This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3344471

Balancing Computational Efficiency and Detection Accuracy in Oversampled Frequency Shift Chirp Modulation

Thomas Ameloot, Hendrik Rogier, *Senior Member, IEEE*, Patrick Van Torre, *Member, IEEE*, Marc Moeneclaey, *Fellow, IEEE*

***Abstract*—Although first proposed a few decades ago, chirp-based modulation has recently seen a surge of popularity as a result of its application in the LoRa standard. Over the past years, this key Internet of Things (IoT) enabler has been well researched, and ever more advanced low-power wide-area networks (LPWANs) are being implemented across the globe, based on this technology. As a result of this international acclaim, multiple actors have invested efforts into implementing frequency shift chirp modulation (FSCM), which is a more general term for the open-source physical layer modulation protocol also embedded in the LoRa standard, on software defined radio (SDR) systems. However, while oversampling, advanced post-processing and other digital techniques have led to significant advances in the technology's capabilities and reliability, real-world deployments of these SDR implementations have to overcome the excessive computational cost associated to these techniques. In response to this challenge, this paper examines several new strategies for symbol detection methods operating on FSCM signals, such as those employed in LoRa modulation, enabling significant computational cost reductions. Examples of these are integrating frequency correction in the dechirping procedure and omitting the downsampling operation by using upsampled downchirps when processing the received samples. In comparison to the standard detection method, computational efficiency gains between 19 % and 36 % are achieved. Hence, applying the methods presented in this work can yield significant reductions in power consumption for real-world SDR-based FSCM systems in state-of-the-art IoT deployments.**

***Index Terms*—Internet of Things, Wireless Sensor Networks, LoRa, Software Defined Radio, Detection**

I. Introduction

In low-power wide-area networks (LPWANs), a vast amount of new applications have been made possible by the emergence of new Internet of Things (IoT) communication technologies. Although other standards exist that often tailor to more specific use cases, LoRa [1], SigFox [2] and NB-IoT [3] have seen the most adoption worldwide. The first of these, LoRa, provides a very respectable range for the amount of power that it consumes by sacrificing some of the achievable data rate. In practice, this is achieved by applying frequency shift chirp modulation (FSCM), also dubbed frequency shift chirp spread spectrum (FSCSS) modulation, a technique that encodes information by varying the start frequency of linear chirps. By encoding data in these chirp signals, thereby spreading information over time and frequency, communication is made possible at very low signal-to-noise ratios (SNRs). Various published works provide very interesting overviews on the application of FSCM in LoRa modulation [4]–[7]. Furthermore, given the especially large research landscape covering the practical communication performance of LoRa, which has featured indoor links [8]–[15], outdoor links [16]–[20], season/weather dependency [21]–[24], network scalability [5], [25]–[27] and more, it should be no surprise that also more theoretical analyses of the underlying FSCM technology have amassed much interest [28], [29].

Recent efforts to implement FSCM on software defined radio (SDR) platforms have both relied on these analyses and strengthened them through the capability of simulating signal transmission and reception under specific conditions. In fact, implementing LoRa and/or other FSCM variations on SDR has been the goal of both open-source projects [30]–[32] and various scientific research efforts [33]–[37]. These SDR implementations have given researchers the tools to more deeply analyse link performance in challenging environments [38]–[44] and achieve improvements on the current communication performance [45]–[51]. However, one aspect that is more rarely highlighted is the computational cost of these SDR-based receivers. While some researchers simply focus on Nyquist-rate sampling, it has been shown that oversampling provides very powerful capabilities for FSCM receivers to correct sampling time offsets (STOs) and carrier frequency offsets (CFOs) [50], [51]. Evidently, this does incur a distinct computational cost, which is disadvantageous. This paper, however, presents new detection strategies that significantly reduce the amount of computational resources needed to perform symbol detection for oversampled FSCM signals, and thereby aims to combine the best of both worlds.

This work was partly funded by the Research Foundation - Flanders (FWO) through the "MUlti-SErvice WIreless NETwork", FWO-FRS Excellence of Science – EOS project, and the "Advanced multi-antenna communications in the radiative near-field through hybrid focusing and spatial multiplexing." project.

T. Ameloot, H. Rogier and P. Van Torre are with Ghent University - imec, IDLab, Department of Information Technology (INTEC), Technologiepark-Zwijnaarde 126, 9052 Zwijnaarde, Belgium (email: thomas.ameloot@ugent.be; hendrik.rogier@ugent.be; patrick.vantorre@ugent.be)

M. Moeneclaey is with Ghent University, Department of Telecommunications and Information Processing (TELIN), Sint-Pietersnieuwstraat 41, 9000 Gent, Belgium (email: marc.moeneclaey@ugent.be)

A. Related Work

Although a lot of literature exists on LoRa modulation and its variations, several related research papers published more recently should be referred to in more detail as these specifically discuss low-complexity FSCM decoding algorithms that correct for STO and/or CFO. On the one hand, in [36], a Nyquist-rate receiving algorithm was presented that, as a first in literature, also focused on attaining a low computational cost. Owing to this goal, its computational efficiency indeed appears to be very good. However, as the authors of [50] agree, this algorithm leaves a lot of detection performance on the table by limiting when and how time and frequency offsets are corrected. In [50] and [51], on the other hand, better estimators for STO and CFO are proposed, based on oversampling. Both incorporate their estimation and correction methods in a better performing detection algorithm. The main difference between these papers is the computational cost associated with refining the estimation of the STO and CFO. In fact, whereas the authors of [50] state that a joint estimation of the STO and CFO would be too complex for low-power end nodes, this is precisely what the authors of [51] propose for their most advanced algorithm version, stating that this solution is more suited to implementation on base station hardware and/or end nodes that have more computational power. Given that their papers were published in a similar time frame, it is our understanding that the authors of [50] and [51] were conducting their work in parallel, without mutual awareness. Hence, it is difficult to compare the overall performance of the algorithm versions presented in these papers directly as both were analyzed differently. However, assuming that the bit error rate (BER) is given by half of the packet error rate (PER), we see that applying the joint STO and CFO estimation method proposed in [51] indeed provides a detection accuracy performance benefit when compared to the method proposed in [50], justifying its supposed higher computational impact.

B. Contributions

This paper presents several new versions of the detection method for oversampled FSCM symbols. These novel methods are significantly more computationally efficient than the state-of-the-art methods featured in [50] and [51]. This is achieved by combining operations in the detection procedure, employing memory for storing often used signals and leveraging specific properties associated to oversampled FSCM signals. In general, the computational efficiency gains achieved here for symbol detection may give rise to a significant reduction of the power consumption of the entire FSCM receiver since trial detections make up the majority of the cost associated to signal presence detection and synchronization procedures [50], [51]. In addition to describing the newly proposed methods in detail, this paper also contains practical and theoretical error performance analyses, as well as updated computational cost assessments for the proposed detection procedures, allowing the direct comparison between old and new detection methods. The paper compares each of these procedures in detail to provide deep insight into how these methods work, how computational efficiency gains are achieved and what the impact of applying certain strategies is on error performance, setting the stage for potential future improvements by the research community.

C. Outline

The paper is structured as follows. First, in Section II, frequency shift chirp modulation itself is discussed, and relevant waveforms are described. In contrast to most other literature, this also includes details on the analog RF front end, and a mathematical formulation for oversampled chirps, as both are highly relevant to the methods described in this work. In Section III, the standard FSCM symbol detection method is described and summarized. This comprises an overview of the full receiver architecture, as computational efficiency gains are attainable at various stages of the receive algorithm. The actual strategies for lowering the computational cost of FSCM detection are presented in Section IV. This part of the paper also provides an overview of the memory requirements for each strategy. Next, in Section V, the newly proposed detection procedures are compared to the standard one in a practical BER analysis, and the error performance impact of using signals that are stored in memory is theoretically determined. Section VI comprises two distinct computational cost estimation efforts, which both clearly illustrate the efficiency to be gained when applying the proposed strategies. Finally, a conclusion completes the paper in Section VII.

