LoRa Signal Synchronization and Detection at Extremely Low Signal-to-Noise Ratios

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

***Abstract*—In recent years, LoRa has been deployed in countless Internet of Things (IoT) applications across the globe. However, as LoRa is a proprietary technology, research into its physical-layer performance has been challenging. Implementing LoRa on software-defined radio (SDR) platforms yields valuable insight into the physical layer of the LoRa standard and paves the way for improvements in packet reception capabilities for LoRa receivers. This article presents an independently developed packet reception algorithm, which drastically improves the physical performance of LoRa communication links. The advanced signal presence detection, synchronization, and symbol detection strategies are shown to significantly increase packet reception ratios in extremely adverse noise conditions. Multiple algorithm variations are presented and compared in terms of bit error rate (BER) performance and computational cost. In comparison to a theoretical system with perfect channel state information, the simulated BER performance of the best performing algorithm only requires an increase of 1.6 dB in signal-to-noise ratio (SNR) to exhibit the same performance. Finally, SDR implementations of the algorithms exhibit average SNR performance gains up to 4.7 dB when compared to commercially available hardware.**

***Index Terms*—Detection, Internet of Things (IoT), LoRa, software-defined radio (SDR), wireless sensor networks.**

I. Introduction

As OUR world becomes increasingly connected, wireless communication technologies are continuously being conceived and improved. For long-range wireless applications, low-power wide-area network (LPWAN) technologies, such as SigFox [1], NB-IoT [2], and LoRa [3], lead the way in terms of global deployment and adoption. In recent years, a large number of researchers have evaluated the effectiveness of LoRa. Several papers [4]–[7] provide excellent overviews of how LoRa technology works. These are supplemented by more in-depth, theoretical reviews and investigations [8], [9]. The applicability of LoRa networks has been assessed in a wide range of indoor [10]–[16] and outdoor [17]–[21] measurement campaigns. These papers show that LoRa achieves excellent wireless performance and highlight a large range of opportunities. However, LoRa network scalability has also been scrutinized on several occasions [5], [22]–[24]. In light of these investigations, the limited available spectrum in industrial, scientific, and medical (ISM) radio bands is an aggravating factor. Additionally, the influence of temperature and atmospheric effects on the quality of outdoor LoRa links is still under study [25]–[27].

Seeing this level of research interest in such a heavily commercialized branch of wireless technology, it is no surprise that multiple efforts have been made to implement LoRa modulation on software-defined radio (SDR) platforms. From online, cooperative efforts [28]–[30] to actual scientific papers [31]–[33], a multitude of solutions exist that facilitate SDR-based LoRa communication. Moreover, through the application of SDR technology, some LoRa implementations also feature different ways of improving communication performance [34]–[37]. Additionally, a number of recent publications show how unchaining LoRa modulation from its traditional host of proprietary transceivers is a valuable way of exploring its capabilities in a diverse set of specific application environments [38]–[42]. Finally, using SDR technology also enables the user to employ other communication technologies and modulation techniques to diversify link options and serve multiple types of wireless networks [43]–[45].

A. Related Work

Recently, a number of papers have been published that present SDR implementations of LoRa modulation. In [31], conventional LoRa modulation and detection architectures are described in detail. Data manipulation steps, such as channel coding, whitening, and interleaving, are presented along with signal-to-noise ratio (SNR) performance assessments for different LoRa spreading factors (SFs). In [32], a concise overview of SDR implementation aspects of LoRa is given. The authors analyze the impact of carrier frequency offset (CFO) and sampling frequency offset errors on communication performance and demonstrate a number of methods for correcting these offsets. In addition, [33] presents an open-source LoRa physical-layer prototype for the SDR ecosystem GNU Radio.

Whereas [31]–[33] present SDR-based LoRa implementations that are compatible with actual LoRa hardware, other

Manuscript received April 21, 2021; revised September 10, 2021; accepted September 25, 2021. Date of publication October 1, 2021; date of current version May 23, 2022. This work was supported in part by the Research Foundation—Flanders (FWO) through the "MUlti-SErvice WIreless NETwork," FWO-FRS Excellence of Science—EOS Project. *(Corresponding author: Thomas Ameloot.)*

Thomas Ameloot, Hendrik Rogier, and Patrick Van Torre are with the IDLab, Department of Information Technology, Ghent University—imec, 9052 Ghent, Belgium (e-mail: thomas.ameloot@ugent.be; hendrik.rogier@ugent.be; patrick.vantorre@ugent.be).

Marc Moeneclaey is with the Department of Telecommunications and Information Processing, Ghent University, 9000 Ghent, Belgium (e-mail: marc.moeneclaey@ugent.be).

Digital Object Identifier 10.1109/JIOT.2021.3117039

implementations exist that also make an effort to improve wireless performance [34]–[37]. For example, in [34], a more advanced software-based detector is presented in moderate detail and all steps that need to be performed to access the raw data encapsulated in the packet are considered. An interesting clock drift correction technique is presented and robustness against frequency errors is achieved. However, compared to commercial hardware, the packet error performance from [34] is observed to be considerably more sensitive to noise. In [35], improved packet capture capabilities are presented for different variations of an improved preamble detection algorithm. This algorithm relaxes a number of requirements for detecting LoRa packets, which results in better signal presence detection performance. Similarly, Kherani and Maurya [36] also considered signal presence detection in high LoRa interference conditions. Packets are detected based on an energy detection approach, which leads to correct signal presence detection at a signal-to-interference ratio (SIR) of -6 dB. However, in [36], no dechirping is performed. Hence, at lower SNRs, performance is expected to be impacted heavily. More LoRa performance improvement methods are theoretically described in [37]. This article first presents an alternative mathematical representation of LoRa signals which ensures intersymbol phase continuity across different LoRa symbols. As stated, this enables coherent detection of LoRa signals. Relevant expressions for the bit error rate (BER) are composed, yielding a moderate SNR performance improvement of 0.7 dB. Additionally, a method is proposed to enhance network capacity by constructing additional orthogonal dimensions for LoRa signaling. Unfortunately, the methods presented in [37] have not yet been tested in practical deployments.

Yet, even when receiver performance is not altered, SDR-based LoRa implementations are still very valuable as they enable much more thorough analyses of the wireless performance of LoRa technology. For example, Poveda *et al.* [38] presented a low-cost, configurable and portable SDR-based system for characterizing interference in the context of Internet of Things (IoT) applications. A demonstration of this system is presented using LoRa signals. Another example is found in [39]. In this article, LoRa orthogonality is empirically analyzed using an SDR platform. More diverse applications also exist, for example, in [40] and [41], where LoRa for satellite-to-Earth communication is mimicked using SDR setups. Furthermore, in [42], SDR technology is used to add an additional encryption layer to the LoRaWAN communication stack. However, it should be mentioned that this encryption layer is added on top of the physical layer. Hence, LoRa modulation and detection are performed by conventional LoRa transceivers.

