



HAL
open science

Time sliding window for the detection of CCSK frames

Camille Monière, Kassem Saied, Bertrand Le Gal, E. Boutillon

► **To cite this version:**

Camille Monière, Kassem Saied, Bertrand Le Gal, E. Boutillon. Time sliding window for the detection of CCSK frames. IEEE Workshop on Signal Processing Systems (SiPS'2021), Oct 2021, Combrria, Portugal. hal-03404782

HAL Id: hal-03404782

<https://hal.science/hal-03404782>

Submitted on 26 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Time sliding window for the detection of CCSK frames

Monière Camille
Lab-STICC (UMR 6285)
Université de Bretagne Sud
Lorient, France

Saied Kassem
Lab-STICC (UMR 6285)
Université de Bretagne Sud
Lorient, France

Le Gal Bertrand
IMS (UMR 5218)
Bordeaux INP
Bordeaux, France

Boutillon Emmanuel
Lab-STICC (UMR 6285)
Université de Bretagne Sud
Lorient, France

Abstract—In wireless communications, frame detection and synchronization are usually performed using a preamble that consumes bandwidth and resources. A new type of frame called Quasi Cyclic Short Packet offers the advantage of avoiding preamble (thus saving resource) while allowing simple detection algorithm. The paper presents a time method to simplify the proposed detection algorithm and makes it robust to channel gain variation. First results show that a receiver can detect reliably short packet transmitted at few 100 Kbits/s at very low signal-to-noise ratio (-10 dB, typically).

Index Terms—Detection, Cyclic Code Shift Keying, Short frame, Internet of Things, Software Implementation

I. INTRODUCTION

In the context of massive Internet of Things (IoT), transmitting small payloads of data in an unsupervised communication system is a real challenge. Usually, the frame is composed of a payload, some redundancy to correct transmission errors, and a preamble known at the receiver side to help the synchronization task. Unfortunately, for short packets, the preamble size is no longer negligible compared to the encoded data size. In IoT context, preamble usage wastes rare spectrum resources and energy.

Reducing “metadata” throughput (i.e., the exchange of information linked to signaling, synchronization, and identification) is a fundamental change in the massive IoT networking [1]. In [2], Polyanskiy shows that classical methods which use coordination for synchronization and collision avoiding are far from the optimum, since the energy used for coordination is simply wasted. The problem of frame detection at low Signal-to-Noise Ratio (SNR) is also examined in the literature. Many papers propose detection, frequency, and time synchronization algorithms based on the transmission of a preamble heading for each frame [3]–[8]. As these classical preamble-based methods allow to greatly simplify the receiver complexity, because a part of the received waveform is known, they consume a significant part of the bandwidth when the message payload is small. Different preamble-less strategies for the transmission/reception of short packets have been addressed in state-of-the-art papers [9]–[13]. The efficiency of these proposed algorithms has been proved only for positive SNR values (i.e., $\text{SNR} > 0$ dB).

Recently, a new frame without preamble called Quasi Cyclic Short Packet (QCSP) has been proposed [14]. A QCSP frame is based on the association of a Cyclic Code Shift Keying (CCSK) modulation and a Non-Binary Error Control Code (NB-ECC). It is shown in [14] that the frame can be detected reliably. For instance, a Miss Detection Probability (\mathcal{P}_{MD}) of 10^{-4} for a False Alarm Probability (\mathcal{P}_{FA}) below 10^{-5} is reach even with SNR values below 0 dB (see section V). Moreover, the proposed algorithm is robust to frequency offset. It is based on coherent correlation, in the frequency domain, at the symbol level, but an incoherent accumulation at the frame level. This paper explores furthermore the problem of QCSP frame detection and proposes three new contributions. The first contribution of the paper is to present a normalization function that makes the detection algorithm immune to variation of channel and receiver gain. The second contribution is to propose the “time sliding window” algorithm that replaces correlation in frequency domain by a correlation in time domain that takes full profit of the QCSP frame structure. Finally, the last contribution is to evaluate the trade-off between detection performance and processing throughput for the existing and proposed algorithm on Intel Xeon processors. In particular, it is shown that the time sliding method offers significant gain in detection performance without significant impact on the maximum processing throughput for a software solution.

The article is structured as follows. Section II presents the system model and the state-of-the-art detection methods of the proposed QCSP frame. In section III, the CCSK based detection method is briefly described. In section IV, components of interest variants are investigated, especially the associated complexity. The results of our simulations and experiments are summarized in sections V and VI. Finally, conclusions and perspectives are reported in section VII.

II. SYSTEM MODEL

This section aims to introduce the system model and its main components. It is focused on the link between CCSK and NB-ECC, the CCSK principle and the channel considered. For readability, vectors are denoted with bold notations.