II. Frequency Shift Chirp Modulation

In FSCM, information is encoded in cyclically shifted linear frequency chirps. More specifically, to encode a symbol $a \in \{0, ..., M-1\}$, where M indicates the size of the symbol alphabet, the complex baseband FSCM signal $s(t, a)$ with symbol duration T_s and power P_s is used. This signal is described by

$$s(t,a) = \sqrt{P_s} \exp\left[\jmath\, \phi(t,a)\right], \tag{1}$$

where

$$\phi(t,a) = \begin{cases} 2\pi \left(a\frac{t}{T_s} + \frac{M}{2}\frac{t^2}{T_s^2} \right) & 0 < t < \tau_a \\ 2\pi \left((1-\frac{t}{T_s})(M-a) + \frac{M}{2}\frac{t^2}{T_s^2} \right) & \tau_a < t < T_s \end{cases}, \tag{2}$$

and

$$\tau_a = (1-a/M)T_s. \tag{3}$$

The value for M is usually chosen as a power of two, with the exponent being described by the spreading factor (SF) of the modulation scheme, yielding $M = 2^{\mathrm{SF}}$. Furthermore, M is also related to the frequency swing B of the chirp through $M = BT_s$. Although its bandwidth is actually slightly larger than B, the signal $s(t, a)$ can be assumed as being contained in the frequency band $[0, B]$ for large M values [29]. The symbol energy in $s(t, a)$ is indicated as $E_s = P_s T_s$.

Why this modulation is referred to as FSCM is perhaps the most obvious when considering the instantaneous frequency f_t of the chirp introduced in (1). As a result of applying the cyclic shift, f_t starts at aB/M. During the symbol interval, f_t increases linearly with slope B/T_s until it reaches B.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3344471

At this point, i.e. when $t = \tau_a$, f_t is reset to 0, and then increases linearly with the same slope during the remainder of the symbol period. A spectrogram of the FSCM signal that encodes the symbol a is shown in the left pane of Fig. 2.

Assuming a sufficiently large value for M, the minimum sampling frequency needed to represent $s(t, a)$ digitally equals $B = M/T_s$. However, as will be outlined in the next section, oversampling is required when the bandwidth of the analog demodulation filter is larger than B. Hence, an oversampling factor is introduced, indicated as $K \in \mathbb{Z}_+$. As a result, the sampling interval becomes equal to $\frac{1}{KB} = \frac{T_s}{KM}$. When oversampling is applied (i.e. $K > 1$), the symbol period T_s contains $N = KM$ samples and the complex envelope $s_N[n, a]$ of the signal that represents the symbol a is described by

$$s_N[n, a] = s\left(\frac{nT_s}{N}, a\right), \tag{4}$$

where $n \in \{0, ..., N-1\}$ denotes the sample index and the subscript N is used to indicate the amount of samples in one symbol period. When no oversampling is used (i.e. $K = 1$ so that $N = M$), it has been shown in [29] and [51] that the resulting samples $s_M[m, a]$ of $s(t, a)$ from (1) can simply be represented as

$$s_M[m, a] = \sqrt{P_s} \exp\left[\jmath 2\pi \left(\frac{am}{M} + \frac{m^2}{2M}\right)\right], \tag{5}$$

Note that there is no need to explicitly include the frequency discontinuity at $t = \tau_a$ in this formulation.

III. Standard Symbol Detection

Several high-level journal publications have documented how to detect FSCM or LoRa symbols [28], [29], [48], [51], [52]. Most of these describe symbol detection at the Nyquist rate, often based on the digital representation of FSCM signals given by (5), or a variation thereof. However, for the purposes of this paper, it is beneficial to take into account the entire receiver, which includes both analog and digital signal processing operations prior to the detection procedure. A block diagram of the receiver operations is provided in Fig. 1.

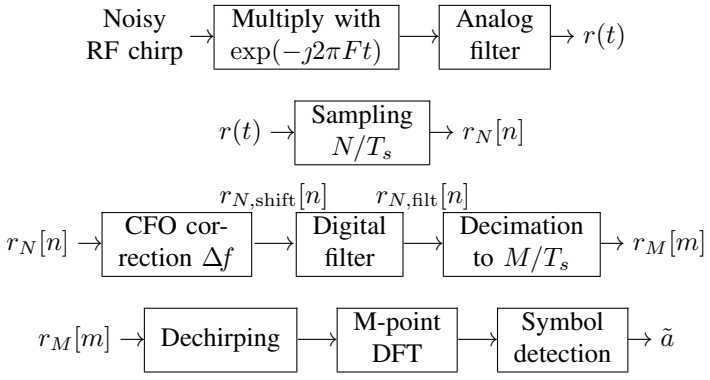

Fig. 1: Block diagram of the standard symbol detection procedure.

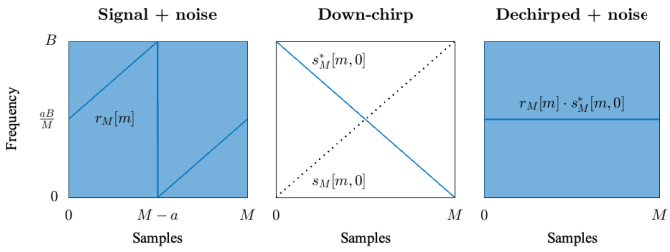

Fig. 2: Spectrograms of the waveforms and noise, relevant for non-oversampled detection. AWGN is indicated by the dark blue area.

The received RF chirp signal consists of a useful signal and additive white Gaussian noise (AWGN). The instantaneous frequency of the useful signal takes values in the range $[F_0 + \Delta F, F_0 + \Delta F + B]$, where F_0 and $F_0 + B$ denote the nominal values of the lowest and highest chirp frequencies, and ΔF represents a deviation from these nominal values. This deviation satisfies $|\Delta F| < \Delta F_{\max}$, where $F_{\max}$ indicates the maximum absolute value of the frequency deviation. As indicated in Fig. 1, the noisy RF chirp signal is multiplied with the local oscillator signal $\exp(-\jmath 2\pi F t)$, where $F = F_0 - \Delta F_{\max}$. The resulting signal is applied to an analog demodulation filter with transfer function $H_a(f)$, which satisfies

$$H_{\text{a}}(f) = \begin{cases} 1 & |f| < B + 2\Delta F_{\max} \\ 0 & |f| > B_{\text{a}} \end{cases}, \tag{6}$$

with $B_a > B + 2\Delta F_{\max}$. Note that $H_{\text{a}}(f)$ does not distort the useful signal component, irrespective of ΔF, with $|\Delta F| < \Delta F_{\max}$. This yields the complex baseband signal $r(t)$, given by

$$r(t) = s(t, a)\exp\left[\jmath 2\pi \Delta f t\right] + \omega(t), \tag{7}$$

where $\omega(t)$ denotes a complex AWGN contribution with power spectral density N_0, and where $\Delta f = \Delta F + \Delta F_{\max}$ is the CFO between the received minimum chirp frequency $F_0 + \Delta F$ and the demodulating carrier frequency F. Because of the particular selection for the local oscillator frequency F, the frequency components of $r(t)$ are essentially in the interval $[\Delta F + \Delta F_{\max}, \Delta F + \Delta F_{\max} + B]$. As $\Delta F + \Delta F_{\max} > 0$, $r(t)$ is not affected by DC offsets in the receiver. The signal $r(t)$ is sampled at a rate $N/T_s = KB$. Although time synchronization is a key challenge in the receive algorithm, we will assume perfect timing; accurate algorithms for the synchronization of FSCM signals have been presented in [50], [51]. The sampling rate satisfies $N/T_s > B + 2\Delta F_{\max} + B_a$, such that no aliasing of the noise occurs in the frequency band $[0, B + 2\Delta F_{\max}]$.

Next, frequency correction is applied to eliminate the CFO before attempting symbol detection. This is achieved by multiplying the samples $r_N[n]$ with $\exp\left[-\jmath 2\pi \frac{\Delta f T_s n}{N}\right]$. Whereas Δf should in reality be estimated, we will assume that its value is known by the receiver to simplify the description of the methods presented in this paper. At this point, the analog

demodulation filter is complemented by a digital counterpart which has passband $[0, B]$ and much steeper transition bands. The transfer function of this digital filter is approximated by the following brickwall characteristic:

$$H_{\mathrm{d}}(e^{\jmath 2\pi f T_s/N}) = \begin{cases} 1 & 0 < f < B \\ 0 & B < f < N/T_s. \end{cases} \tag{8}$$

We assume that M is sufficiently large, so that the distortion of the useful signal caused by the brickwall filter can be safely ignored. The effect of a practical digital filter on the detector performance will be illustrated in Section V. The signal at the output of the digital filter is denoted $r_{N,\mathrm{filt}}[n]$, with the subscript N referring to the number of samples per symbol interval. Next, $r_{N,\mathrm{filt}}[n]$ is downsampled by the factor K to the sampling rate B to avoid dealing with oversampled signals in the actual detection procedure. The resulting signal is indicated as $r_M[m]$, where the subscript M indicates that the number of samples per symbol interval has been reduced to M. If no noise would be present and perfect time and frequency synchronization were indeed achieved, this signal would be identical to $s_M[m, a]$, as introduced in (5).

