Time implementation and evaluation

To demonstrate the applicability of the proposed approach in the nanosatellite context, a windowed adaptation of the previously described detection algorithm is developed in C/C++ 14. In contrast to the approach delineated in section III, the implemented algorithm operates continuously over a window spanning a duration of $4 \times T_p$ to ensure real-time data processing and minimize latency. This window is divided into four zones encompassing samples received at $N-3$, $N-2$, $N-1$, and N. As depicted in figure 12, the search zone examined at time N encompasses sub-windows $N-3$, $N-2$, and $N-1$. In this system, new samples derived from sub-window N are crucial for computing the energy, related to the $T(k,p)$ function, in sub-window $N-1$. This process, conducted across the four sub-windows and employing a sample aging mechanism, effectively suppresses side effects and prevents redundancy in preamble detection.

The algorithm implemented on the hardware is essentially a rewrite of the initial algorithm, with numerous improvements and optimizations made to leverage the functions available in the hardware. However, these optimizations do not alter the behavior of the algorithm. Thus, the maximum theoretical detection performances remain the same between the initial version and the implemented one, as it is the same algorithm.

To minimize detector execution time, subsequently reducing energy consumption and thermal dissipation, we leverage the Single Instruction, Multiple Data (SIMD) features present in contemporary ARM processors, specifically tailored for space utilization [33]. Current ARM cores enable us to process multiple data in parallel using their 128b NEON vector units [34], [35]. This allows, thanks to intrinsic NEON, to enhance regular signal processing algorithm execution times by almost a factor 4 when these algorithms perform floating-point calculations and up to $16\times$ when 8 bit data are processed [36]. However, to take advantage of this feature, it is necessary to:

- describe appropriately the treatments to achieve so that the compiler automatically performs the parallelization,

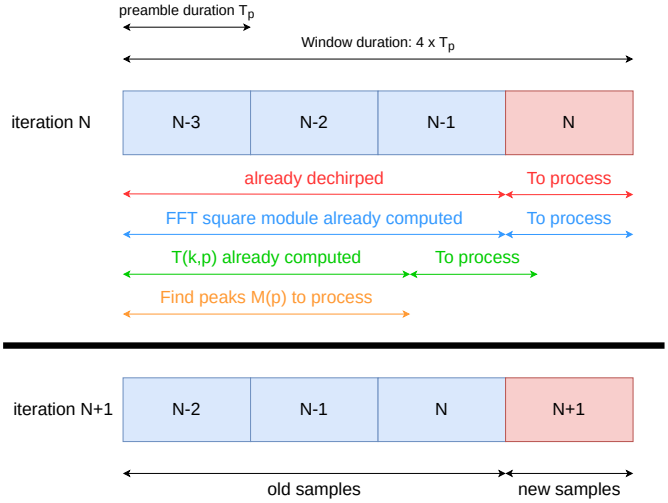

Fig. 12: Sliding window operation.

- describe manually the computing parallelization with intrinsic NEON.

Whatever the method, it is often necessary to reformulate the algorithms as well as the data structures to take full advantage of these features.

The explicit use of intrinsic NEON allows us manually optimizing time-consuming calculation kernels within the algorithm presented in section III. First of all, we study the open source FFT solutions available in the literature such as FFTW [37] because this very classic operation in signal processing has already been studied many times. Among all the ARM core compatible open source solutions, *pfft* library provides results 10% better than other solutions in our application context. Then, we manually optimize other treatments including:

- *Dechirping* process, which is a multiplication between two complex vectors,
- Complex vector module calculus,
- Preamble energy accumulation,
- Argmax research of a complex vector.

Source codes optimized for ARM based platforms are available as open source[1].

In order to verify real-time capabilities of our detector, we target 3 ARM cores with different performances and capabilities, all of them being able to be integrated in nanosatellites:

- $\mathcal{P}_1$ - An ARM Cortex-A72 64 bits (Raspberry Pi 4), whose clock frequency is set to 1.5GHz. This platform is equivalent to the hardened and radiation tolerant ARM core LS1046 developed by Teledyne e2v for space use.
- $\mathcal{P}_2$ - An ARM Cortex-A9 32 bits at 650MHz from a Xilinx ZedBoard development board (SoC Zynq 7020).
- $\mathcal{P}_3$ - An ARM Cortex-A53 64 bits at 2.3GHz from a Xilinx KRIA-KV260 developement board (SoC Zynq Ultrascale+).