A. Structure of a QCSP frame

Let’s consider $\text{GF}(q)$ the Galois Field of order $q = 2^p$, $p \in \mathbb{N}$. Fig. 1 shows the overall communication chain of the QCSP

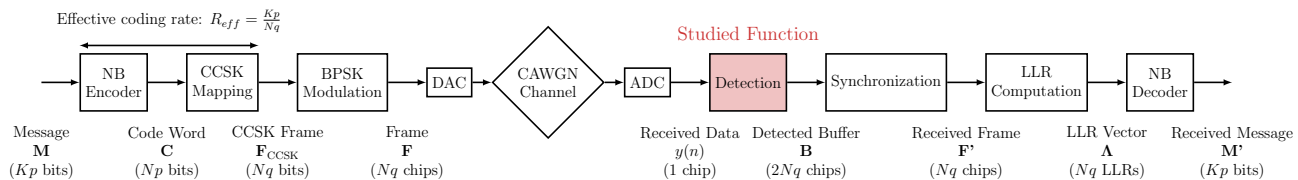


Fig. 1. QCSP System Model.

system. At the transmitter side, a NB-ECC of coding rate $R_c = \frac{K}{N}$, encodes an input message \mathbf{M} of K symbols of p bits to N symbols of p bits forming the codeword \mathbf{C} . The NB-ECC can be a NB-LDPC, since they are efficient for small frame length [15]. The CCSK modulation maps each GF symbol c of the codeword \mathbf{C} to a circular shift of an initial binary sequence \mathbf{P}_0 , with good auto-correlation properties. Even if \mathbf{P}_0 's length can be any integer [16], the paper studies only the case where it is set to q , defining a modulation rate $R_m = \frac{p}{q}$. The circularly k -right-shifted version of \mathbf{P}_0 is denoted \mathbf{P}_k , thus, $P_k(i) = P_0(i - k \bmod q)$, $i = 0, 1, \dots, q-1$. A symbol c belongs to $\text{GF}(q)$ but can also be considered as an element of $\llbracket 0, q-1 \rrbracket$ using its binary representation. With this convention, the CCSK consists of mapping each c to a circular c -right-shift of \mathbf{P}_0 such that

$$\forall c \in \mathbf{C} : \text{CCSK}(c) = \mathbf{P}_c. \quad (1)$$

The combination of the two rates: R_m and R_c , defines the QCSP effective rate $R_{eff} = \frac{Kp}{Nq}$. The whole CCSK frame \mathbf{F}_{CCSK} is the concatenation of N encoded CCSK symbols. Finally, the QCSP frame \mathbf{F} is obtained after applying Binary Phase Shift Keying (BPSK).

B. Demodulation of a QCSP frame

The Complex Additive White Gaussian Noise (CAWGN) channel adds to the signal a complex white noise of power σ^2 . The SNR of the channel is defined at the chip level as $\frac{1}{\sigma^2}$. In a synchronous channel, the received frame \mathbf{F}' is the sum of the frame \mathbf{F} and a vector of CAWGN \mathbf{Z} . Each received CCSK symbol \mathbf{Y}_ℓ is demodulated by correlating \mathbf{Y}_ℓ with each of the q possible received sequences \mathbf{P}_k , $k = 0, 1, \dots, q-1$ to generate the vector $\mathbf{L}_\ell = (L_\ell(0), L_\ell(1), \dots, L_\ell(q-1))$ where the k components are defined as

$$L_\ell(k) = \sum_{i=0}^{q-1} Y_\ell(i) P_k(i \bmod q). \quad (2)$$

Since $P_k(i) = P_0(i - k \bmod q)$ by construction, \mathbf{L}_ℓ is equal to the circular correlation of \mathbf{Y}_ℓ and \mathbf{P}_0 , i.e.,

$$\mathbf{L}_\ell = \mathbf{Y}_\ell \star \mathbf{P}_0, \quad (3)$$

where \star denotes the circular correlation. This computation can thus be done in the spectral domain [17], [16] as

$$\mathbf{L}_\ell = \text{IFFT}(\text{FFT}(\mathbf{Y}_\ell)^* \odot \text{FFT}(\mathbf{P}_0)), \quad (4)$$

where operator \odot denotes the element-wise (or Hadamard) product of two vectors and \mathbf{X}^* represents the conjugate vector of \mathbf{X} , i.e., $\mathbf{X}^* = \text{Re}(\mathbf{X}) - j\text{Im}(\mathbf{X})$.

From \mathbf{L}_ℓ , the Log-Likelihood Ratio (LLR) related to \mathbf{Y}_ℓ is computed as $\text{Re}(\frac{2\mathbf{L}_\ell}{\sigma^2})$ [17] and the full frame LLR vector (\mathbf{A} on Fig. 1) is sent to the NB decoder which retrieve a message \mathbf{M}' .