The actual symbol detection procedure starts by dechirping the decimated waveform $r_M[m]$. This involves multiplying $r_M[m]$ with the down-chirp signal $s_M^*[m, 0]$. The latter signal is the complex conjugate of the up-chirp signal $s_M[m, 0]$ that encodes a symbol value of zero. Assuming perfect synchronization, it follows from (5) that the resulting dechirped signal is given by

$$r_{M,\mathrm{dech}}[m] = r_M[m] s_M^*[m, 0] \tag{9}$$

$$= P_s \exp\left[\jmath 2\pi a \frac{m}{M}\right] + \omega_{M,\mathrm{dech}}[m], \tag{10}$$

where the complex noise contributions $\omega_{M,\mathrm{dech}}[m]$ are statistically independent with $E[|\omega_{M,\mathrm{dech}}[m]|^2] = N_0 B P_s$. Hence, $r_{M,\mathrm{dech}}[m]$ represents a sampled orthogonal M-ary frequency shift keying (FSK) modulated signal. The dechirping of the noisy received signal is illustrated in Fig. 2, which shows the spectrograms of the relevant signals. As these spectrograms are periodic in frequency with period B, only the frequency interval $[0, B]$ is shown. Next, the discrete Fourier transform (DFT) of $r_{M,\mathrm{dech}}[m]$ is taken. As M samples describe a full symbol at this point, this transform is referred to as the M-point DFT. The resulting DFT bins are described by

$$X_M[i] = \mathcal{F}_M\{r_{M,\mathrm{dech}}[m]\} = \begin{cases} MP_s + \Omega_M[i] & i = a \\ \Omega_M[i] & i \neq a \end{cases}, \tag{11}$$

where $i \in \{0, ..., M-1\}$ denotes the DFT bin index and $\Omega_M[i]$ is the M-point DFT of $\omega_{M,\mathrm{dech}}[m]$. The complex Gaussian noise variables $\Omega_M[i]$ are statistically independent with $E[|\Omega_M[i]|^2] = MN_0BP_s$. After dechirping and Fourier transforming the received samples, the last step in the detection procedure consists of determining the DFT bin with the largest magnitude. The detected symbol $\tilde{a}$ equals the index of this largest-magnitude bin:

$$\tilde{a} = \mathrm{argmax}_{i\in\{0,\ldots,M-1\}}\big(\,|X_M[i]|\,\big). \tag{12}$$

The SNR experienced by the detector is given by the ratio of the square of the useful component from (11) to the variance of the noise contribution in (11):

$$\mathrm{SNR}_{\mathrm{det}} = \frac{(MP_s)^2}{MN_0BP_s} = \frac{E_s}{N_0}, \tag{13}$$

where $E_s = P_s T_s$ denotes the received energy per symbol interval. For a given E_s, the error performance of the standard detector for FSCM is the same as with orthogonal M-FSK modulation.

It should be noted that the symbol detection procedure is not only essential for retrieving the data payload, but also plays an important role in acquiring time and frequency synchronization. For instance, the synchronization algorithm presented in [51] involves detections for several trial values of the time delay and CFO estimates. It was found in [51] that the standard symbol detection procedure is responsible for about $98\,\%$ of the computational cost related to receiving a FSCM packet. Hence, reducing the number of computations in the detection method is expected have a strong impact on the computational efficiency and associated power consumption of the entire digital receiver / base station.

IV. Proposed Detection Methods

This paper proposes three strategies for reducing the computational cost of the FSCM detection procedure. The first strategy, outlined in Section IV-A, consists in integrating the CFO correction and the dechirping into a single operation. Next, in Section IV-B, the second strategy is proposed, which involves symbol detection without downsampling. This second strategy avoids the cost associated to downsampling, at the expense of dechirping and Fourier transforming an oversampled signal. Finally, Section IV-C describes the third strategy, which benefits from storing repeatedly used signals in memory rather than computing these signals each time they are needed. These three strategies can be combined to minimize the computational complexity.

A. Integrating Frequency Correction and Dechirping

In this strategy, we switch the order of CFO correction and digital filtering, and integrate the CFO correction with the dechirping process. First, the sampled demodulator output signal $r_N[n]$ is fed to a digital brickwall filter with passband $[\Delta f, \Delta f + B]$, and the resulting signal is downsampled by a factor K. The useful component after downsampling is represented as $s_M[m, a{+}\Delta f T_s]$, the expression of which is obtained by replacing in (5) the symbol value a by $a + \Delta f T_s$. The CFO correction (multiplication with $\exp\left[-\jmath 2\pi \frac{\Delta f T_s n}{N}\right]$) and the dechirping (multiplication with $s_M^*[m, 0]$) are combined into a single multiplication of the downsampled signal with $s_M^*[m, \Delta f T_s]$. The operation of the integrated CFO correction and dechirping is illustrated by the spectrograms from Fig. 3. These are periodic in frequency with a period equal to the considered sampling rate. As the signal obtained after the

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3344471

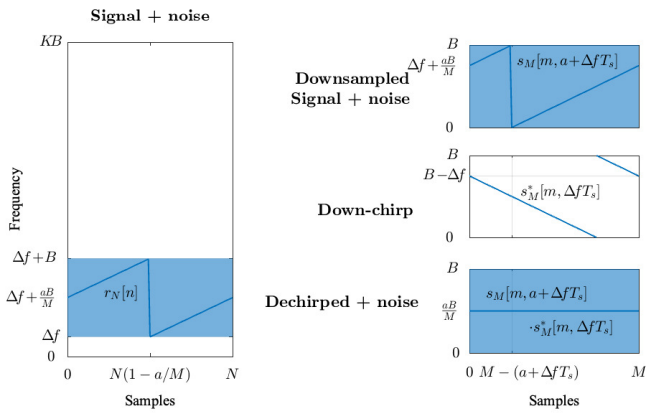

Fig. 3: Spectrograms of the waveforms and filtered noise, relevant for detection with integrated frequency correction and dechirping. Noise from $r_N[n]$ that is in $[\Delta f, \Delta f + B]$ is indicated by the dark blue area. After downsampling, this noise is projected to the dark blue area on the top right.

integrated CFO correction and dechirping is the same as $r_{M,\text{dech}}[m]$ from (9), the detection is performed by applying an M-point DFT and determining the index of the largest-magnitude DFT bin, similar to (11)-(12).

B. Detection With Oversampling

We first assume that the CFO correction and the digital filtering have been applied as in the standard detector strategy from Section III, but here we compute the DFT bins $X_M[i]$ with $i = 0, ..., M-1$ from (11) as the first M bins of the N-point DFT of the upsampled version of $r_{M,\text{dech}}[m]$. Upsampling $r_{M,\text{dech}}[m]$ by a factor K yields the signal $r_{N,\text{dech,up}}[n]$, which is defined as

$$r_{N,\text{dech,up}}[mK+k] = \begin{cases} r_{M,\text{dech}}[m] & k = 0 \\ 0 & k = 1, 2, ..., K-1 \end{cases}, \tag{14}$$

for $m \in \{0, 1, ..., M-1\}$. Denoting the N-point DFT of $r_{N,\text{dech,up}}[n]$ by $X_{N,\text{up}}[l]$, it is easily verified that

$$X_{N,\text{up}}[i + qM] = X_M[i], \tag{15}$$

for $i = 0, ..., M-1$ and $q = 1, ..., K-1$. Hence, for $i = 0, ..., M-1$, $X_M[i]$ equals the first M bins of the N-point DFT of $r_{N,\text{dech,up}}[n]$. It follows from (9) and (14) that $r_{N,\text{dech,up}}[n]$ can be obtained as $r_{N,\text{dech,up}}[n] = r_{N,\text{filt}}[n] s^*_{N,\text{up}}[n, 0]$, with $s_{N,\text{up}}[n, 0]$ denoting the upsampled version of $s_M[m, 0]$. Hence, the dechirped signal $r_{N,\text{dech,up}}[n]$ results from multiplying $r_{N,\text{filt}}[n]$ with the upsampled down-chirp signal.

A similar approach can be taken when the CFO correction and the dechirping are combined. In this case, the rate-KB signal at the output of the brickwall digital filter with passband $[\Delta f, \Delta f + B]$ is dechirped by multiplying with the upsampled version $s^*_{N,\text{up}}[n, \Delta f T_s]$ of $s^*_M[m, \Delta f T_s]$. Next, the N-point DFT of the dechirped signal is computed, and the index of the largest-magnitude bin among the first M DFT bins is determined.