Finally, LoRa implementations on SDR are also very useful for multiservice IoT applications. For example, [43] presents a reconfigurable IoT gateway based on an SDR platform, which also facilitates communication using WiFi, Bluetooth Low Energy, and ZigBee. Similarly, in [44], a multiservice emergency aid platform is implemented that facilitates communication using GSM, WiFi, and LoRa. Ultimately, we mention that [45] also presents a programmable gateway architecture for multiservice IoT applications, which also includes LoRa capabilities.

B. Contributions

This article presents an independently developed LoRa packet reception algorithm for application on SDR. The presented algorithm intends to further improve the physical-layer performance of SDR-based LoRa receivers through a number of advanced synchronization and detection procedures. In contrast to existing research, there is a strong emphasis on the impact of accurate time and frequency synchronization on the BER and packet reception ratio (PRR) performance of the LoRa links. As a consequence, the observed SNR performance gains are much larger than those presented in other papers that describe enhanced LoRa receivers. In fact, through exhaustive synchronization efforts, BERs are observed that nearly approach the performance of a link with perfect channel state information (CSI). As a result of the aforementioned synchronization efforts, the system is also highly tolerant to CFOs, e.g., resulting from static Doppler shifts or a hardware-related frequency mismatch. This article also features an in-depth comparison of the preamble detection performance for different detection settings at very low SNRs and provides a number of illustrations of LoRa waveforms throughout the synchronization procedure. Compared to commercial hardware, the SDR implementation of the presented algorithm may be used to enhance either the range or reliability of existing LoRa networks, given the excellent SNR improvements featured here. Finally, in contrast to the related literature, the LoRa receiving algorithm presented in this work is described in such detail that it can be directly simulated or implemented on any SDR platform.

C. Contents

This article is structured as follows. First, LoRa modulation and symbol detection is discussed from a signal processing perspective in Section II. Descriptions of LoRa chirps and symbols are presented in Section II-A. The procedure to detect these symbols is presented in Section II-B. For reference purposes, BER results are presented for a channel with additive white Gaussian noise (AWGN) in Section II-C. In Section III, the basic structure of LoRa packets is briefly summarized. Section IV describes the LoRa receiving algorithm in its entirety, introducing signal presence detection, several synchronization steps, and an advanced symbol detection procedure. Throughout Section IV, a number of key algorithm design choices are investigated in detail and compared to alternatives in order to achieve the best possible performance. Section V illustrates the simulated performance of the proposed LoRa receiving algorithm. A number of algorithm variations are compared to show the tradeoffs that can be made concerning BER performance and computational cost. In the penultimate section of this article, an actual SDR implementation of the LoRa receiving algorithm presented in Section IV is compared to commercial LoRa hardware through the packet reception rates observed in both systems. Finally, a conclusion completes this article in Section VII.

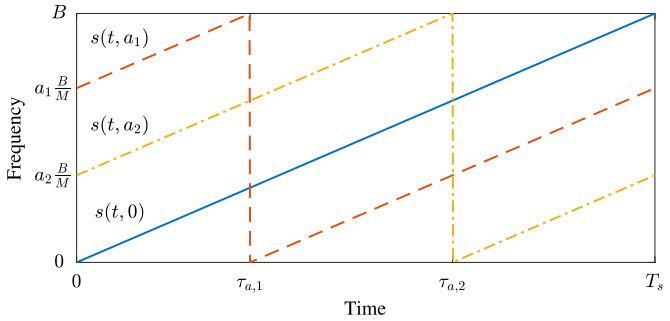


Fig. 1. Instantaneous frequency of a pure up-chirp $s(t, 0)$ and two LoRa waveforms for arbitrary symbols $s(t, a_1)$ and $s(t, a_2)$.

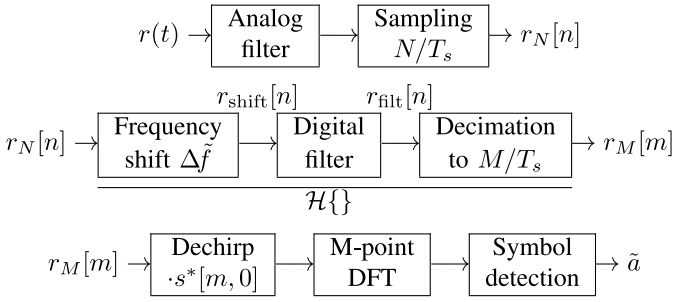


Fig. 2. Block diagram of the proposed LoRa receiver and symbol detector.

II. LoRa Modulation and Detection

Basic mathematical descriptions of LoRa modulation can be found in [8], [9], [37], and [46]. This article mostly adheres to the notation used in [9]. Additionally, oversampling is applied, and modifications are made to the original detection procedure for detecting LoRa symbols without oversampling to accommodate for these oversampled LoRa signals.

A. Data Modulation

As illustrated in Fig. 1, LoRa modulation is based on linear frequency chirps. A LoRa symbol is defined in the continuous time interval $[0, T_s]$, where it is described by an instantaneous frequency in the range $[0, B]$. The duration T_s and the frequency swing B of the chirp waveform are related through the spreading factor (SF), which itself is related to the size M of the symbol alphabet through $M = 2^{\text{SF}} = BT_s$. For a symbol $a \in \{0, 1, \ldots, M-1\}$, the instantaneous baseband frequency f is given by

$$f(t, a) = \frac{a}{M} B + \frac{t}{T_s} B \,(\text{mod}\, B) \qquad t \in [0, T_s]. \tag{1}$$

As shown in the above equation, the initial frequency of the waveform is related to the symbol value through $f(0, a) = (a/M)B$. The modulo operation can be substituted by introducing the time instant τ_a, which is related to the symbol value through $\tau_a = (1 - a/M)T_s$. This represents the time at which the instantaneous frequency reaches B. At this point, f undergoes a reset to zero. The instantaneous frequency of the LoRa signal can be rewritten as

$$f(t, a) = \begin{cases} \frac{a}{T_s} + \frac{M}{T_s^2} t, & 0 < t < \tau_a \\ \frac{a-M}{T_s} + \frac{M}{T_s^2} t, & \tau_a < t < T_s. \end{cases} \tag{2}$$

The LoRa signal, with symbol energy E_s, is described by

$$s(t, a) = \sqrt{\frac{E_s}{T_s}} \exp\big[\, \jmath\, \phi(t, a)\,\big]. \tag{3}$$

The instantaneous phase of this signal is given by

$$\begin{aligned} \phi(t, a) &= 2\pi \int_0^t f(\tau, a)\, \mathrm{d}\tau \\ &= \begin{cases} 2\pi\left(a\frac{t}{T_s} + \frac{M}{2}\frac{t^2}{T_s^2}\right), & 0 < t < \tau_a \\ 2\pi\left(\left(1 - \frac{t}{T_s}\right)(M - a) + \frac{M}{2}\frac{t^2}{T_s^2}\right), & \tau_a < t < T_s. \end{cases} \end{aligned} \tag{4}$$