C. Channel model

In this paper, the channel is considered asynchronous where the time of arrival is unknown. Let's define Δ the start of the frame. Clock precision errors are also considered, leading to a random frequency offset added to the frame. It results in an unknown rotation of the received frame. The resulting random symbol rotation is denoted as: θ in the sequel (giving $\frac{\theta}{q}$ at chip level) to stress out its effect on symbols. Finally, the phase offset is unknown too but has no effect in the paper's context since only correlation norms are used in detection. It is then set to zero in the model. It is also assumed that there is enough time between each message to ensure no interference. With \mathbf{F} the frame sent and \mathcal{I} the interval $[\Delta, \Delta + Nq - 1]$, the channel output \mathbf{Y} is sampled at frequency f_c into samples $y(n)$ such that, for all $n \in \mathbb{N}$:

$$y(n) = \begin{cases} Z(n) & \text{if } n \notin \mathcal{I}, \\ F(n - \Delta)e^{jn\frac{\theta}{q}} + Z(n) & \text{else} \end{cases} \quad (5)$$

The sampling frequency f_c is equal to the inverse of the chip period T_c , i.e. we consider that the optimal sampling time is known. This is achievable in practice by testing in parallel all phase hypotheses resulting from oversampling and by keeping the best one.

The goal of the detection stage is to coarsely approximate Δ and θ , i.e., find a time interval and a rotation interval containing Δ and θ respectively. The synchronization stage then mitigates or even removes their effects. It thus processes received data and when something is detected, it outputs a buffer (\mathbf{B} on Fig. 1) of $2Nq$ chips which should contain the expected frame. This size is related to an uncertainty on symbol synchronization at low SNR. Synchronization steps executed forward are not addressed in this paper.

III. CCSK BASED DETECTION

This section sums up the detection method presented in [14]. It first describes the score function used to assess if a new frame arrived, then the principle of a time and frequency

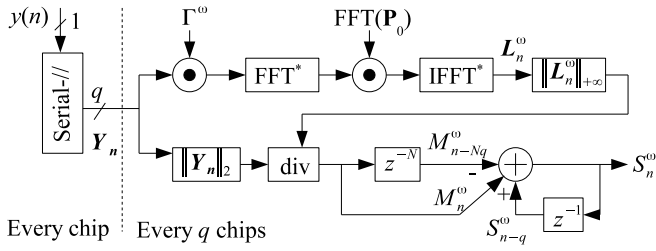


Fig. 2. Score Processing Unit performed in the frequency domain.

research grid is presented. Finally, it shows that the proposed method is sensitive to a scaling factor in input, which lead us to propose a solution to overcome this problem.

A. Score function

CCSK based detection consists in comparing a score function to a threshold value U_0 . The score is computed using the last Nq received chips (length of a frame) at time n divided in N sub-vectors $\mathbf{Y}_{n-(N-1)q}, \mathbf{Y}_{n-(N-2)q}, \dots, \mathbf{Y}_n$, where for all $m \in \mathbb{N}$, $\mathbf{Y}_m(i) = (y(m - (q-1) + i))_{i=0,1,\dots,q-1}$ (length of a symbol). If the score value exceed threshold U_0 , a new frame arrival is assumed.

The score function S_n^ω corresponds to a filter output that is maximized for a frame arrived at time n with a frequency offset $f = \frac{\omega}{2\pi q}$. The filter is locally coherent at the symbol level and non-coherent at the frame level due to potential residual frequency offset. At the symbol level, the first step is to mitigate the frequency offset by multiplying term by term \mathbf{Y}_n with the vector Γ^ω which is a pure complex sinusoidal of frequency $-f$, i.e. $\Gamma^\omega = (1, e^{-j\frac{\omega}{q}}, e^{-j\frac{2\omega}{q}}, \dots, e^{-j\frac{(q-1)\omega}{q}})$. This operation gives $\mathbf{Y}_n^\omega = \mathbf{Y}_n \odot \Gamma^\omega$. The residual frequency offset of \mathbf{Y}_n^ω is supposed low enough to allow a coherent demodulation of the symbol, thus, using (3), the correlation vector \mathbf{L}_n^ω is computed as

$$\mathbf{L}_n^\omega = (\mathbf{Y}_n \odot \Gamma^\omega) \star \mathbf{P}_0. \quad (6)$$

The maximum absolute value of \mathbf{L}_n^ω is taken in order to perform the non-coherent integration over the whole frame, which gives with, $M_n^\omega = \|\mathbf{L}_n^\omega\|_\infty = \max\{|L_n^\omega(i)|, i = 0, 1, \dots, q-1\}$,

$$S_n^\omega = \sum_{i=0}^{N-1} M_{n-iq}^\omega. \quad (7)$$

However, if the input is scaled by a factor χ , then the score function S_n^ω is scaled by the same factor χ . This is problematic since S_n^ω is compared to the threshold U_0 to evaluate the presence of a new frame. The first contribution of the paper is to normalize M_n^ω by the 2-norm of \mathbf{Y}_n to be independent of the input scaling factor, i.e.