An interesting interpretation of detection with oversampling follows from the identity

$$\frac{1}{K} \sum_{k=0}^{K-1} e^{\jmath 2\pi \frac{kn}{K}} = \sum_{m=0}^{M-1} \delta_{n-mK} \tag{16}$$

with $n = 0, ..., N-1$. Considering the case where the CFO correction takes place before the digital filtering, the upsampled down-chirp signal $s^*_{N,\text{up}}[n, 0]$ can be written as

$$\begin{aligned} s^*_{N,\text{up}}[n, 0] &= s^*_N[n, 0] \sum_{m=0}^{M-1} \delta_{n-mK} \\ &= \frac{1}{K} \sum_{k=0}^{K-1} s^*_N[n, 0] e^{\jmath 2\pi \frac{kn}{K}}, \end{aligned} \tag{17}$$

where $s^*_N[n, 0] = s^*(\frac{n}{N} T_s, 0)$ represents the oversampled down-chirp signal. The second line in (17) indicates that the upsampled down-chirp $s^*_{N,\text{up}}[n, 0]$ can be interpreted as $1/K$ times the sum of K frequency-shifted oversampled down-chirps. The k-th term in this sum represents the rate-KB samples of $s^*(t, 0)\exp[\jmath 2\pi k B t]$, i.e. the down-chirp signal $s^*(t, 0)$ which is shifted upward in frequency by an amount kB. Hence, the proposed detection with oversampling that uses the upsampled down-chirp signal $s^*_{N,\text{up}}[n, 0]$ is equivalent to performing dechirping by means of a sum of K frequency-shifted oversampled down-chirp signals.

The spectrograms from Fig. 4 illustrate the effect of using only the k-th term from (17) in the dechirping procedure. After dechirping, the useful signal gives rise to frequencies $\left(k + \frac{a}{M}\right) B$ and $\left(k - 1 + \frac{a}{M}\right) B$, in the first and second part, respectively, of the symbol interval, whereas the spectrogram due to the noise contribution is bounded by a parallelogram. Actually, some leakage to other frequencies is also present, but this effect is negligible for large M. The result of using all K terms from (17) is displayed in Fig. 5, which is a superposition of spectrograms of the type shown in Fig. 4, for $k \in \{0, ..., K-1\}$. Note that for $k = 0$, the contributions to the frequency band $[-B, 0]$ are aliased into the band $[(K-1)B, KB]$. The resulting spectrogram after dechirping is the periodic extension (with period B) of the spectrogram from Fig. 2.

As we just need the first M bins from the N-point DFT, only the frequency interval $[0, B]$ of the spectrogram after dechirping is relevant. We observe from Fig. 5 that the terms from (17) that contribute to $[0, B]$ are the terms with $k = 0$ and $k = 1$ (ignoring the small leakage caused by the other terms). Hence, the summation in (17) can be safely limited to these to terms. Fig. 6 illustrates the dechirping operation when only the terms with $k = 0$ and $k = 1$ are used. The part of the spectrogram of the dechirped signal, limited to the frequency interval $[0, B]$, is, when ignoring small leakage, the same as in Fig. 5.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3344471

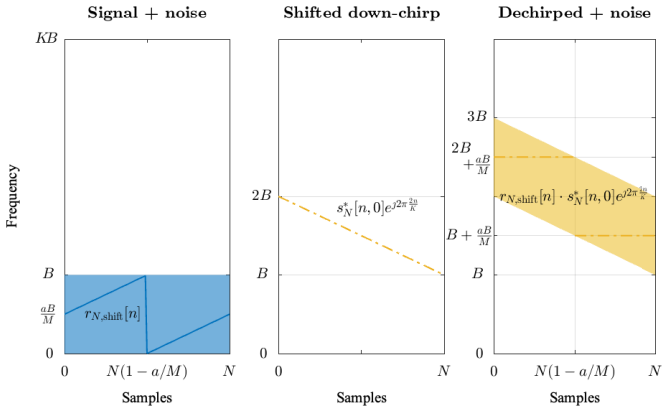

Fig. 4: Spectrograms of the (CFO-corrected) waveforms and filtered noise relevant for oversampled FSCM detection when considering the $k = 2$ term from (17) with $K = 4$. AWGN from $r_{N,\text{shift}}[n]$ that is in $[0, B]$ is indicated by the dark blue area. The relevant noise contribution after dechirping is indicated by the yellow area.

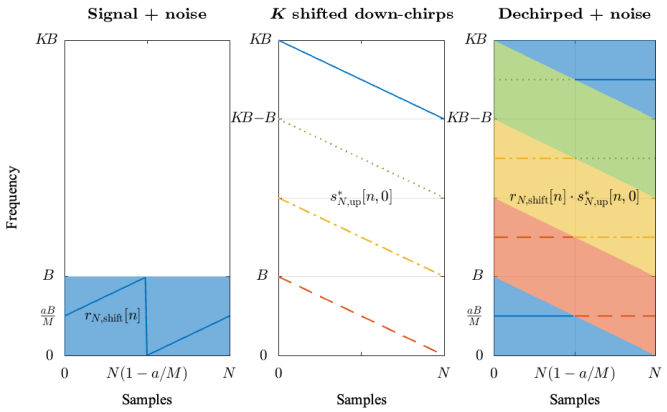

Fig. 5: Spectrograms of the (CFO-corrected) waveforms and noise relevant for detection with the upsampled down-chirp chirp $s_{N,\text{up}}^*[n,0]$. Each frequency shift of $s_N^*[n,0]$ in $s_{N,\text{up}}^*[n,0]$ is colored differently. Noise contributions after dechirping are colored based on which frequency-shifted version of $s_N^*[n,0]$ the original (dark blue) noise contribution was multiplied with.

C. Memory strategies

When assessing the computational cost of the detection methods presented above, the applied memory usage strategy may have a large impact on the results. For example, the sample values for a down-chirp that is the same in every detection attempt can be stored in memory instead of being calculated, significantly improving the computational efficiency of the detection procedure (assuming a digital system where reading one chirp signal from memory is more efficient than synthesizing its samples directly, which may require many multiplications and additions). Naturally, some algorithm versions have an inherently lower computational cost, owing to a reduced amount of required operations. However, if an inherently more expensive algorithm version

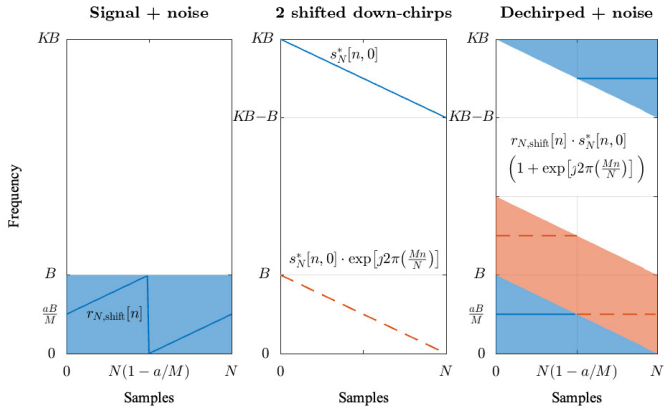

Fig. 6: Spectrograms of the (CFO-corrected) waveforms and filtered noise relevant for detection with only the first two terms of (17). Chirps and noise contributions are colored the same way as in Fig. 5.

offers more opportunities for storing often-used waveforms (or filter coefficients) in memory, it may still be more appealing. As a result, in the next section, a distinction is made between two memory strategies. In the limited-memory strategy, only the basic down-chirp signal (not shifted in frequency) and a single set of filter coefficients are stored in memory, whereas in the full-memory strategy, a set of frequency-shifted down-chirp signals and filter coefficients are stored.

Naturally, it is important to also assess the memory cost and detection accuracy impact of applying these strategies. For the limited-memory solution, a relatively small memory is sufficient to store the down-chirp signal ($N = KM$ complex samples) and the filter coefficients that describe the filter $H_\text{d}(e^{j2\pi f T_s/N})$. Assuming that this filter implementation is based on a recursive filter, the number of coefficients equals $O_\text{filter} + 1$, where O_filter denotes the filter order. As a proper CFO correction is applied to either the received samples (separate CFO correction and dechirping) or the down-chirp signal (integrated CFO correction and dechirping), applying this limited-memory usage strategy will not impact the BER performance. However, this is not the case for the strategy that stores frequency-shifted versions of the down-chirps and filter coefficients in memory, as only a finite number of frequency shifts can be realized in this way. Based on the symbol error rates presented in [43] for a signal undergoing static Doppler shifts, a frequency step of $B/(8M)$ between stored down-chirp copies should be sufficient to limit the detection performance impact of applying this strategy. This selection of the frequency step will be justified by the error performance analysis in the next section. This full-memory strategy is expected to consume a relatively large amount of memory. In practice, this amounts to N_shift times N samples for the chirps, where N_shift denotes the number of frequency shifts implemented. For a resolution of $B/(8M)$, the value of N_shift equals $16M\Delta F_\text{max}/B$. Finally, an additional $N_\text{shift}(O_\text{filter}+1)$ filter coefficients need to be stored as well.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3344471

V. Error Performance Analysis

In this section, the new FSCM detection procedure variations are investigated and compared to the standard method summarized in Section III. Each algorithm will be characterized by a label of the form X/Y-Z, where X/Y and Z refer to the signal processing and memory usage, respectively.