B. Data Detection

Consider the received symbol $r(t)$, obtained through RF-to-baseband demodulation by an SDR receiver, and described by

$$r(t) = s(t) \cdot \exp\big[\, \jmath 2\pi\, \Delta f\, t\,\big]. \tag{5}$$

For this symbol, a CFO is present and described by Δf. As shown in Fig. 2, this signal is filtered by an analog anti-aliasing filter and sampled by an analog-to-digital converter (ADC) at a sample rate of N/T_s with $N = KM$ for $K \in \mathbb{Z}^+$. In this expression, K denotes the oversampling factor applied w.r.t. the minimum sampling rate of M/T_s. This sampling operation produces a sequence of N received samples, denoted by $r_N[n]$ for $n = \{0, 1, \ldots, N-1\}$. As demonstrated in Fig. 2, which shows the block diagram for the SDR receiver and LoRa symbol detector, several operations are performed to manipulate the received samples before actual symbol detection is attempted. For example, by applying oversampling, a possible frequency shift, which is unknown when sampling the LoRa signal, can be corrected in postprocessing by the detector. Additionally, oversampling allows us to digitally filter the received signal with a more brickwall-like transfer function than provided by the analog filters in the front end of the SDR receiver. After this operation, the signal is decimated to a sample rate of M/T_s to facilitate detection with M instead of N samples.

The full data detection procedure visualized in Fig. 2 is now described in more detail. As mentioned earlier, the frequency error Δf can be corrected when processing the oversampled LoRa signal $r_N[n]$. While this error is unknown when first receiving the signal, estimates for Δf can be calculated when synchronization is performed (see Section IV). Such an estimate is denoted by $\Delta\tilde{f}$. The frequency-shifted samples $r_{\text{shift}}[n]$ are calculated by applying

$$r_{\text{shift}}[n] = r_N[n] \cdot \exp\left[-\jmath 2\pi\, \Delta\tilde{f}\, \frac{nT_s}{N}\right]. \tag{6}$$

If no estimate is available for Δf, e.g., when still performing signal presence detection (see Section IV-B1), its value is chosen at 0. Next, to perform decimation without losing information, an anti-aliasing filter is applied to suppress frequency components from noise or interference outside of $[0, B]$. This filter operation is performed in the frequency domain by multiplying the N-point discrete Fourier transform (DFT), represented by the operator $\mathcal{F}_N$, of $r_{\text{shift}}[n]$ with $\boldsymbol{H}_h$,

given by

$$\boldsymbol{H}_h = \begin{cases} 1, & h \in [0, M-1] \\ 0, & h \in [M, N-1] \end{cases} \tag{7}$$

which is the N-point ($h \in [0, N-1]$) discrete implementation of the filter window described by $H(f)$

$$H(f) = \begin{cases} 1, & f \in [0, B] \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The filtered signal is obtained by applying the inverse N-point DFT. Hence, the filtered signal $r_{\text{filt}}[n]$ is given by

$$r_{\text{filt}}[n] = \mathcal{F}_N^{-1}\{\mathcal{F}_N\{r_{\text{shift}}[n]\} \cdot \boldsymbol{H}_i\}. \tag{9}$$

After applying the anti-aliasing filter, decimation is performed to reduce the amount of samples used to describe the symbol from N to M. The decimated samples $r_M[m]$ are obtained by applying

$$r_M[m] = r_{\text{filt}}[mK] \tag{10}$$

where $m \in [0, M-1]$. For future reference, the frequency shift, filtering, and decimation operations are described by the single operator $\mathcal{H}$. Hence, the decimated samples can also be described as

$$r_M[m] = \mathcal{H}\{r_N[n]\}. \tag{11}$$

The new sampling rate of M/T_s yields $t = mT_s/M$. Hence, the complex LoRa envelope of the received symbol a is described by

$$r_M[m] = s[m, a] = \sqrt{\frac{E_s}{M}} \exp\left[\jmath 2\pi \left(\frac{am}{M} + \frac{m^2}{2M}\right)\right]. \tag{12}$$

Note that this description no longer requires a modulo operator or a piecewise formulation as the sampling of the $-2\pi Mt/T_s$ term from (4) gives rise to $-Mt/T_s = -m, m \in \mathbb{Z}$ and thus $e^{-\jmath 2\pi m} = 1$. Additionally, the denominator of the amplitude of the signal is changed to $\sqrt{M}$ to ensure that the energy in the symbol is still equal to E_s when calculated through $\sum_{m=1}^{M} |s[m]|^2$.

Data detection can be realized by multiplying these samples with the complex conjugate (as indicated by the asterisk) of a pure up-chirp described by $s[m, 0]$ and applying the M-point DFT, represented by $\mathcal{F}_M$, to the result. If no noise is considered, the result $\boldsymbol{X}_i$ of this operation is given by

$$\boldsymbol{X}_i = \mathcal{F}_M\{r_M[m] \cdot s^*[m, 0]\} = \begin{cases} E_s, & i = a \\ 0, & i \neq a \end{cases} \tag{13}$$

where $i \in \{0, 1, \ldots, M-1\}$ denotes the output bin number of the M-point DFT.

The detected symbol $\tilde{a}$ is given by

$$\tilde{a} = \arg_i \max(|\boldsymbol{X}_i|). \tag{14}$$

The signals employed in this detection procedure are visualized in Fig. 3, where $r_M[m]$ is described by an arbitrary symbol $s[m, a]$ and, because of the sampling frequency $M/Ts = B$, baseband frequencies are represented in the interval $[0, B]$.

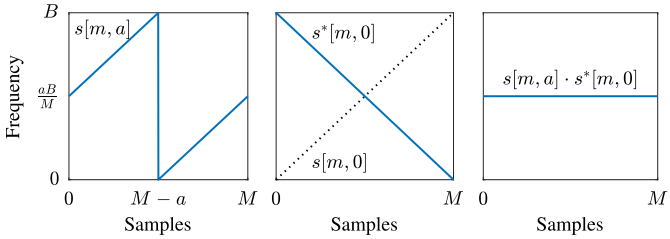


Fig. 3. Instantaneous frequency of the received signal $r_M[m] = s[m, a]$, the complex conjugate up-chirp $s^*[m, 0]$, and the product of both in the absence of noise, with no oversampling applied.