$$M_n^\omega = \frac{\|\mathbf{L}_n^\omega\|_\infty}{\|\mathbf{Y}_n\|_2}. \quad (8)$$

Given that S_n^ω is obtained through an averaging moving filter of length N , (7) can be simplified as

$$S_n^\omega = S_{n-q}^\omega + M_n^\omega - M_{n-Nq}^\omega. \quad (9)$$

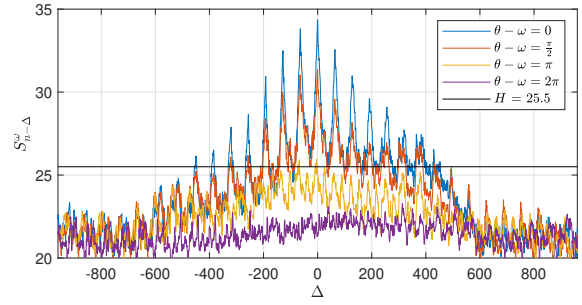


Fig. 3. Example of score values S_n^ω as a function of Δ and of the residual frequency offset error $\theta - \omega$ at an SNR of -7 dB.

The overall architecture that allows to compute simply the score S_n^ω every q chips is given in Fig. 2, and is called Score Processing Unit (SPU). In [14], the correlation is done in the frequency domain using (4).

Fig. 3 shows an example of a correlation result for a frame of length $N = 10$, with a CCSK modulation of length $q = 64$, with CAWGN, and with an SNR of -7 dB. The value of Δ is equal to 0 at the exact time of arrival of the frame. In this curve, several frequency errors are indicated, with $\theta - \omega$ taking the values $0, \pi/2, \pi$, and 2π (a full rotation of each CCSK symbol). In this example, the threshold value is set to $U_0 = 25$. One can note that when the residual frequency offset is high (greater than a rotation of π for each CCSK symbol, typically), the score magnitude is significantly reduced; Likewise, when the time offset error verifies $\Delta = q/2 \pmod q$, the detection of the frame more difficult. The solution is to explore in parallel several hypotheses of time delay and frequency offset.

B. Search grid

Let us assume, without loss of generality, that the frequency offset θ is bounded between $-\pi$ and π , i.e. giving at maximum a half clockwise or counterclockwise rotation per CCSK symbol. Then it is possible to divide this frequency interval into p_ω sub-intervals, each associated to a score filter $S_n^{\omega(r)}$ with $\omega(r) = \pi(-1 + \frac{2r+1}{p_\omega})$, $r = 0, 1, \dots, p_\omega - 1$. So, the maximum distance between θ and the closest $\omega(r)$ value is bounded by $\vartheta = \frac{\pi}{p_\omega}$. For example, when $p_\omega = 4$, $\vartheta = \frac{\pi}{4}$ which corresponds to $1/8^{th}$ of residual rotation per CCSK symbol.

For the time uncertainty, a similar approach is used to limit the computational resource: the CCSK size q is divided into p_Δ section of length $l = \frac{q}{p_\Delta}$. Each l new received chips, the score functions $S_{tl}^{\omega(r)}$, $t \in \mathbb{N}, r = 0, 1, \dots, p_\omega$ are computed. In summary, every CCSK symbol (thus q received chips), $p_\omega p_\Delta$ circular correlations (2) are computed and the $p_\omega p_\Delta$ score functions are updated using (5).

Each time a value of S_n^ω goes above the threshold U_0 , the local maximum score value is searched in a window of values around (ω, n) . Once the local maximum found, the synchronization process starts (not described in the paper). The whole detection system is shown in Fig. 4.

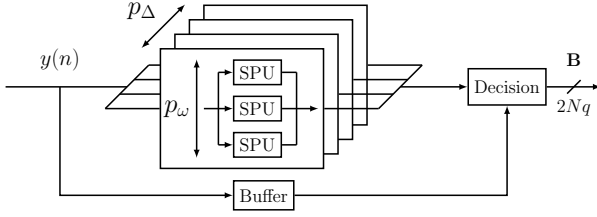


Fig. 4. Complete theoretical detection system.

IV. TIME SLIDING WINDOWS

In this section, the frequency domain computation of the correlation vector \mathbf{L}_n^ω is replaced by a “time sliding” computation. The name “time sliding” comes from the computation scheduling that uses the circular property of the CCSK modulation to reduce dramatically the computation burden.