- X denotes whether the CFO correction and dechirping are separated (X = S) or integrated (X = I).
- Y indicates whether the dechirping is applied to the downsampled signal (Y = D) or to the oversampled signal (Y = O).
- Z refers to the memory storage strategy. Limited memory storage (of down-chirp and filter coefficients, none shifted in frequency) is indicated by Z = L, and full memory storage (of frequency-shifted down-chirps and filter coefficients) corresponds to Z = F. Note that Z = F can occur only for integrated CFO correction and dechirping, i.e. for X = I.

The sequence of signal processing operations corresponding to the different X/Y algorithms, is shown in Table I.

TABLE I: Overview of detection method operations.

Detection methods

Operations	S/D	I/D	S/O	I/O
	CFO correction		CFO correction	
	Digital filtering			
	Downsampling		Dechirping	Integrated CFO correction and dechirping
	Dechirping	Integrated CFO correction and dechirping		
	M-point Fourier transform		N-point Fourier transform	
	Determine symbol value			

A. Practical BER Analysis

Considering spreading factors of 7 ($M = 128$) and 12 ($M = 4096$), an oversampling factor K of 4, a CFO $\Delta f = \Delta F_{\max} + \Delta F$, with $\Delta F_{\max} = B/2$ and ΔF uniformly distributed in $[-B/2, B/2]$, and perfect knowledge of Δf at the receiver, Fig. 7 shows the bit error rate (BER) performance related to all detection algorithms, based on the applied memory strategy. The BER is displayed versus the SNR at the input of the receiver, where the SNR is defined as the ratio of the useful signal power to the noise power in the signal bandwidth B:

$$\mathrm{SNR}_{\mathrm{in}} = \frac{E_s/T_s}{N_0 B} = \frac{1}{M} \cdot \frac{E_s}{N_0}. \tag{18}$$

For each $\mathrm{SNR}_{\mathrm{in}}$ value, detection of random FSCM symbols was performed until at least 500 bit errors were observed. For a given $H_{\mathrm{d}}(e^{j2\pi f T_s/N})$ and a given estimate of the CFO, the considered detection algorithms are mathematically equivalent, and, therefore, yield the same BER. To provide deeper insight into the relative capabilities of each method, Table II also quantifies the difference in $\mathrm{SNR}_{\mathrm{in}}$ at a BER of 10^{-3} when compared to the ideal case.

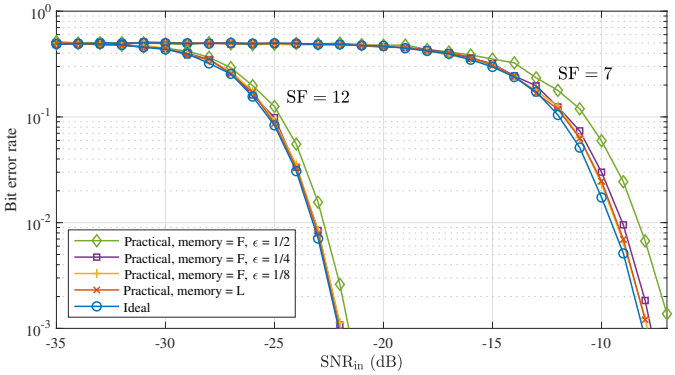

Fig. 7: Bit error rates observed for different memory strategies at spreading factor values of 7 and 12.

TABLE II: $\mathrm{SNR}_{\mathrm{in}}$ degradation at a bit error rate of 10^{-3}.

	$\mathrm{SNR}_{\mathrm{in}}$ degradation (dB)	
	SF = 7	SF = 12
Practical, memory = L	0.21	0.03
Practical, memory = F, $\epsilon = 1/8$	0.26	0.08
Practical, memory = F, $\epsilon = 1/4$	0.41	0.11
Practical, memory = F, $\epsilon = 1/2$	1.38	0.47

- The BER curve labeled *ideal* assumes perfect compensation of the CFO, has $H_{\mathrm{d}}(e^{j2\pi f T_s/N})$ represent a perfect brickwall filter so that no aliasing occurs, and ignores the suppression of the useful signal components outside the brickwall filter passband. In this case, the detection algorithms S/D, I/D, S/O and I/O yield the same BER curve as orthogonal M-FSK.
- The BER curve labeled *practical, memory = L*, corresponds to a practical implementation of $H_{\mathrm{d}}(e^{j2\pi f T_s/N})$. More specifically, forward-backward filtering [53] is performed to obtain $H_{\mathrm{d}}(e^{j2\pi f T_s/N}) = |H_{\mathrm{el}}(e^{j2\pi f T_s/N})|^2$, where $H_{\mathrm{el}}(e^{j2\pi f T_s/N})$ represents a 5th-order elliptic filter with $1\,\mathrm{dB}$ of passband ripple and $20\,\mathrm{dB}$ of stopband attenuation. The limited-memory strategy is used and perfect compensation of the CFO is assumed. The presented curve is valid for all of the considered signal processing strategies (S/D, I/D, S/O and I/O). The practical filter causes aliasing and distortion of the useful signal, yielding some BER degradation compared to orthogonal M-FSK. It is observed that this degradation is very small, especially for large M.
- The curve labeled *practical, memory = F* refers to the memory strategy F, where frequency-shifted versions of the down-chirps and filter coefficients are stored. As only a finite number of frequency shifts can be handled, CFO compensation is not perfect. Assuming that the frequency shifts of the stored signals and coefficients are equally spaced with spacing $\epsilon B/M$, a CFO compensation error occurs, which is uniformly distributed in the interval $[-\epsilon\frac{B}{2M}, \epsilon\frac{B}{2M}]$. The filter $H_{\mathrm{d}}(e^{j2\pi f T_s/N})$ is implemented in the same way as for "practical, memory L", but now using stored frequency-shifted coefficients. For a given ϵ, the BER is the same for the I/D and I/O algorithms. Fig. 7 compares the cases $\epsilon = 1/8$, $\epsilon = 1/4$ and $\epsilon = 1/2$.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3344471

The degradation compared to the case 'practical, memory = L' gets larger with increasing ϵ and decreasing SF. For $\epsilon = 1/8$ and SF ≥ 7, the degradation is negligibly small. To investigate this issue more deeply, Section V-B presents a theoretical analysis on the impact of the CFO compensation error on the detection accuracy of the system.

Overall, both Fig. 7 and Table II confirm that the strategies applied in the proposed algorithms have no significant impact on detection accuracy. Yet, it must be remembered that employing the full-memory strategy (F) requires a frequency spacing of $B/(8M)$ (or smaller) between the stored chirps for this to be true, especially at lower spreading factors, as will be confirmed in the next section as well.

B. CFO Compensation Error Analysis

To more thoroughly investigate the impact of the CFO compensation error identified in the previous section, an approximate BER analysis of the FSCM communication system which uses the full-memory strategy is presented in this section. We ignore the signal distortion introduced by the digital filter, so that the DFT bin values $X_M[i]$ corresponding to a transmitted symbol value a can be decomposed as

$$X_M[i] = MP_s A[i] + \Omega_M[i] \tag{19}$$

where the noise contribution $\Omega_M[i]$ is the same as in (11), and $A[i]$ represents the frequency leakage:

$$A[i] = \left| \frac{\sin(\pi(a+\nu-i))}{M\sin(\frac{\pi}{M}(a+\nu-i))} \right|. \tag{20}$$

The parameter ν denotes the normalized residual frequency offset, which is uniformly distributed in $\left(-\frac{\epsilon}{2}, \frac{\epsilon}{2}\right)$, with $0 < \epsilon \leq 1$ a design parameter inversely proportional to the required memory size.

A symbol error occurs when $|X_M[a]| < \max_{i\neq a}|X_M[i]|$. The symbol error probability SER is upper bounded as $\mathrm{SER} \leq \mathrm{SER_{up}}$, with $\mathrm{SER_{up}} = \sum_{i\neq a} \mathrm{PEP}[i]$; the quantity $\mathrm{PEP}[i] = \Pr[|X_M[a]| < |X_M[i]|]$ is the pairwise error probability (PEP), which is a function of $A[a]$ and $A[i]$. The PEPs can be computed according to eq. (B-21) in [54]. An upper bound on the BER is obtained as $\mathrm{BER_{up}} = (1/2)\mathrm{SER_{up}}$.