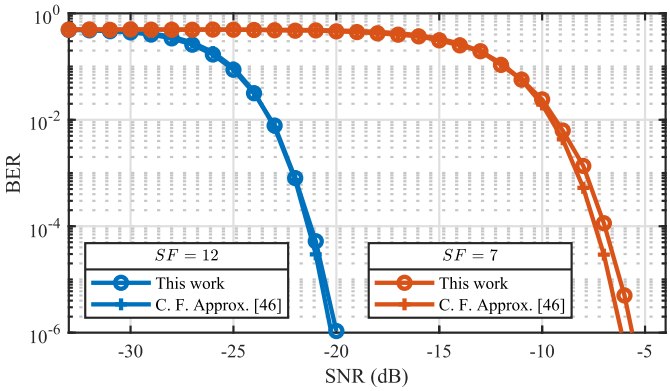


Fig. 4. BERs for SF = 12 and SF = 7 compared to closed-form approximations for the BER presented in [46].

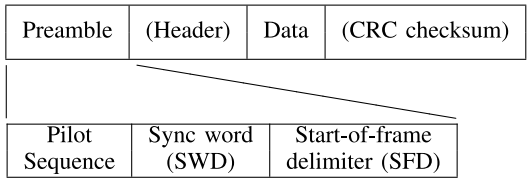


Fig. 5. LoRa packet structure.

C. Bit Error Rate

As detecting a LoRa signal with oversampling required modifications to the data detection procedure, it is interesting to examine the performance of the resulting method. Hence, BER simulations were performed for different SNR values for an AWGN channel. The SNR is defined as $E_s/(N_0 M)$, where N_0 is the power spectral density of the noise. In Fig. 4, the results of these simulations are compared to the approximate analytical results for LoRa BER performance presented in [46]. The calculated BERs and those predicted through the closed-form expressions in [46] agree very well. For SF = 7, there is a small performance penalty for lower BER values. However, in general, it can be concluded that the modifications made to the detection procedure to enable oversampling do not have a strong influence on the performance of the system.

III. LoRa Packet Structure

As the structure of LoRa packets will be referenced on multiple occasions in Section IV, it is briefly summarized here. Based on the definition of LoRa symbols presented in Section II, LoRa packets can be constructed. In general, these take the form shown in Fig. 5. The preamble starts with a

fixed amount of up-chirps ($s[m, 0]$), which are used to detect the presence of LoRa packets. This pilot sequence is followed by a short sync word and a so-called start-of-frame delimiter (SFD). The sync word consists of two regular LoRa symbols with known, nonzero values and aims to separate different LoRa networks using the same frequency band in the same geographical area. The SFD consists of 2.25 down-chirps and is used by LoRa receivers to achieve proper synchronization when capturing a packet.

As indicated above, the preamble is followed by a number of symbols which may consist of a header, the actual data symbols, and a cyclic redundancy check (CRC), the former and latter of which are optional. Since certain LoRa settings also feature forward error correction (FEC), data are encoded according to a predefined code rate CR $\in \{4/5, 4/6, 4/7, 4/8\}$ [4]. As our paper focuses on the uncoded physical layer of LoRa modulation, the remainder of this work assumes the optional packet header and CRC checksum to be part of the data payload. A pilot sequence of 8 up-chirps is considered, which yields a preamble length of 12.25 symbols. The interleaving and decoding steps that need to be performed to access the data encapsulated in the payload have been documented in [31] and are considered to be part of the data link layer. Consequently, these are not discussed in this article.

IV. LoRa Receiving Algorithm

When receiving LoRa packets, signal presence detection and synchronization need to be performed before the raw data can be detected. In the following sections, the implementation and optimization of these processes are discussed in more detail. Subsequently, a more sophisticated strategy for detecting oversampled LoRa packets is presented.

A. Signal Presence Detection

Assume that the received signal contains a packet that starts at the time instant $t = t_1 > 0$. This signal is represented by a sequence of samples $r_N[n]$, taken at the rate N/T_s, with $N = KM$, for which the sample with index $n = 0$ corresponds to the time instant $t = 0$. The first sample of the packet is described by $n_1 = \lceil t_1 \cdot N/T_s \rceil$. The actual start of the LoRa packet is denoted by $\eta_1 = t_1 \cdot N/T_s$. Note that η_1 is a fractional number, with its fractional part describing the amount of subsample synchronization (see Section IV-B3) that would need to be applied to achieve this point of perfect time synchronization. Additionally, a possible CFO must be taken into account. This offset is described by Δf, such that the received LoRa signal has its instantaneous frequencies in the range $[\Delta f, \Delta f + B]$. The detection of oversampled LoRa packets is realized using a stepped delay algorithm that isolates N samples from the sequence of received samples for each delay step. The index of the first sample of this interval is given by qN, such that the N received samples for a given delay step q are represented by

$$r_{N,q}[n] = r_N[n + qN] \tag{15}$$

where $n \in \{0, 1, \ldots, N-1\}$ and $q \in \mathbb{Z}$. For each delay step, the sequence of corresponding samples $r_{N,q}[n]$ is processed using the symbol detection procedure presented in Section II-B. As

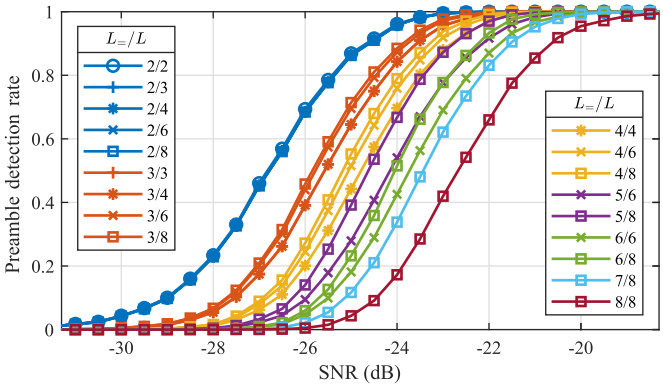


Fig. 6. Preamble detection rates for different detection settings ($L_=/L$) at $SF = 12$ and $K = 4$.

no estimate is available for the frequency offset, $\Delta\tilde{f}$ is still zero. To avoid that the digital anti-aliasing filter causes significant signal distortion during the synchronization process when no frequency correction is yet applied, we limit our attention to the case where $-\Delta f_{\max} < \Delta f < \Delta f_{\max}$, with $\Delta f_{\max}$ limited to a few percent of B. Such a limited CFO is not expected to have a significant impact on the preamble detection performance, as preamble detection only relies on detecting the presence of chirps instead of detecting the symbol value that is encoded in those chirps. When the latter is performed, the impact of the CFO is much larger.