The detectors are implemented on different platforms using specific compilers: GCC 12 for platform $\mathcal{P}_1$, arm-none-eabi-g++ for platform $\mathcal{P}_2$, and aarch64-none-elf-g++ for platform $\mathcal{P}_3$.

T functionality of the C++-based detector is first validated on platform $\mathcal{P}_2$ utilizing widely adopted Software Defined Radio (SDR) modules such as ETTUS b205 and HackRF, as illustrated in Figure 13. The laboratory test bench employes the Analog Devices RF front-end board AD-FMCOMMS3-EBZ, housing the AD9361 transceiver, in tandem with platform $\mathcal{P}_2$. This transceiver mirrors the RF front-end used in Eutelsat's nanosatellite. The experiments are conducted using optimized antennas tailored for the ISM band. Therefore, this laboratory test bench is a reproduction on the ground of the available hardware in Eutelsat's nanosatellite.

In Matlab, a script is written to generate traffics of preambles for transmission. These traffics are transmitted using the HackRF as a transmitter. Then, our program processes in real-time samples received by the RF front-end using the algorithm proposed in this paper (section III). Execution times are gauged on synthetic data simulating signal acquisitions over 60 s with preamble collisions. The table III details the

[1]Software codes developed as part of this study will be publicly available on GitHub once the work has been accepted for publication.

Fig. 13: Laboratory test bench: HackRF SDR sends a traffic of preambles - AD9361 RF front-end receives an RF signal and downgrades the signal to baseband - Our program runs on ARM Cortex-A9 (Xilinx Zynq 7020 SoC) and processes received signal in real-time to detect preambles.

Load	Plateform	tmin	tmax	tavg	tmed
Low 0.2	$\mathcal{P}_1$	2.1	4.4	2.4	2.4
	$\mathcal{P}_2$	12.3	14.2	13.6	13.9
	$\mathcal{P}_3$	3.8	4.7	4.2	4.6
High 2.2	$\mathcal{P}_1$	2.1	5.2	2.4	2.4
	$\mathcal{P}_2$	12.4	14.3	13.5	13.5
	$\mathcal{P}_3$	3.9	4.7	4.2	4.2

TABLE III: Window processing times on the 3 platforms (in ms)

minimum, maximum, average, and median execution times for the three platforms when $\alpha = 4$, $N_{itr} = 3$, and $SF \in \{7, 8, 9\}$. Two distinct load scenarios are assessed: $Load = 0.2$ (low) and $Load = 2.2$ (high). The execution times of the detectors fluctuate based on the number of preambles within the sliding window. Consequently, it is pertinent to prioritize the maximum execution times for consideration.

In our experiments, considering that $B = 7.8$ kHz, one iteration of the sliding window represents the arrival of $\alpha \times N_p \times 2^{SF}$ new IQ samples. The real-time constraint is therefore $B \times \alpha \times N_p \times 2^{SF} = 131$ ms. Results in table III show that execution times of the three platforms are much lower than the acceptable threshold of 131 ms. Thus on the ARM A72 platform ($\mathcal{P}_1$), the execution time in worst-case is 5.2 ms and then only represents 4% of the resources available. On the Zynq targets, results fluctuate according to the complexity of the ARM cores. On platform $\mathcal{P}_2$, the worst-case max execution time reaches 14.3 ms whereas on platform $\mathcal{P}_3$ the latter is 4.7 ms. These results demonstrate the interest of the proposed algorithm for space use and validate its low computation complexity.

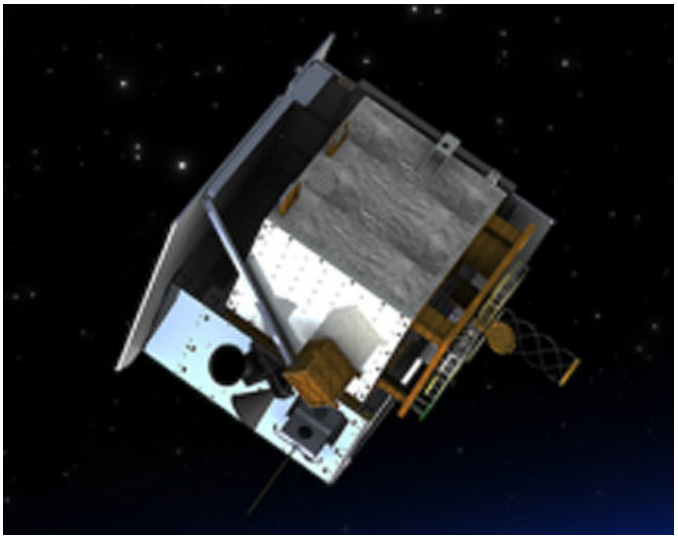

Fig. 14: Loft Orbital's Yam-3 LEO nanosatellite hosting Eutelsat's ELO2 payload (picture taken from [38]).