Let us express the k^{th} component $L_n^\omega(k)$ of \mathbf{L}_n^ω given in (6) in the time domain. To lighten notations, $y_n(i)$ will refer to $y_n(i) = y(n - (q - 1) + i)$, thus $\mathbf{Y}_n = (y_n(0), y_n(1), \dots, y_n(q - 1))$. Moreover, we have the following useful properties: $y_n(-1) = y(n - q)$, $y_n(q - 1) = y(n)$ and, for all n, i , $y_{n-1}(i) = y_n(i - 1)$.

$$L_n^\omega(k) = \sum_{i=0}^{q-1} y_n(i) P_k(i) e^{-j \frac{i\omega}{q}} \quad (10)$$

Since M_n^ω is equal to the maximum absolute value of vector \mathbf{L}_n^ω , its value is not affected if \mathbf{L}_n^ω is replaced in (8) by $\bar{\mathbf{L}}_n^\omega(k) = e^{j(n-q)\frac{\omega}{q}} \mathbf{L}_n^\omega$. Thus, (10) can be replaced by

$$\bar{L}_n^\omega(k) = \sum_{i=0}^{q-1} y_n(i) P_k(i) e^{-j(n-q+i)\frac{\omega}{q}}, \quad (11)$$

$\bar{L}_n^\omega(k)$ can be expressed as a function of $\bar{L}_{n-1}^\omega(k-1)$ and the values $y(n-q)$ and $y(n)$. According to (11), $\bar{L}_{n-1}^\omega(k-1)$ becomes

$$\bar{L}_{n-1}^\omega(k-1) = \sum_{i=0}^{q-1} y_{n-1}(i) P_{k-1}(i) e^{-j(n-1-q+i)\frac{\omega}{q}} \quad (12)$$

By the definition of the CCSK modulation, for all k and i values, $P_{k-1}(i) = P_0(k-1+i) = P_k(i-1)$, where additions are performed modulo q , thus (12) can be rewritten as

$$\bar{L}_{n-1}^\omega(k-1) = \sum_{i=0}^{q-1} y_n(i-1) P_k(i-1) e^{-j(n-q+i-1)\frac{\omega}{q}} \quad (13)$$

By changing the summation i by $i' = i - 1$, (13) can be rewritten as

$$\bar{L}_{n-1}^\omega(k-1) = \sum_{i'=-1}^{q-2} y_n(i') P_k(i') e^{-j(n-(q-1)+i')\frac{\omega}{q}} \quad (14)$$

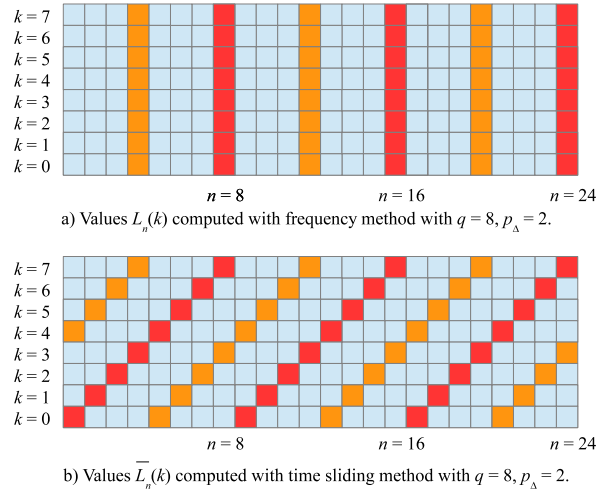


Fig. 5. Comparison between frequency based and time based computation of correlations for $q = 2$ and $p_\Delta = 2$. Red and orange points correspond respectively to the output of two distinct SPUs.

Thus, subtracting $y_n(-1)P_k(-1)e^{j(n-q)\frac{\omega}{q}}$ and adding $y_n(q-1)P_k(q-1)e^{jn\frac{\omega}{q}}$ to $\bar{L}_{n-1}^\omega(k-1)$ gives $\bar{L}_n^\omega(k)$. In summary, by denoting $d_n^\omega = (y(n) - y(n-q))e^{jn\frac{\omega}{q}}e^{-jn\frac{\omega}{q}}$,

$$\bar{L}_n^\omega(k) = \bar{L}_{n-1}^\omega(k-1) + P_k(q-1)d_n^\omega. \quad (15)$$

This recursive equation can be exactly implemented in hardware as long as the multiplication by $e^{j\omega}$ does not introduce a rounding error that would be accumulated in the integrator. Particularly, the values ϑ of interest $\vartheta = \{\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{8}\}$ (corresponding to $p_\omega = \{2, 4, 8\}$ respectively) are not affected.