By default, a LoRa packet is considered successfully received when four equal LoRa symbols are detected in a row [47]. These symbols are assumed to be part of the pilot sequence of up-chirps that form the start of the preamble. When a signal is detected, a very rough estimate for the first sample of the packet is given by $\tilde{q}N$, where $\tilde{q}$ indicates the delay step that provoked detection. In [35], it is shown that LoRa receivers can be made more sensitive by lowering the amount of equal symbols required for packet detection and relaxing the need for them to be consecutive. Yet, Edward *et al.* [35] only presented LoRa packet detection performance for limited variations on these constraints. In Fig. 6, the preamble detection rate is considered for a wider range of detection settings. By doing so, the algorithm can be optimized to achieve the highest detection rate. The presented variations are labeled as the amount of equal symbols required for detection, indicated as $L_=$, over the total amount of symbols considered in each step of the delay algorithm, indicated as L. Hence, when $L_=$ and L are equal, L consecutive equal symbols are required for packet detection. When $L_= < L$, only $L_=$ symbols need to be the same in L consecutive symbol intervals to trigger the detector. For each data point in Fig. 6, 10^4 attempts were made to detect a LoRa packet embedded in AWGN noise and preceded by a random number ($\in \mathbb{Z} \cap [15N, 25N]$) of noise samples to simulate a random reception delay.

Fig. 6 clearly illustrates how better packet detection is facilitated by relaxing the requirements for packet detection through decreasing $L_=$ or increasing L. For example, the 90 % packet detection threshold is lowered from -23 to -25 dB

TABLE I
FPDR AT DIFFERENT DETECTION SETTINGS ($L_=/L$) AT SF= 12 AND $K = 4$

$L_=/L$	FPDR (%)
2/2	2.38
2/3	10.96
2/4	14.48
2/6	17.88
2/8	19.55

$L_=/L$	FPDR (%)
3/3	0.13
3/4	0.19
3/8	0.37
4/4	< 0.01
others	< 0.01

when using a 2/2 setting instead of the default 4/4. Fig. 6 also shows that there is no distinguishable impact on packet detection performance when increasing L for $L_= = 2$. Note that due to LoRa's excellent performance for very low SNR values, no signal threshold should be met when detecting packets. This does increase the odds of misinterpreting a random noise sequence as a LoRa packet. Such a false positive signal presence detection may cause a delay in the overall packet detection process. Yet, despite the impact on the computational cost of the algorithm, effective packet reception is not expected to be jeopardized as these false positives will be identified when assessing the value of the sync word described in Section III, and when applying the (optional) CRC. To quantify the impact of false positives on the receiver, the detection rate of false positives, as measured in the experiment that yielded Fig. 6, is given in Table I. This shows that the false positive detection rate (FPDR) is only significant for $L_= = 2$ and $L \geq 3$. Based on these considerations, the detection setting of 2/2 is used from now on.

B. Synchronization

The synchronization procedure applied in this work utilizes both the up-chirps in the pilot sequence of the LoRa packet and the down-chirps in the SFD. When detecting the symbol values for these chirps, timing and frequency errors will result in nonzero symbol values for both. The current estimation for the first sample of the packet is denoted by $\tilde{n}_1$ and the estimation for the frequency offset is described by $\Delta\tilde{f}$. During the synchronization procedure of the algorithm, both estimations will be made and improved in each subsequent synchronization step. In general, we define the relative timing and frequency errors as

$$\epsilon_n = \frac{\tilde{n}_1 - \eta_1}{N} \quad \text{and} \quad \epsilon_f = \frac{\Delta\tilde{f} - \Delta f}{B}. \tag{16}$$

An illustration of the up-chirps in the pilot sequence and the down-chirps in the SFD, with indication of these errors, is shown in Fig. 7.

A timing error shifts the signal in such a way that the frequency reset (at B) occurs either earlier or later. Hence, such a timing error causes a symbol error which is approximated by $\epsilon_n M (\text{mod } M)$. Similarly, a frequency error also causes a symbol error which is approximated by $-\epsilon_f M (\text{mod } M)$. Note that this error has the opposite sign of the frequency error. Hence, a timing error and a frequency error may cancel out to a certain degree. The symbol level found when processing an up-chirp from the pilot sequence in the absence of noise is

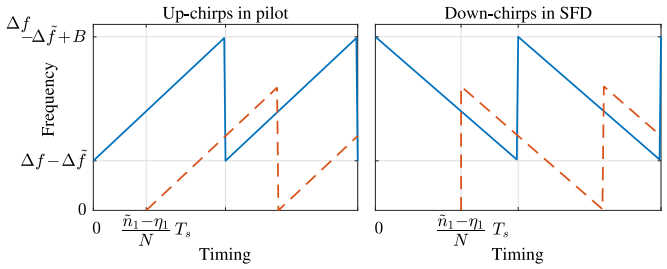


Fig. 7. Instantaneous frequencies of the actual preamble chirps (solid line) and the estimated preamble chirps (dashed line), relative to the operating frequency of the receiver. Hence, for a negative frequency estimation error ($\Delta\tilde{f} - \Delta f < 0$), the lowest frequency of the actual preamble chirps is $\Delta f - \Delta\tilde{f}$ Hz higher than what the receiver expects.

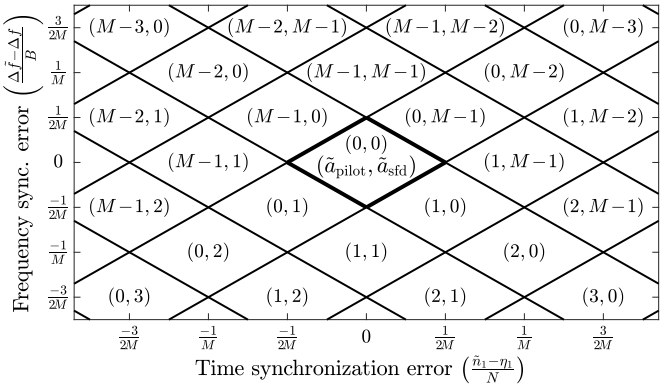


Fig. 8. $\tilde{a}_{\text{pilot}}$ and $\tilde{a}_{\text{sfd}}$ for different time and frequency synchronization errors.

given by

$$\tilde{a}_{\text{pilot}} = \left\lfloor \left(\epsilon_n - \epsilon_f\right) M \right\rceil (\text{mod } M) \tag{17}$$

where $\lfloor \cdot \rceil$ indicates rounding to the nearest integer. The down-chirps in the SFD can be processed in a similar way. In theory, their symbol values should be detected by using complex conjugate down-chirps. However, when a sample rate of M/T_s is adopted, a regular down-chirp $\bar{s}[m, 0]$ can be constructed relatively easily by applying $\bar{s}[m, 0] = s^*[m, 0]$. Hence, a complex conjugate down-chirp is actually equivalent to a normal up-chirp $s[m, 0]$. As a result, the symbol values for the down-chirps in the SFD can be detected by substituting the complex conjugate up-chirp in (13) by a regular up-chirp. In the absence of noise, this yields

$$\tilde{a}_{\text{sfd}} = \left\lfloor \left(-\epsilon_n - \epsilon_f\right) M \right\rceil (\text{mod } M). \tag{18}$$

When plotting the detected symbol values for $\tilde{a}_{\text{pilot}}$ and $\tilde{a}_{\text{sfd}}$ as a function of different time and frequency synchronization errors, (17) and (18) yield a diamond pattern of rhombuses, as shown in Fig. 8.