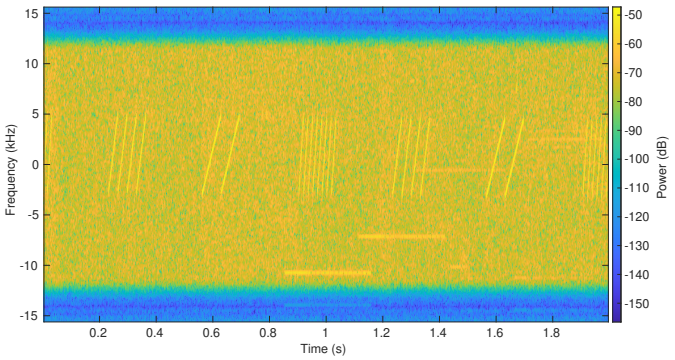

Fig. 15: Two seconds spectrogram of the received signal recorded by ELO2 nanosatellite while passing above Eutelsat's transmitter in Brazil during our transmission. Our preambles are well received.

VI. Real signal recordings provided by Eutelsat

Thanks to our industrial partner, the French satellite operator Eutelsat, it is possible to carry on the evaluation of our algorithm on real recorded data from their ELO2 platform. ELO2 is a payload hosted in the Loft Orbital's Yam-3 LEO nanosatellite, represented in figure 14.

The signal transmitted from the ground to ELO2 is intentionally simple. Only one transmitter is used. The transmitted signal consists of a sequence of preambles spaced 200 ms apart: an SF7 preamble, then 200 ms later an SF8 preamble, then 200 ms later an SF9 preamble, and so on. The purpose of this simple transmission scenario is to observe if our algorithm works during a real transmission. While simulation allows us to estimate the quality of the communication link using models of physical phenomena (Doppler effect, CFO, STO, etc.), it is only a model, not reality. That is why we are conducting this real transmission to a satellite.

The signal is transmitted with an $EIRP$ (Effective Isotropic Radiated Power) of 24dBm. This transmission power is relatively strong compared to the maximum of 14dBm in the 868MHz band in Europe, for example. We intentionally chose to transmit at high power to increase our chances of ensuring correct transmission. The transmitter, located in Brazil, operates in the ISM band at the carrier frequency of $f_c = 902.3$MHz. Therefore, the transmission takes place in a free and heavily congested frequency band. The maximum elevation of the satellite during the transmission is 70 degrees. Apart from the transmission power, all other parameters of the transmission are representative of the expected use case in our IoT context of devices communicating in free bands to low Earth orbit satellites.

Figure 15 represents a 2-second slice of the almost 3-minute long spectrum recording performed by ELO2 during our transmission. This recording is made on a 896 kHz bandwidth around the carrier frequency $f_c = 902.3$ MHz. We downsample the signal through filtering and decimation to reach our working frequency in oversampling mode with a factor of $\alpha = 4$. Thanks to a relatively high transmission power, we can observe our transmitted preambles quite well. They are identifiable by their chirped waveform, the repetition of the number of chirps, the swept bandwidth of 7.8 kHz, and their duration of 131 ms. Figure 16 represents the function $T(k, p)$ for preamble detection with $SF = 7$ on the same signal slice as figure 15. We clearly observe the appearance of our patterns, which are well distinguishable. Note that here we have reverted to the Nyquist rate $\alpha = 1$ to apply the algorithm to the real signal, hence the appearance of $2 \times N_p - 1 = 15$ peaks as anticipated. The same $T(k, p)$ shapes are obtained for $SF = 8$ and $SF = 9$.

Here, Figure 15 represents a 2-second segment of the signal, taken from the middle of the 3-minute recording. However, if we look at the very beginning or the end of the recording, we observe that our preambles drift within the observed band until they exit, when the satellite's elevation angle becomes too low. This is due to a Doppler effect that becomes significant when the transmitter is very close to the Field of View (FoV) boundary, typically at the beginning and end of our recording. This is not a problem as the objective of a LEO constellation is to cover geographical areas using multiple satellites, allowing ground devices to always be within the FoV of a satellite away from the FoV boundaries.