Fig. 5 compares the effect of the correlation using FFT and time sliding methods, for a $q = 8$ CCSK. In both cases, q values $L_n^\omega(k)$ are computed but with a fundamental difference: with the FFT method, every $l = q/p_\Delta$ chips a full vector \mathbf{L}_n^ω of length q is computed (a vertical bar in Fig. 5 every q cycles) while time sliding based SPUs generate every $l = q/p_\Delta$ chips a full diagonal of elements $\bar{L}_{n+k}^\omega(k \bmod n)$.

To make (8) computable, all the $\bar{L}_{n+k}^\omega(k)$, $k = 0, 1, \dots, q-1$ values should be available at time n . The time sliding method then imposes to use $p_\Delta = q$ and $l = 1$. Hardware implementation of time sliding correlation method is depicted in Fig. 6.

In summary, the algorithm parameters impacting frame detection are (p_Δ, p_ω) , with p_Δ set at q when using time sliding correlation. These parameters directly impact performance and complexity. The following two sections are dedicated respectively to the performance and complexity evaluation.

V. PERFORMANCE EVALUATION

In this section, we compare the performance in terms of “Receiver Operating Characteristic” (ROC). ROC curve gives the evolution of False Alarm Probability ($\mathcal{P}_{FA} = \text{Prob}(S_n^\omega \geq U_0)$ when no frame is received) and the undetected probability or Miss Detection Probability ($\mathcal{P}_{MD} = \text{Prob}(S_n^\omega < U_0)$ when a frame is received at time n). To compare with work reported

VI. SOFTWARE IMPLEMENTATION

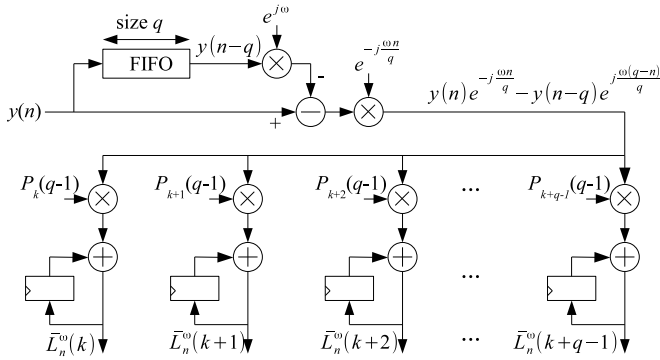


Fig. 6. Time slide correlation hardware architecture for a given frequency offset value ω .

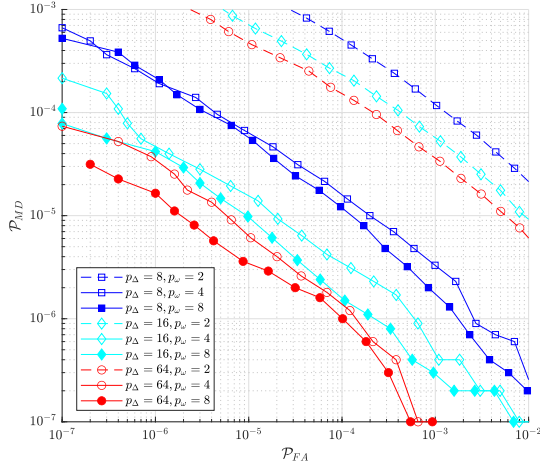


Fig. 7. ROC curve for frame of size $N = 60$, $q = 64$ at SNR of -10 dB for several parallelism levels p_{Δ} and p_{ω} .

in [14], the frame size is set to $N = 60$ with a CCSK length of $q = 64$.

The impact of (p_{Δ}, p_{ω}) values on detection performances are presented in Fig. 7. Different configuration were evaluated with $p_{\Delta} = \{8, 16, 64\}$, which correspond to $l = \{8, 4, 1\}$ respectively, and $p_{\omega} = \{2, 4, 8\}$ which correspond to a $\vartheta = \{\frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{8}\}$ respectively. Results show that a linear increase of the total parallelism level (represented by the product $p_{\Delta}p_{\omega}$) does not result in a linear performance improvement. If doubling $p_{\omega} = 2$ to 4 improves performance for every possible p_{Δ} , it is not the case for $p_{\omega} = 4$ to 8.

Moreover, a given $p_{\Delta}p_{\omega}$ lead to different performance for each different (p_{Δ}, p_{ω}) couple. In other words, a constant total parallelism level (thus, a constant complexity) does not ensure the same performance. For example, achieving $p_{\Delta}p_{\omega} = 128$ with $(p_{\Delta} = 64, p_{\omega} = 2)$ results in a decade less performance than with $(p_{\Delta} = 16, p_{\omega} = 8)$.

Thus, a trade-off is achievable between targeted performance and complexity. For instance, a $\mathcal{P}_{MD} < 10^{-4}$ with $\mathcal{P}_{FA} < 10^{-5}$ can be reach for $(p_{\Delta} = 8, p_{\omega} = 4)$ resulting in a complexity $p_{\Delta}p_{\omega} = 32$.