In this implementation of LoRa, packet synchronization is achieved in four steps.

1) First, a coarse symbol synchronization step is performed to align $\tilde{n}_1$ with the start of the first of the $L_=$ symbols that triggered the packet detector.
2) Next, the first symbol of the preamble is identified by analyzing the amount of up- and down-chirps that are present before and after the detected symbol.

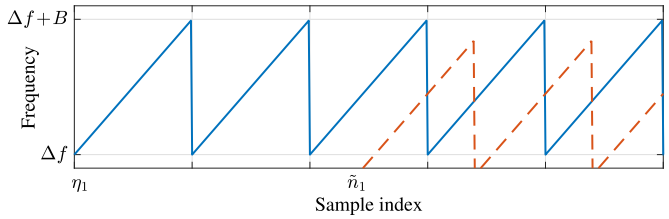


Fig. 9. Instantaneous frequencies of the actual preamble chirps (solid line) and the estimated preamble chirps (dashed line), prior to synchronization.

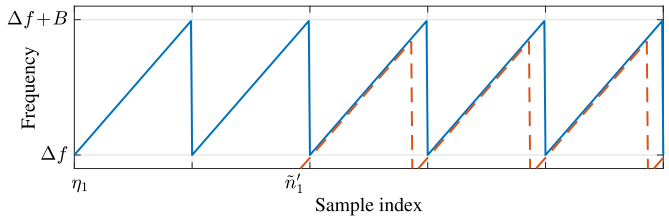


Fig. 10. Instantaneous frequencies of the actual preamble chirps (solid line) and the estimated preamble chirps (dashed line), after coarse symbol synchronization.

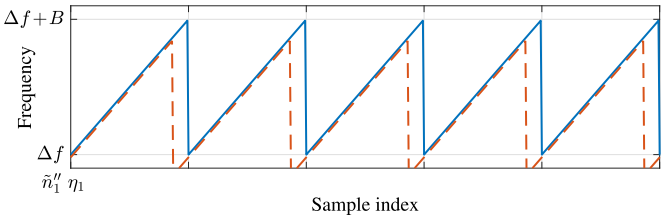


Fig. 11. Instantaneous frequencies of the actual preamble chirps (solid line) and the estimated preamble chirps (dashed line), after frame synchronization.

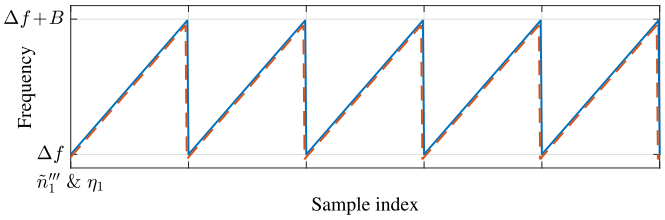


Fig. 12. Instantaneous frequencies of the actual preamble chirps (solid line) and the estimated preamble chirps (dashed line), after joint symbol and carrier frequency synchronization. Small timing and frequency errors may still be present.

3) Subsequently, an additional synchronization step is performed to correct a potential constant frequency mismatch between the detector and the received packet based on the symbol levels of the up-chirps in the pilot sequence and the down-chirps that form the SFD.
4) Finally, by employing subsample synchronization methods (see Section IV-B3), synchronization can be improved further, which involves timing corrections smaller than $T_s/(KM)$. These corrections ensure that energy is not divided between two adjacent frequency bins when performing data detection.

Below, all synchronization steps are described in detail based on the stepwise refinement of the first preamble sample estimation $\tilde{n}_1$ and the estimation for the frequency offset $\Delta\tilde{f}$. This means that the value for $\tilde{n}_1$ is updated after each subsequent synchronization step, each time reducing the distance between $\tilde{n}_1$ and η_1. At the end of each section, an update for $\tilde{n}_1$ and $\Delta\tilde{f}$ is indicated by one or more added primes. The instantaneous frequencies of the actual preamble and the estimation for the preamble are also visualized in Figs. 9–12. These illustrate how each step contributes to achieving full synchronization.

1) Coarse Symbol Synchronization: In this first synchronization step, the index of the first sample of the up-chirp that triggered the detector is estimated. No frequency correction is applied yet, i.e., $\Delta f' = 0$. This operation is fairly straightforward as this update for $\tilde{n}_1$, denoted as $\tilde{n}_1'$, can be calculated based on the symbol level $\tilde{a}_{\text{pilot}}$ found when processing this pilot chirp. Only this pilot chirp can be considered in this assessment as we do not yet know exactly how many pilot symbols precede or follow this chirp. When $\tilde{a}_{\text{pilot}} = i$, we set

$$\tilde{n}_1' = \tilde{n}_1 - i\frac{N}{M}. \tag{19}$$

This operation corresponds to a horizontal shift in Fig. 8, from a rhombus with $\tilde{a}_{\text{pilot}} = i$ to a rhombus with $\tilde{a}_{\text{pilot}} = 0$.

2) Frame Synchronization: As the pilot sequence at the start of the preamble is eight up-chirps long (see Section III), there could be up to seven symbols preceding or up to seven symbols trailing the symbol that triggered the detector. In order to find the first symbol of the pilot sequence, trial detections are performed on the full preamble for a range of delay steps (k), selecting the delay which results in the best detection. Hence, for each of the possible cases, which correspond to eight delay steps ($k \in \{-7, -6, \ldots, 0\}$), eight trial detections of possible pilot symbols are performed. Each trial detection involves selecting M samples (through changing the modifier q) and assessing the symbol strength for this q value. Still no frequency correction is applied, i.e., $\Delta f'' = 0$. For each trial detection, the samples are selected according to

$$r_{M,q}[m] = \mathcal{H}\{r_N[\tilde{n}_1' + qN + n]\}. \tag{20}$$

By iterating over $k \in \{-7, -6, \ldots, 0\}$, the aforementioned symbol strengths are calculated for the pilot sequence (pilot) as

$$\boldsymbol{F}_{k,\text{pilot}} = \sum_{q=k}^{k+7} \max\left(\left|\mathcal{F}_M\{r_{M,q}[m] \cdot s^*[m,0]\}\right|\right). \tag{21}$$

In a similar manner, the strength of the sync word (swd) candidates can be calculated as

$$\boldsymbol{F}_{k,\text{swd}} = \sum_{q=k+8}^{k+9} \max\left(\left|\mathcal{F}_M\{r_{M,q}[m] \cdot s^*[m,0]\}\right|\right). \tag{22}$$

Finally, the two full down-chirps in the SFD (sfd) can be detected, yielding the symbol strength for all SFD candidates

$$\boldsymbol{F}_{k,\text{sfd}} = \sum_{q=k+10}^{k+11} \max\left(\left|\mathcal{F}_M\{r_{M,q}[m] \cdot s[m,0]\}\right|\right). \tag{23}$$

Now, the combined strength of the pilot sequence, the sync word and the SFD can be evaluated for each delay step k.