If we consider the portion of the recording where the preambles have not exited the observed band, we then detect 100% of the transmitted preambles, with no false positives. Therefore, our algorithm enables the detection of our preambles transmitted from the ground to a low Earth orbit satellite in a crowded ISM band, interfered with by other communication systems. Further work will focus on transmitting new real signals, with lower transmission power and with collisions between our preambles (for example, using multiple ground transmitters or using a single transmitter to simulate collided preamble traffic).

VII. Conclusion

In this paper was presented a solution for detecting uplink IoT communications from terminals on the ground to LEO

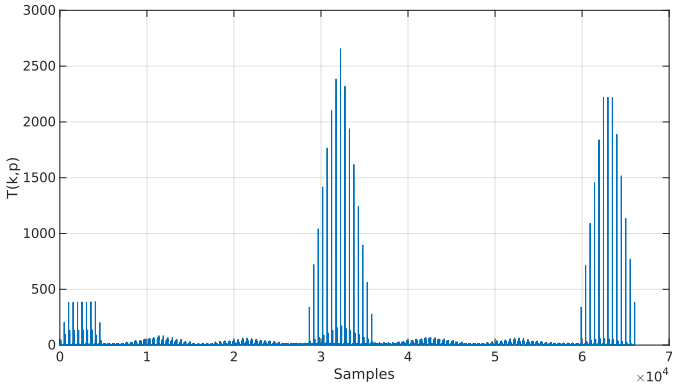

Fig. 16: $T(k,p)$ function for $SF = 7$ of the real signal on the same slice of signal as figure 15.

satellites. A unique IoT packet preamble based on chirps was designed to be transmitted by IoT objects. A corresponding algorithm was developed to detect these preambles from the satellite, whatever the LPWAN protocol used to transmit the IoT packet associated to the detected preamble. Detection performances were estimated through a Matlab simulation of such a satellite communication system (noisy communication channel, significant Doppler rates, typical received power and link budgets, etc.). This algorithm was implemented on three ARM cores integrated in AMD-Xilinx Zynq and Zynq Ultrascale+ platforms which are similar to the hardware available in our industrial partner nanosatellite, Eutelsat's ELO2 payload. These implementations exhibited the real-time capabilities of the proposed algorithm on the given hardware, with approximately 10% of CPU time used. Experiments continued with the utilization of real data transmission from Eutelsat's transmitter on the ground to their ELO2 payload in the Loft Orbital's Yam-3 LEO nanosatellite in orbit. Promising results were obtained, with 100% of transmitted preambles detected by our algorithm. These first results illustrated the proper operation of the detector when dealing with real transmissions between the ground and a satellite.

Future work will focus on Earth-satellite transmissions of more realistic traffic scenarios with more and more preamble collisions. FPGA accelerators and fixed-point computing will be developed to enhance detector performances in terms of execution time, latency and resource utilization. Future main objective is to deploy the detector on ELO2 to achieve both on-board detection and demodulation.

REFERENCES

[1] G. Colavolpe, T. Foggi, M. Ricciulli, Y. Zanettini, and J.-P. Mediano-Alameda, "Reception of LoRa Signals From LEO Satellites," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 3587–3602, 2019.

[2] T. Wu, D. Qu, and G. Zhang, "Research on LoRa Adaptability in the LEO Satellites Internet of Things," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2019, pp. 131–135.

[3] A. K. Dwivedi, S. Chaudhari, N. Varshney, and P. K. Varshney, "Performance Analysis of LEO Satellite-Based IoT Networks in the Presence of Interference," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[4] G. M. Capez, S. Henn, J. A. Fraire, and R. Garello, "Sparse Satellite Constellation Design for Global and Regional Direct-to-Satellite IoT Services," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 5, pp. 3786–3801, 2022.

[5] I. Ali, N. Al-Dhahir, and J. Hershey, "Doppler characterization for LEO satellites," *IEEE Transactions on Communications*, vol. 46, no. 3, pp. 309–313, 1998.

[6] Z. Qu, G. Zhang, H. Cao, and J. Xie, "LEO Satellite Constellation for Internet of Things," *IEEE Access*, vol. 5, pp. 18 391–18 401, 2017.

[7] M. Ben Temim, G. Ferré, and R. Tajan, "A New LoRa-like Transceiver Suited for LEO Satellite Communications," *Sensors*, vol. 22, 2022.