Fig. 8 shows $\mathrm{BER_{up}}$ versus ν, for $M = 2^7$ and $M = 2^{12}$; for each value of M, the operating SNR is set such that $\mathrm{BER_{up}} = 10^{-3}$ for $\nu = 0$. Also displayed is $\mathrm{BER_{up}}$ corresponding to $A[a] = \left|\frac{\sin(\pi\nu)}{M\sin(\frac{\pi\nu}{M})}\right|$ and $A[i] = 0$ for $i \neq a$, i.e. the reduction of the signal component at the correct bin index $i = a$ is taken into account, but the leakage to other bins is ignored. We observe that the leakage to other bins impacts $\mathrm{BER_{up}}$ for $M = 2^7$ (especially for large $|\nu|$), whereas the effect of this leakage is negligible for $M = 2^{12}$.

This different behavior can be explained by investigating the dependence of $A[i]$ on the bin index. Fig. 9 shows $A[i]$ for $M = 2^7$ and $\nu = 1/4$; virtually the same values of $A[i]$ would be obtained for $M = 2^{12}$, because for $M \gg |i-a|$, $A[i] \approx \left|\frac{\sin(\pi(a+\nu-i))}{\pi(a+\nu-i)}\right|$ becomes essentially independent of M. We observe that $A[i]$ takes considerable values only for i close to a (say, for $|i-a| \leq 10$). For $|i-a| > 10$, the leakage $A[i]$ can be safely ignored when computing the corresponding $\mathrm{PEP}[i]$ at the operating SNR; this is illustrated in Fig. 10, which shows $\mathrm{PEP}[i]$ with and without taking leakage into account, for $M = 2^7$ and $\nu = 0.25$; the operating SNR is set such that $\mathrm{BER_{up}} = 10^{-3}$ at $\nu = 0$. We can express $\mathrm{SER_{up}}$ as $\mathrm{SER_{up}} = (1+\delta)\mathrm{SER_{up,0}}$, where

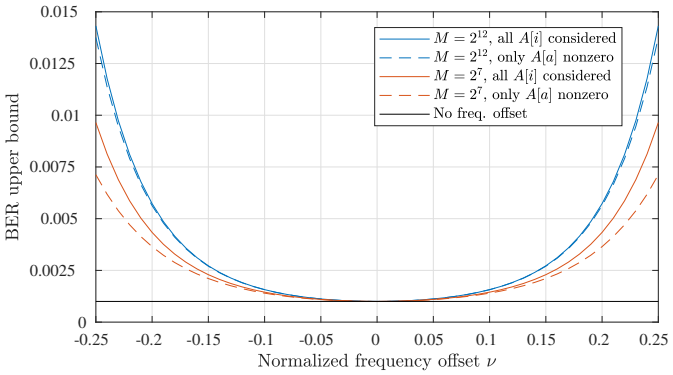

Fig. 8: Upper bound for the BER for normalized frequency offsets ν.

$$\mathrm{SER_{up,0}} = (M-1)\mathrm{PEP}_0, \tag{21}$$

$$\delta = \frac{1}{M-1}\sum_{i\neq a}\frac{\mathrm{PEP}[i] - \mathrm{PEP}_0}{\mathrm{PEP}_0}, \tag{22}$$

and PEP_0 denotes the PEP when the leakage $A[i]$ for $i \neq a$ is ignored. Hence, $\mathrm{SER_{up,0}}$ corresponds to the value of SERup when leakage is ignored (i.e. only $A[a]$ is taken into account), and δ incorporates the effect of the leakage terms $A[i]$ with $i \neq a$; the summation in (22) can be safely restricted to the interval $[a-10, a-1] \cup [a+1, a+10]$. For $M = 2^7$ and $M = 2^{12}$, we obtain $\delta = 0.3535$ and $\delta = 0.0408$, respectively, which confirms that the effect of the leakage on the error performance becomes negligible for $M = 2^{12}$.

The numerical results from Fig. 8 allow to determine $\overline{\mathrm{BER}}_{\mathrm{up}}$, which is the average of $\mathrm{BER_{up}}$ over $\nu \in \left(-\frac{\epsilon}{2}, \frac{\epsilon}{2}\right)$. From $\mathrm{BER_{up}}$ at $\nu = 0$ and $\overline{\mathrm{BER}}_{\mathrm{up}}$, we have computed the corresponding degradation in SNR, caused by the uniformly distributed residual frequency offset. This degradation is shown in Table III, for $M \in \{2^7, 2^{12}\}$ (i.e. SF = 7 and SF = 12) and $\epsilon \in \{1/8, 1/4, 1/2\}$. For $M = 2^{12}$, the degradations from Table III agree very well with the simulation results from Fig. 7 and Table II, which show negligible degradations for $\epsilon = 1/8$ and $\epsilon = 1/4$, and a degradation of about 0.47 dB for $\epsilon = 1/2$. However, for $M = 2^7$, the simulations from Fig. 7 and Table II show degradations which are considerably larger than those from Table III. This discrepancy for $M = 2^7$ results from ignoring in the BER analysis the distortion caused by the digital filter: as the spectral width of the FSCM signal increases beyond B with decreasing M, the corresponding signal distortion is smaller for $M = 2^{12}$ than for $M = 2^7$.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3344471

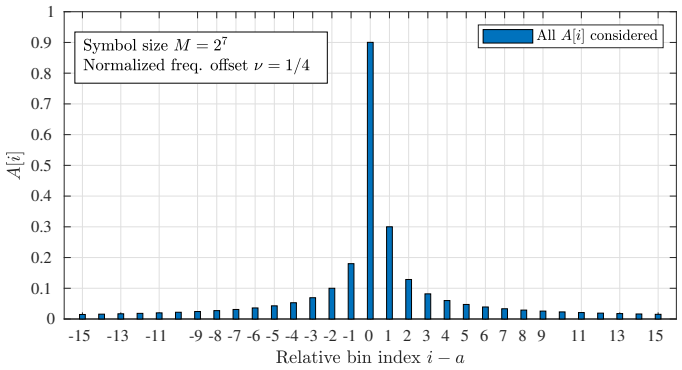

Fig. 9: DFT bin signal magnitudes for $M = 2^7$ and $\nu = 1/4$.

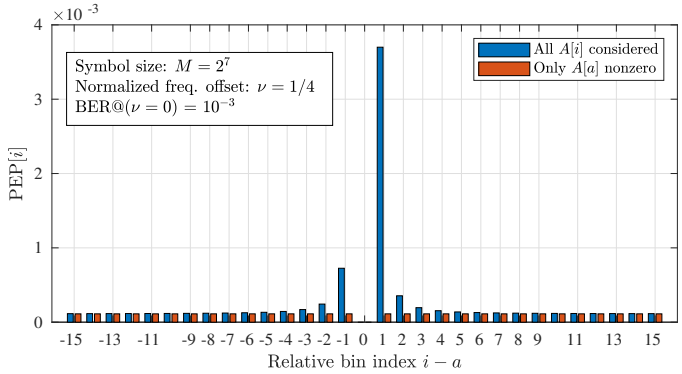

Fig. 10: Pairwise error probabilities as a function of the relative bin index $i - a$.

VI. Computational complexity

This section provides an overview of the computational costs associated to the detection algorithm versions outlined in Table I. Below, the computational impact of the different operations applied in each detection method is estimated and analyzed, providing a detailed overview of the differences between the algorithm versions. Additionally, the average computation times observed for each symbol detection procedure are also presented. These timing measures provide a more general view on the computational cost differences between the detection methods under study.

A. Cost of Individual Operations

To provide a detailed analysis first, the computational cost of each individual operation in every procedure is estimated. More specifically, using a spreading factor of 7, each operation in the detection procedure was executed 5×10^5 times in Matlab, running on a single thread on an Apple Silicon M1 processor with 16 GB of memory, while measuring the elapsed computation time. The obtained value was then divided by the time it takes to execute the sum of two (random) floating point variables the same amount of times. Finally, the resulting ratio was normalized through division by M (remind that $M = 2^{\mathrm{SF}}$). Hence, a normalized ratio x indicates that the considered operation, when processing a signal segment of duration $1/B$, is equivalent to x two-number floating point additions. This normalized ratio is shown in Table IV as a function of K,

TABLE III: Theoretical SNR degradation at a bit error rate of 10^{-3}.

	SNR degradation (dB)	
	SF = 7	SF = 12
$\epsilon = 1/8$	0.0215	0.0199
$\epsilon = 1/4$	0.0919	0.0853
$\epsilon = 1/2$	0.4711	0.4358

for the various operations. Under the assumption that reading from (cache) memory is significantly faster than synthesizing the signals on the spot, the cost associated to reading from memory is set to zero in in Table IV. Also ignored are some operations, such as the computation of $O_{\mathrm{filter}} + 1$ (possibly frequency-shifted) filter coefficients, that do not depend on M or K and, therefore, are much cheaper than the operations listed in Table IV.