This measure will reach its maximum value when the complex conjugate chirps in (21) and (22) and the regular up-chirp in (23) are correctly aligned with the up- and down-chirps in the preamble.

This leads to the corresponding value for k

$$k_{\max} = \arg_k \max\left(\boldsymbol{F}_{k,\text{pilot}} + \boldsymbol{F}_{k,\text{swd}} + \boldsymbol{F}_{k,\text{sfd}}\right). \tag{24}$$

Based on the result of the above equation, the new value for $\tilde{n}_1$ is found as

$$\tilde{n}_1'' = \tilde{n}_1' + k_{\max} \cdot N. \tag{25}$$

In Fig. 8, this synchronization step corresponds to a horizontal shift to the left, from a rhombus with $\tilde{a}_{\text{pilot}} = 0$ to another rhombus with $\tilde{a}_{\text{pilot}} = 0$, the latter being located on the diagonal characterized by

$$\left|\epsilon_n - \epsilon_f\right| < \frac{1}{2M}. \tag{26}$$

3) Joint Symbol and Carrier Frequency Synchronization: To avoid signal distortion by the digital anti-aliasing filter during data detection, the frequency mismatch Δf needs to be compensated. An estimate of the frequency mismatch can be obtained by performing detection on the down-chirps of the SFD. Performing this detection with $\Delta\tilde{f}'' = 0$ and the timing estimate $\tilde{n}_1''$ resulting from the frame synchronization, a symbol value $\tilde{a}_{\text{sfd}} = i$ is obtained, while the detection of the pilot symbol yields $\tilde{a}_{\text{pilot}} = 0$. This corresponds to the rhombus in Fig. 8, with $(\tilde{a}_{\text{pilot}}, \tilde{a}_{\text{sfd}}) = (0, i)$ characterized by (26). Joint symbol and carrier frequency synchronization involves the computation of the timing correction term $\Delta\tilde{n}$ and the frequency estimate $\Delta\tilde{f}$, according to

$$\begin{cases} \Delta\tilde{n} = \dfrac{1}{2}\, i\, \dfrac{N}{M} \\ \Delta\tilde{f} = \dfrac{1}{2}\, i\, \dfrac{B}{M} \end{cases} \quad \text{for} \quad i \leq \frac{M}{2} \tag{27}$$

or

$$\begin{cases} \Delta\tilde{n} = -\dfrac{1}{2}\,(M-i)\, \dfrac{N}{M} \\ \Delta\tilde{f} = -\dfrac{1}{2}\,(M-i)\, \dfrac{B}{M} \end{cases} \quad \text{for} \quad i > \frac{M}{2}. \tag{28}$$

In Fig. 8, an application of this timing and frequency correction corresponds to a shift from the rhombus with $(\tilde{a}_{\text{pilot}}, \tilde{a}_{\text{sfd}}) = (0, i)$ to the rhombus with $(\tilde{a}_{\text{pilot}}, \tilde{a}_{\text{sfd}}) = (0, 0)$. The latter rhombus is characterized by (26) and

$$\left|\epsilon_n + \epsilon_f\right| < \frac{1}{2M}. \tag{29}$$

Note that (27) and (28) yield an integer multiple of $K/2$ for $\Delta\tilde{n}$. Hence, when K is odd, a fractional number can be obtained for $\Delta\tilde{n}$. This must be taken into consideration when calculating the new approximation for the start of the packet

$$\tilde{n}_1''' = \tilde{n}_1'' + \lfloor \Delta\tilde{n} \rfloor + \delta\tilde{n}. \tag{30}$$

The remainder from $\lfloor \Delta\tilde{n} \rfloor$ can be taken into account by applying subsample synchronization. Subsample synchronization is realized by subtracting a small constant time shift $\delta\tilde{n}/K \in \mathbb{R}$, smaller than T_s/N, from m in the up-chirp $s[m, 0]$ used in the detection process. Hence, (13) is replaced by

$$\boldsymbol{X}_i' = \mathcal{F}\left\{r_M[m] \cdot s^*\left[m - \delta\tilde{n}/K, 0\right]\right\}. \tag{31}$$

This effectively adds a small shift to the timing of the detector. In this synchronization step, the size of this time shift (expressed in sampling intervals) is given by

$$\delta\tilde{n} = \Delta\tilde{n} - \lfloor \Delta\tilde{n} \rfloor. \tag{32}$$

Here, $\delta\tilde{n}$ will either be zero or half a sample interval, referred to the rate N/T_s. However, in the next section, this value may be replaced by any fractional number. The frequency offset is corrected in the detection procedure itself, as described in Section II-B. To conform to the previously introduced notation, three primes are added to $\Delta\tilde{f}$, yielding $\Delta\tilde{f}''' = \Delta\tilde{f}$. When now considering the accuracy of $\tilde{n}_1'''$ and $\Delta\tilde{f}'''$, we note that the time and frequency errors have been reduced to a random point inside the center rhombus.

4) Advanced Synchronization: Despite the synchronization steps taken in the previous sections, there is still one mechanism that can noticeably degrade bit error performance, given the current accuracy of $\tilde{n}_1'''$ and $\Delta\tilde{f}$. When the actual synchronization errors lead to a point that is close to the edges of the center rhombus in Fig. 8, a significant portion of the symbol energy will appear in the wrong DFT frequency bin after detection. If the received packet has a low SNR, this leads to a significant number of symbol errors as random noise often causes the detector to return $a - 1$ or $a + 1$ instead of the original symbol a, as illustrated in Fig. 13.

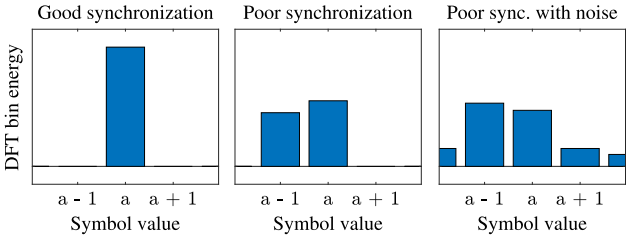


Fig. 13. DFT bin energies for an arbitrary symbol a with good synchronization, poor synchronization, and poor synchronization in combination with a low SNR. In the latter, a symbol error is observed.

To overcome these issues, more advanced synchronization strategies can be employed to better approximate η_1 and Δf. These strategies can expand on subsample time and frequency synchronization by adding an additional time offset to $s[m, 0]$, as introduced in Section IV-B3, and by further modifying $\Delta\tilde{f}$. This means that the symbol levels that would be detected when the signal is processed would not change when locally shifting the timing of the detector and the frequency correction applied to $r_N[n]$ if no noise were present. Yet, in case of a low SNR, this final improvement in synchronization accuracy is in fact found to positively impact the BER, as demonstrated in Section V-B.