[8] Semtech, "Semtech LoRa technology overview," 2023. [Online]. Available: https://www.semtech.com/lora

[9] T. Lu, "LoRaWAN Protocol Expands Network Capacity with New Long Range – Frequency Hopping Spread Spectrum Technology," 2020. [Online]. Available: https://blog.semtech.com/lorawan-protocol-expands-network-capacity-with-new-long-range-/frequency-hopping-spread-spectrum-technology

[10] 3GPP, "Standardization of NB-IOT completed," 2016. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/nb-iot-complete

[11] J. Luo, A. Ito, A. Sasaki, M. Hasegawa, Y. Nagao, Y. Hiramatsu, K. Torii, S. Ashibe, and T. Aoki, "A Study on Adjacent Interference of LoRa," in *Proceedings of CANDARW*, 2020.

[12] G. Ferré, "Collision and packet loss analysis in a LoRaWAN network," in *Proceedings of EUSIPCO*, 2017.

[13] A. A. Tesfay, E. P. Simon, G. Ferré, and L. Clavier, "Serial Interference Cancellation for Improving uplink in LoRa-like Networks," in *Proceedings of PIMRC*, 2020.

[14] A. N. de São José, N. Chopinet, E. P. Simon, A. Boé, T. Vantroys, C. Gransart, and V. Deniau, "A Comparative Analysis of LoRa and LoRaWAN in the Presence of Jammers and Transient Interference," in *Proceedings of EMC Europe*, 2022.

[15] M. Xhonneux, J. Tapparel, P. Scheepers, O. Afisiadis, A. Balatsoukas-Stimming, D. Bol, J. Louveaux, and A. Burg, "A two-user successive interference cancellation lora receiver with soft-decoding," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 948–953.

[16] J.-M. Kang, D.-W. Lim, and K.-M. Kang, "On the lora modulation for iot: Optimal preamble detection and its performance analysis," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 4973–4986, 2022.

[17] P. Edward, A. Muhammad, S. Elzeiny, M. Ashour, T. Elshabrawy, and J. Robert, "Enhancing the capture capabilities of lora receivers," in *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 2019, pp. 1–6.

[18] J. Tapparel, A. Balatsoukas-Stimming, and A. Burg, "Lora preamble detection robust to inter-channel interference," in *2023 IEEE 24th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2023, pp. 516–520.

[19] R. Ghanaatian, O. Afisiadis, M. Cotting, and A. Burg, "Lora digital receiver analysis and implementation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 1498–1502.

[20] C. Bernier, F. Dehmas, and N. Deparis, "Low complexity lora frame synchronization for ultra-low power software-defined radios," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 3140–3152, 2020.

[21] W. Wu and W. Wang, "Preamble structure and timing advance method for satellite iot," *IEEE Wireless Communications Letters*, pp. 1–1, 2024.

[22] Eutelsat, https://www.eutelsat.com/en/home.html, 2023.

[23] G. Pasolini, "On the LoRa Chirp Spread Spectrum Modulation: Signal Properties and Their Impact on Transmitter and Receiver Architectures," *IEEE Transactions on Wireless Communications*, vol. 21, no. 1, pp. 357–369, 2022.

[24] STMicroelectronics, "Ultra-low power, high performance, sub-1 GHz transceiver," 2021. [Online]. Available: https://www.st.com/resource/en/datasheet/s2-lp.pdf

[25] L. Vangelista, B. Jechoux, J.-X. Canonici, and M. Zorzi, "Golden modulation: a new and effective waveform for massive iot," *IEEE Transactions on Communications*, pp. 1–1, 2023.

[26] B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication," in *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, 2016, pp. 1–5.

[27] T. T. Nguyen, H. H. Nguyen, R. Barton, and P. Grossetete, "Efficient Design of Chirp Spread Spectrum Modulation for Low-Power Wide-Area Networks," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9503–9515, 2019.