Let us first compare the algorithms with separate CFO correction and dechirping (X = S) to those with integrated CFO correction and dechirping (X = I). For both I/D and I/O, relatively large computational cost savings are achieved by eliminating the separate frequency shifting operation, whereas the cost of dechirping remains the same as with S/D and S/O, respectively, which is as expected. Next, we compare the algorithms with downsampled dechirping (Y = D) to those with oversampled dechirping (Y = O). Here, the application of the dechirping and the FFT on the oversampled rather than the downsampled signal reduces the computational cost, because the cost of downsampling plus processing the downsampled signal in the S/D and I/D algorithms turns out to be larger than the cost of processing the oversampled signal in the S/O and I/O algorithms.

In general, Table IV shows that a meaningful computational cost reduction can be achieved when memory is actively used, confirming the assumption that reading from memory is usually advantageous in comparison to direct chirp synthesis. The total computational cost's unequal dependency on the oversampling factor is also clearly visible. For example, to their advantage, the standard S/D method and the new I/D method both feature a fixed cost (w.r.t. K) for detecting the symbol after downsampling, while the cost of frequency correction (when applied), filtering and downsampling all depend on the oversampling factor. In contrast, for S/O and I/O nearly all of the operations are more expensive when a higher oversampling factor is employed. The limited memory strategy does not seem to provide much added value to these methods as it is only useful when CFO correction and dechirping are separate. However, when the full memory strategy is chosen, the true added value of oversampled dechirping is uncovered. As a result, the combination of the I/O method with this memory usage strategy is estimated to be the most efficient detection method of all those under study, featuring a total cost of $146K + 2$, with the I/D-F method following closely behind at a total cost of $190K + 28$. When $K = 4$, the cost of I/O-F is estimated to be about $45\,\%$ lower than S/D-L. As shown in Fig. 7, this substantial cost reduction comes with a negligible BER performance degradation when taking ϵ = 1/8. For I/D-F, the cost reduction is lower w.r.t. S/D-L, at $25\,\%$. Finally, the total cost for S/O-L amounts to $216K + 2$, which for $K = 4$

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3344471

TABLE IV: Computational cost relative to one floating point operation for different detection methods. Strategies feature either separated (S/Y) or integrated (I/Y) frequency correction and downsampled (X/D) or oversampled (X/O) dechirping. The studied strategies are combined with either limited memory usage (-L) or full memory usage (-F).

	S/D-L	I/D-F	S/O-L	I/O-F
CFO correction	$70K$		$70K$	
Digital filtering	$120K$	$120K$	$120K$	$120K$
Downsampling	$70K$	$70K$		
Dechirping	10	10	$10K$	$10K$
FFT	≈ 16	≈ 16	$\approx 16K$	$\approx 16K$
Determine symbol value	2	2	2	2
Total	$260K+28$	$190K+28$	$216K+2$	$146K+2$

K=oversampling factor

is about 20 % less than for the standard S/D-L algorithm. In summary, the new algorithms are estimated to be about 20 % to 45 % cheaper than the standard S/D detection procedure, while featuring practically identical BER performance (see Fig. 7).

B. Overall Cost per Detection

To provide a slightly more relatable overview of the computation times needed for symbol detection, absolute timings were also collected on the same computation platform as described above. The times needed to perform 10^3 symbol detections were registered for all algorithm versions, employing an oversampling factor K of 4 where applicable. The results of these measurements are shown in Fig. 11, which displays the computation times on a logarithmic scale versus SF. It appears that the Matlab overhead related to function calls and variable management is quite significant for lower spreading factors as lowering the SF yields diminishing returns for reducing the associated computation time. However, for higher SF values, the expected linear relation of log(computation time) versus SF is discernible.

Overall, the results in Fig. 11 support the main finding from Section VI-A, which is the superiority of integrated CFO correction and dechirping with full memory usage over separate CFO correction and dechirping with limited memory usage. While Table IV indicates that for SF $=$ 7, the S/D and I/D algorithms are outperformed by the S/O and I/O algorithms, respectively, it is observed from Fig. 11 that this trend extends to the case SF $<$ 8, but is reversed for SF $>$ 8. Still, it should be noted that such minor advantages or disadvantages can tip the scales in either direction. For example, when using a slightly less efficient downsampling method (more specifically: checking sample indices against multiples of K using a modulo operation instead of calculating these indices up front), the I/D-F method is significantly impacted, performing worse than I/O-F for all spreading factors. (Note that the actual data related to this sub-optimal implementation of I/D-F are omitted from this paper.) When compared to the limited-memory standard method (S/D-L), the computational efficiency gain observed for I/D-F and I/O-F ranges from 19 % to 36 %, depending on the SF; with an average of 29.5 %. These numbers are slightly lower than the 25 % to 45 % baseline estimated in Section VI-A, in part because the entries in Table IV do not include the Matlab overhead mentioned above. In any case, it is generally clear that the newly proposed methods are significantly more efficient than the standard one, especially when sufficient high-speed (cache) memory is available.

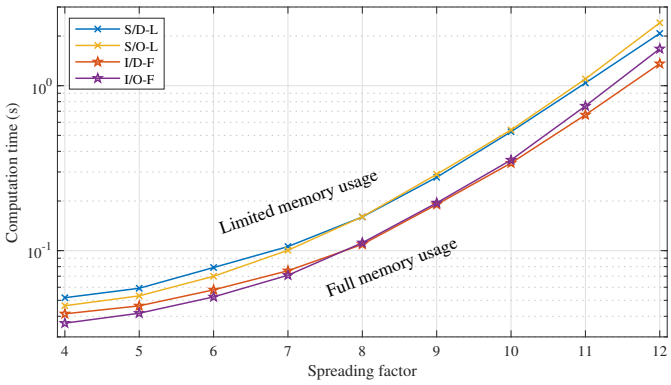

Fig. 11: Computation times for 10^3 trial detections at a wide range of spreading factors and with $K = 4$.

VII. Conclusion

In this paper, novel symbol detection methods for handling frequency shift chirp symbols were proposed and compared, both in terms of detection accuracy and computational cost. Several strategies, such as integrating frequency correction in the dechirping procedure and omitting the downsampling operation by processing the received samples with upsampled down-chirps, have been combined. Overall, the most efficient versions of the newly proposed methods feature significant computational efficiency gains when compared to the current state-of-the-art, while featuring comparable bit error rate performance. In fact, when paired with an adequate memory usage strategy, computational efficiency gains up to 36 % were measured for the most advanced of the proposed methods. When adopted on a suitable software defined radio platform, this significant improvement may drastically decrease its power consumption.

References

[1] LoRa Alliance, “LoRa,” https://www.lora-alliance.org/.
[2] SigFox Foundation, “SigFox,” https://www.sigfox.com/en.
[3] GSM Association, “NarrowBand - Internet of Things (NB-IoT),” https://www.gsma.com/iot/narrow-band-internet-of-things-nb-iot/.
[4] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, “A Study of LoRa: Long Range & Low Power Networks for the Internet of Things,” *Sensors*, vol. 16, no. 9, 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/9/1466

[5] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa Low-Power Wide-Area Networks Scale?" in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, New York, NY, USA, 2016, pp. 59–67. [Online]. Available: http://doi.acm.org/10.1145/2988287.2989163

[6] U. Noreen, A. Bounceur, and L. Clavier, "A Study of LoRa Low Power and Wide Area Network Technology," in *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, May 2017, pp. 1–6.

[7] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-Range Communications in Unlicensed Bands: the Rising Stars in the IoT and Smart City Scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.

[8] L. Gregora, L. Vojtech, and M. Neruda, "Indoor Signal Propagation of LoRa Technology," in *2016 17th International Conference on Mechatronics - Mechatronika (ME)*, Dec 2016, pp. 1–4.

[9] P. Neumann, J. Montavont, and T. Noël, "Indoor Deployment of Low-Power Wide Area Networks (LPWAN): A LoRaWAN Case Study," in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016, pp. 1–8.

[10] L. H. Trinh, V. X. Bui, F. Ferrero, T. Q. K. Nguyen, and M. H. Le, "Signal Propagation of LoRa Technology Using for Smart Building Applications," in *2017 IEEE Conference on Antenna Measurements Applications (CAMA)*, Dec 2017, pp. 381–384.