The main objective of the presented work is to implement a hardware receiver on ASIC/FPGA targets. Currently, the increasing processing performance of software programmable devices such as many-cores, associated easy to use programming model and efficient algorithmic descriptions [18]–[20], made the implementation of high-throughput communication system possible. Software-based implementation could not achieve throughput and energy efficiencies of ASIC/FPGA implementations however they provide flexibility, scalability and enables rapid prototyping. Consequently, to evaluate the proposed detection techniques a software based implementation was first released. Even if software implementation results could not clearly predict hardware implementation characteristics, it could provide trends and validate the behavior of the proposed techniques.

A multi-core system has been used for benchmarking. It is composed of a dual socket Intel Xeon Gold 6148 CPU with 256 GB of RAM memory. Each Xeon processor is composed of 20 physical processor cores that share a 28160KB L3 cache memory. The working frequency is 2.60 GHz but the turbo-boost feature enables cores to run up to 3.70 GHz when the power dissipation constraint is met. This platform was selected for benchmarking purpose because system receiver is not required to be low-power, and may not be a battery constrained system like the emitter.

The communication system was developed in C++11 language. The FFT based correlation implementation use the well-known deeply optimized FFTW3 library [21]. At the opposite the time sliding correlation method is implemented thanks to handmade C/C++ codes. The sources are written using floating point values in a way that can take advantage of GCC auto-vectorization feature. The throughput at emitter and receiver side was measured using the C++ Chrono API.

The emitter system includes a NB-LDPC encoder, a CCSK mapper and the BPSK modulation. Parameters are the same as previously ($N = 60$, $q = 64$) with a NB-LDPC code rate of $R_c = \frac{1}{3}$. These results in an effective code rate $R_{eff} = \frac{1}{32}$, calculated given the formula in section II, also visible in Fig. 1.

The measured emitter throughput reach up to 2.6 GChips/s. A such high value is due to the low complexity of encoding and mapping stages. Since CCSK mapping is only a rotation (efficiently done by memory remapping and copy), it is performed without visible cost on the platform. This value, compared with the upcoming transmitter results, highlights the low complexity of the emission process.

At receiver side, the system includes only the CCSK detector. Measured throughputs for different setups are reported in Table I. The throughput depends on the correlation method and the selected parameters. On one hand, executed on a single processor core, the FFT-based detector achieves 0.64 MChips/s when $p_{\Delta} = 16$ and $p_{\omega} = 8$. However, the throughput is improved up to 3.61 MChips/s by dedicating one processor core for each p_{Δ} filter (MT FFT in Table I). Doing so does not provide a p_{Δ} speedup since thread synchronization overhead.

TABLE I
CHIP RATES FOR DIFFERENT SETS OF PARAMETERS

Method	p_{Δ}	p_{ω}	Chip Rate	
FFT	8	4	2.14	MChips/s
		8	1.26	MChips/s
	16	4	1.01	MChips/s
		8	0.64	MChips/s
MT FFT	8	4	5.74	MChips/s
		8	4.38	MChips/s
	16	4	4.26	MChips/s
		8	3.61	MChips/s
Time Sliding	64	4	3.21	MChips/s
		8	1.99	MChips/s

On the other hand, the time sliding approach, that provides the best detection performances thanks to $p_{\Delta} = q$, achieves a decoding throughput of 1.99 MChips/s up to 3.21 MChips/s when on a single processor. Unfortunately, contrary to the FFT method, using multiple cores decreases the performance, task synchronization costing more than it gains.

At this point, FFT method is the most efficient method in terms of throughput for the receiver software implementation due to multi-core execution. These reception chip rates, which reach a few MChips/s, are already similar to those required in the Low-Rate Wireless Networks (LRWN) domain [22]. It makes the software receiver a viable prototype for CCSSK modulation evaluation.

These throughput performances should be higher for a hardware receiver implementation. However, the best correlation approach would certainly be different. Indeed, time sliding method seems more hardware friendly. This assertion is supported by the computational complexity comparison of the two methods. These complexity estimations reported in Table II tend to suggest that the hardware implementation of the time sliding approach would be less complex in terms of number of computations and arithmetic operation costs. Indeed, for time sliding estimation, in hardware, it is possible to replace the multiplications by $P_k(q-1)$ by an add/sub multiplexer because P_0 values are -1 or 1 only. So it can be implemented efficiently on ASIC/FPGA targets.

VII. CONCLUSION

The paper shows that software detection of QCSP frames is possible at very low SNR (-10 dB) for chip throughput around few MChips/s, compliant with the LRWN context. Two algorithmic contributions are also presented: the normalization of the correlation function that is mandatory when the channel gain varies and an efficient alternative to the frequency correlation in the form of the time sliding correlation. In the future, the work will be extended in several directions. First, floating-point arithmetic will be replaced by fixed-point operations in the software implementation to improve the throughput. Then, high level synthesis will be used to migrate the QCSP frame detector to an FPGA to allow real time transmission.