[28] LoRa Alliance, "RP2-1.0.2 LoRaWAN Regional Parameters," 2020. [Online]. Available: https://lora-alliance.org/resource_hub/rp2-102-lorawan-regional-parameters/
[29] L. Qu, J. He, and C. Assi, "Understanding the benefits of successive interference cancellation in multi-rate multi-hop wireless networks," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 354–360.
[30] Y. Qian, L. Ma, and X. Liang, "The acquisition method of symmetry chirp signal used in leo satellite internet of things," *IEEE Communications Letters*, vol. 23, no. 9, pp. 1572–1575, 2019.
[31] ——, "The performance of chirp signal used in leo satellite internet of things," *IEEE Communications Letters*, vol. 23, no. 8, pp. 1319–1322, 2019.
[32] S. Jung, S. Jeong, J. Kang, J. G. Ryu, and J. Kang, "Transceiver design and performance analysis for lr-fhss-based direct-to-satellite iot," *IEEE Communications Letters*, vol. 27, no. 12, pp. 3310–3314, 2023.
[33] Arm, "Neon Overview," https://developer.arm.com/Architectures/Neon, 2023, last accessed 2022-04-14.
[34] *ARM NEON Intrinsics Reference (IHI 0073A)*, ARM, May 2014.
[35] P. Zamojski and R. Elfouly, "Developing standardized SIMD API between intel and ARM NEON," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018, pp. 1410–1415.
[36] B. Le Gal and C. Jego, "High-throughput LDPC decoder on low-power embedded processors," *IEEE Communications Letters*, vol. 19, no. 11, pp. 1861–1864, 2015.
[37] M. Frigo and S. Johnson, "The design and implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005.
[38] G. D. Krebs, "YAM 3, 5, 6. Gunter's Space Page," https://space.skyrocket.de/doc_sdat/yam-3.htm, 2023.

VIII. Biography

Matthieu MAGNANT received the M.S. degree in 2023 from the ENSEIRB-MATMECA engineering school, Bordeaux, France, in the field of Electrical Engineering and Embedded Systems. He is now a PhD student at the IMS laboratory in collaboration with Eutelsat R&D and Innovation team. His thesis focuses on the design and the optimization of a solution for detecting and processing interference aboard a Low-Earth Orbit satellite. He is the author of 2 peer-reviewed publications in conferences.

Bertrand LE GAL received the M.S. and Ph.D. degrees from the University of South Brittany, Lorient, France, in 2002 and 2005, respectively. In 2006, he became an Associate Professor in the IMS Laboratory, ENSEIRB-MATMECA engineering school, France. He received in 2023 the accreditation to supervise doctoral research from University of Bordeaux, France. Finally, he joined in 2023 the IRISA/INRIA laboratory of the University of Rennes 1 and the ENSSAT graduate engineering school at Lannion, France. His research focused on Algorithm-Architecture- Matching of ECC decoders on both hardware (ASIC/FPGA) and software targets (CPU/DSP/GPU). Its research topics currently includes complete digital communication system implementations and also RISC-V processor design under performance and security constraints. He is the author or co-author of 33 peer-reviewed publications in international journals and more than one hundred peer-reviewed articles in international conferences.

Guillaume FERRÉ received the Ph.D. degree in 2006 from the Limoges University of Technology in the field of digital communications and signal processing. From 2006 to 2008 he was postdoctoral researcher at the Limoges XLIM laboratory and then at the IMS laboratory in Bordeaux. Since 2021 he is full professor at ENSEIRB-MATMECA, an engineering school of Bordeaux INP. After several administrative responsibilities in the Telecommunications Department of ENSEIRB-MATMECA, Guillaume Ferré is now the Director of Industrial Relations at ENSEIRB-MATMECA. He carries out his research activities within the IMS laboratory in the "signal and image" team. These fields of research concern the circuits and systems for digital communications, it includes: Signal processing and digital communications, Digital enhancement for wideband power amplifiers and time interleaved analog to digital converters. Guillaume Ferré is the author of more than 170 papers in international journals and conferences. He is also author of 12 patents. He currently supervises 6 PhD students with a significant part of industrial research activities. He is member of several Technical Program Committees. He is the Principal Investigator (PI) of many national and international projects, at the local level he is responsible for two research activities related to IoT, including one to investigate on the smart campus.

Florian COLLARD received an engineering degree from TELECOM Bretagne. He finalized his degree with the specialized master "Space Communication Systems" at "Institut Supérieur de l'Aéronautique et de l'Espace" at Toulouse. He pursued a PhD in collaboration with Eutelsat Innovation team and ISAE SCANR team and received it in 2014. His research focused on the analysis and the optimization of the Enhanced Spread Spectrum Aloha. He continued to work in Eutelsat as Communications Systems Engineer. He has been involved in the development of the SmartLNB, an innovative connectivity solution for low-throughput applications. In recent years, he has been involved in the specification and the deployment of the first low-earth orbit satellites operated by Eutelsat with a focus on the physical layer definition.