[11] J. Haxhibeqiri, A. Karaağaç, F. Van den Abeele, W. Joseph, I. Moerman, and J. Hoebeke, "LoRa Indoor Coverage and Performance in an Industrial Environment: Case Study," 2017, pp. 1–8.

[12] J. Petäjäjärvi, K. Mikhaylov, R. Yasmin, M. Hämäläinen, and J. Iinatti, "Evaluation of LoRa LPWAN Technology for Indoor Remote Health and Wellbeing Monitoring," *International Journal of Wireless Information Networks*, vol. 24, no. 2, pp. 153–165, June 2017.

[13] T. Ameloot, P. Van Torre, and H. Rogier, "A Compact Low-Power LoRa IoT Sensor Node with Extended Dynamic Range for Channel Measurements," *Sensors*, vol. 18, no. 7, p. 2137, Jul 2018. [Online]. Available: http://dx.doi.org/10.3390/s18072137

[14] T. Ameloot, P. Van Torre, and H. Rogier, "LoRa Indoor Performance: an Office Environment Case Study," in *2018 International Applied Computational Electromagnetics Society Symposium - China (ACES)*, July 2018.

[15] ——, "Indoor Body-to-Body LoRa Link Characterization," in *2019 9th IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications*, September 2019.

[16] J. Petäjäjärvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the Coverage of LPWANs: Range Evaluation and Channel Attenuation Model for LoRa Technology," in *2015 14th International Conference on ITS Telecommunications (ITST)*, Dec 2015, pp. 55–59.

[17] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti, "Performance of a Low-Power Wide-Area Network Based on LoRa Technology: Doppler Robustness, Scalability, and Coverage," *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, 2017. [Online]. Available: https://doi.org/10.1177/1550147717699412

[18] O. Iova, A. L. Murphy, G. P. Picco, L. Ghiro, D. Molteni, F. Ossi, and F. Cagnacci, "LoRa from the City to the Mountains: Exploration of Hardware and Environmental Factors," in *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*. USA: Junction Publishing, 2017, pp. 317–322. [Online]. Available: http://dl.acm.org/citation.cfm?id=3108009.3108091

[19] T. Ameloot, P. Van Torre, and H. Rogier, "LoRa Base-Station-to-Body Communication With SIMO Front-to-Back Diversity," *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 1, pp. 397–405, 2021.

[20] ——, "Experimental Parameter Optimization for Adaptive LoRa Modulation in Body-Centric Applications," in *2020 14th European Conference on Antennas and Propagation (EuCAP)*, 2020.

[21] M. Cattani, C. A. Boano, and K. Römer, "An Experimental Evaluation of the Reliability of LoRa Long-Range Low-Power Wireless Communication," *Journal of Sensor and Actuator Networks*, vol. 6, no. 2, 2017. [Online]. Available: http://www.mdpi.com/2224-2708/6/2/7

[22] C. A. Boano, M. Cattani, and K. Römer, "Impact of Temperature Variations on the Reliability of LoRa - An Experimental Evaluation," in *Proceedings of the 7th International Conference on Sensor Networks - Volume 1*. SciTePress, 2018, pp. 39–50.

[23] T. Ameloot, P. Van Torre, and H. Rogier, "Periodic LoRa Signal Fluctuations in Urban and Suburban Environments," in *2019 13th European Conference on Antennas and Propagation (EuCAP)*, 2019.

[24] ——, "Variable Link Performance due to Weather Effects in Long-Range, Low-Power Wireless Sensor Networks," *submitted to MDPI Sensors*.

[25] O. Georgiou and U. Raza, "Low Power Wide Area Network Analysis: Can LoRa Scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, April 2017.

[26] K. Mikhaylov, J. Petäjäjärvi, and T. Haenninen, "Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology," in *22th European Wireless Conference*, May 2016, pp. 1–6.

[27] B. Reynders, W. Meert, and S. Pollin, "Range and Coexistence Analysis of Long Range Unlicensed Communication," in *2016 23rd International Conference on Telecommunications (ICT)*, 2016, pp. 1–6.

[28] L. Vangelista, "Frequency Shift Chirp Modulation: The LoRa Modulation," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1818–1821, Dec 2017.

[29] M. Chiani and A. Elzanaty, "On the LoRa Modulation for IoT: Waveform Properties and Spectral Analysis," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8463–8470, 2019.

[30] B. Sikken, "Decoding LoRa," https://revspace.nl/DecodingLora.

[31] J. Blum, "LoRa SDR Project," https://github.com/myriadrf/LoRa-SDR.

[32] RTL-SDRangelove, "Github Project Page," https://github.com/hexameron/rtl-sdrangelove.

[33] A. Marquet, N. Montavont, and G. Z. Papadopoulos, "Towards an SDR Implementation of LoRa: Reverse-Engineering, Demodulation Strategies and Assessment over Rayleigh Channel," *Computer Communications*, vol. 153, pp. 595–605, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366419314665

[34] 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.

[35] J. Tapparel, O. Afisiadis, P. Mayoraz, A. Balatsoukas-Stimming, and A. Burg, "An Open-Source LoRa Physical Layer Prototype on GNU Radio," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2020, pp. 1–5.

[36] 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.

[37] M. Xhonneux, J. Louveaux, and D. Bol, "Implementing a LoRa Software-Defined Radio on a General-Purpose ULP Microcontroller," in *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, 2021, pp. 105–110.

[38] H. Poveda, K. Navarro, F. Merchan, E. Ramos, and D. Gonzalez Gonzalez, "A Software Defined Radio-Based Prototype for Wireless Metrics Studies in IoT Applications," *Wireless Personal Communications*, 2021.

[39] A. Lavric, A. I. Petrariu, E. Coca, and V. Popa, "LoRa Traffic Generator Based on Software Defined Radio Technology for LoRa Modulation Orthogonality Analysis: Empirical and Experimental Evaluation," *Sensors*, vol. 20, no. 15, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/15/4123

[40] A. A. Doroshkin, A. M. Zadorozhny, O. N. Kus, V. Y. Prokopyev, and Y. M. Prokopyev, "Experimental Study of LoRa Modulation Immunity to Doppler Effect in CubeSat Radio Communications," *IEEE Access*, vol. 7, pp. 75 721–75 731, 2019.

[41] Fernandez, Lara and Ruiz de Azua Ortega, Joan and Calveras, Anna and Camps, Adriano, "Assessing LoRa for Satellite-to-Earth Communications Considering the Impact of Ionospheric Scintillation," *IEEE Access*, vol. 8, pp. 165 570–165 582, 01 2020.

[42] S. Hassayoun, S. Lahouar, and K. Besbes, "SDR Bridge for a Secure Wireless Sensor Network (WSN)," 06 2020, pp. 1–5.

[43] T. Ameloot, M. Moeneclaey, P. Van Torre, and H. Rogier, "Characterizing the Impact of Doppler Effects on Body-Centric LoRa Links with SDR," *Sensors*, vol. 21, no. 12, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/12/4049

[44] C. Demeslay, P. Rostaing, and R. Gautier, "Simple and Efficient LoRa Receiver Scheme for Multi-Path Channel," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[45] P. Robyns., P. Quax., W. Lamotte., and W. Thenaers., "A Multi-Channel Software Decoder for the LoRa Modulation Scheme," in *Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security - Volume 1*. SciTePress, 2018, pp. 41–51.

[46] 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.

[47] A. A. Kherani and K. M. P. Maurya, "Improved Packet Detection in LoRa-Like Chirp Spread Spectrum Systems," in *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2019, pp. 1–4.

[48] T. Elshabrawy, P. Edward, M. Ashour, and J. Robert, "On the Different Mathematical Realizations for the Digital Synthesis of LoRa-Based Modulation," in *European Wireless 2019; 25th European Wireless Conference*, 2019, pp. 1–6.

[49] S. Elzeiny, P. Edward, and T. Elshabrawy, "LoRa Performance Enhancement through List Decoding Technique," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.

[50] M. Xhonneux, O. Afisiadis, D. Bol, and J. Louveaux, "A Low-Complexity LoRa Synchronization Algorithm Robust to Sampling Time Offsets," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3756–3769, 2022.

[51] T. Ameloot, H. Rogier, M. Moeneclaey, and P. Van Torre, "LoRa Signal Synchronization and Detection at Extremely Low Signal-to-Noise Ratios," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8869–8882, 2022.

[52] T. Elshabrawy and J. Robert, "Closed-Form Approximation of LoRa Modulation BER Performance," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1778–1781, 2018.

[53] J. Smith, *Introduction to Digital Filters with Audio Applications - Chapter 10: Filters Preserving Phase*. Charleston: Booksurge, 2012.

[54] J. Proakis and M. Salehi, *Digital Communications*. New York: McGraw-Hill Education, 2007.