TABLE II
COMPLEXITY COMPARISON OF THE DIFFERENT METHODS.

Method	Add	Multiply
FFT	$p_{\omega} p_{\Delta} 2q \log_2(q)$	$p_{\omega} p_{\Delta} q(\log_2(q) + 2)$
Time Sliding (Algorithm)	$p_{\omega} q(1 + q)$	$p_{\omega} q^2$
Time Sliding (Hardware)	$p_{\omega} q(1 + q)$	$p_{\omega} q$

REFERENCES

- [1] G. Durisi, T. Koch, and P. Popovski, "Toward Massive, Ultrareliable, and Low-Latency Wireless Communication With Short Packets," *Proceedings of the IEEE*, vol. 104, no. 9, 2016.
- [2] Y. Polyanskiy, "Asynchronous Communication: Exact Synchronization, Universality, and Dispersion," *IEEE Transactions on Information Theory*, vol. 59, no. 3, 2013.
- [3] S. Kim, K. Joo, and Y. Lim, "A delay-robust random access preamble detection algorithm for LTE system," in *Proceedings of IEEE Radio and Wireless Symposium*, 2012.
- [4] G. Wu, S. Hu, and S. Li, "Low complexity time-frequency synchronization for transform domain communications systems," in *Proceedings of ChinaSIP*, 2015.
- [5] Zhenzhen Ye, Chunjie Duan, P. V. Orlik, Jinyun Zhang, and A. A. Abouzeid, "A Synchronization Design for UWB-Based Wireless Multimedia Systems," *IEEE Transactions on Broadcasting*, vol. 56, 2010.
- [6] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, 1997.
- [7] D. Zhu and R. W. Heath, "Directional timing synchronization in wide-band millimeter wave cellular systems with low-resolution ADCs," in *Proceedings of ACSSC*, 2017.
- [8] M. Schluter, M. Dörpinghaus, and G. P. Fettweis, "Bounds on Phase, Frequency, and Timing Synchronization in Fully Digital Receivers With 1-bit Quantization and Oversampling," *IEEE Transactions on Communications*, vol. 68, no. 10, 2020.
- [9] D. Godard, "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems," *IEEE Transactions on Communications*, vol. 28, no. 11, 1980.
- [10] T. Fujita, D. Uchida, Y. Fujino, O. Kagami, and K. Watanabe, "A burst modulation/demodulation method for short-packet wireless communication systems," in *Proceedings of APCC*, 2008.
- [11] A. Azari, P. Popovski, G. Miao, and C. Stefanovic, "Grant-Free Radio Access for Short-Packet Communications over 5G Networks," in *Proceedings of GLOBECOM*, 2017.
- [12] B. Bloessl and F. Dressler, "mSync: Physical Layer Frame Synchronization without Preamble Symbols," *IEEE Transactions on Mobile Computing*, vol. 17, no. 10, 2018.
- [13] P. Walk, P. Jung, B. Hassibi, and H. Jafarkhani, "MOCZ for Blind Short-Packet Communication: Practical Aspects," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, 2020.
- [14] K. Saied, A. C. A. Ghouwayel, and E. Boutillon, "Quasi Cyclic Short Packet for asynchronous preamble-less transmission in very low SNRs," *Preprint HAL*, no. 02884668, 2020.
- [15] M. Davey and D. MacKay, "Low-density parity check codes over GF(q)," *IEEE Communications Letters*, vol. 2, no. 6, 1998.
- [16] G. Dillard, M. Reuter, J. Zeidler, and B. Zeidler, "Cyclic code shift keying: A low probability of intercept communication technique," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 3, 2003.
- [17] O. Abassi, L. Conde-Canencia, M. Mansour, and E. Boutillon, "Non-binary low-density parity-check coded cyclic code-shift keying," in *Proceedings of WCNC*, 2013.
- [18] C.-L. I and al., "Recent Progress on C-RAN Centralization and Cloudification," *IEEE Access*, vol. 2, 2014.
- [19] A. Checko and al., "Cloud RAN for Mobile Networks—A Technology Overview," *IEEE Communications Surveys & Tutorials*, vol. 17, 2015.
- [20] C. Mavromoustakis, G. Mastorakis, and C. Dobre, Eds., *Advances in Mobile Cloud Computing and Big Data in the 5G Era*, ser. Studies in Big Data, 2017.
- [21] M. Frigo and S. Johnson, "The Design and Implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, 2005.
- [22] "IEEE Std 802.15.4-2020, IEEE Standard for Low-Rate Wireless Networks," 2020.