To realize this more advanced time and frequency synchronization step, a well-chosen performance indicator should be optimized as a function of the very small time ($\delta\tilde{n}/K$) and frequency ($\delta\tilde{f}$) offsets subtracted from m in $s[m, 0]$ and added

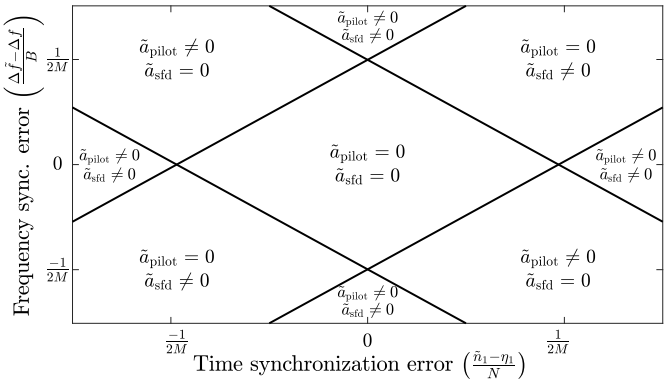


Fig. 14. Illustration of the optimization space boundaries for $\delta\tilde{n}$ and $\delta\tilde{f}$, governed by $(\tilde{a}_{\text{pilot}}, \tilde{a}_{\text{sfd}}) = (0, 0)$.

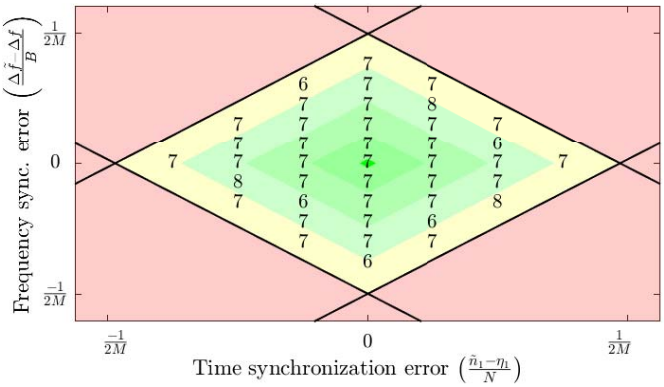


Fig. 15. Example of detected symbol values in $\boldsymbol{A}$ for an arbitrary symbol value of 7 in low SNR conditions. Red areas indicate incorrect synchronization. In the yellow area, synchronization is poor and leads to symbol errors through the mechanism demonstrated in Fig. 13. In greener areas, less symbol errors occur at low SNRs. The advanced detection procedure presented in this chapter selects the modal value of all candidate values in this space as the detected symbol.

to $\Delta\tilde{f}$, respectively. The corresponding time and frequency synchronization estimates are described by

$$\tilde{n}_1 = \tilde{n}_1''' + \delta\tilde{n} \qquad \text{and} \qquad \Delta\tilde{f} = \Delta\tilde{f}''' + \delta\tilde{f}. \tag{33}$$

The boundaries of the optimization space for $\delta\tilde{n}$ and $\delta\tilde{f}$ are governed by the values of $\tilde{a}_{\text{pilot}}$ and $\tilde{a}_{\text{sfd}}$, which have to be zero for all test points. As illustrated in Fig. 14, this optimization space corresponds to the center rhombus in Fig. 8. A good choice for the performance indicator is the energy in the strongest frequency bin when detecting the symbol values in the preamble, as this measure will reach a maximum for $\tilde{n}_1''' + \delta\tilde{n} = \eta_1$ and $\Delta\tilde{f} + \delta\tilde{f} = \Delta f$, which corresponds to the center of the optimization space shown in Fig. 14. This optimization space is mapped out by detecting the symbol values for $\tilde{a}_{\text{pilot}}$ and $\tilde{a}_{\text{sfd}}$ as a function of different $\delta\tilde{n}$ and $\delta\tilde{f}$. As introduced in Section IV-B3, the errors associated with the estimates $\Delta\tilde{n}_1'''$ and $\Delta\tilde{f}'''$ correspond to a random point inside this rhombus. Hence, the range for $\delta\tilde{n}$ and $\delta\tilde{f}$ should cover an area around this point four times the area of this rhombus, as the worst case scenarios for $\Delta\tilde{n}_1'''$ and $\Delta\tilde{f}'''$ correspond to the vertices of the rhombus. The resolution of $\delta\tilde{n}$ and $\delta\tilde{f}$ can be chosen freely.

As introduced for timing corrections of $1/2$ times the sample interval in Section IV-B3, subsample synchronization is again achieved by altering the timing of the detector through subtracting $\delta\tilde{n}/K$ from m in the up-chirp $s[m, 0]$ used in the detection process, as demonstrated in (31). Optimization algorithms such as gradient descent [48] can be applied to reduce the (possibly significant) computational cost of this synchronization step if there are any relevant timing or power consumption restrictions. When the optimal values for $\delta\tilde{n}$ and $\delta\tilde{f}$ are determined (based on the chosen performance indicator), the symbols in the payload can be detected using these optimized synchronization settings.

Values near the center of the optimization space shown in Fig. 14 are more likely to yield a correct symbol decision than values near the edge, as these values are closer to the perfect synchronization point $(\epsilon_n, \epsilon_f) = (0, 0)$. Hence, when symbol detection is performed for a given combination of $\delta\tilde{n}$ and $\delta\tilde{f}$, symbol energy will be concentrated more in a single DFT bin. As an alternative to maximizing a performance indicator, the optimization space of $\delta\tilde{n}$ and $\delta\tilde{f}$ can also be explored without performing this calculation. In this case, the average values for all $\delta\tilde{n}$ and $\delta\tilde{f}$ that lead to $(\tilde{a}_{\text{pilot}}, \tilde{a}_{\text{sfd}}) = (0, 0)$ in this space can be used to get close to $(\epsilon_n, \epsilon_f) = (0, 0)$, which again leads to an optimized configuration of $s[m - \delta\tilde{n}/K, 0]$ and $\Delta\tilde{f} + \delta\tilde{f}$ for detecting the actual data symbols.

C. Advanced Detection

The advanced synchronization step presented above can also be extended to the detection procedure. Instead of processing the data in the packet once, with fixed values for $\delta\tilde{n}$ and $\delta\tilde{f}$, one can also perform symbol detection on each symbol in the packet for every combination of $\delta\tilde{n}$ and $\delta\tilde{f}$ that leads to synchronization points in the optimization space governed by $\tilde{a}_{\text{pilot}} = \tilde{a}_{\text{sfd}} = 0$ and shown in Fig. 15. For packets with a very low SNR, random symbol errors that occur for very specific combinations of $\delta\tilde{n}$ and $\delta\tilde{f}$ could then be mitigated by selecting the candidate value that occurs most frequently for a given symbol. Suppose we construct a matrix $\boldsymbol{A}$, which contains the detected symbol value when applying a certain combination of $\delta\tilde{n}$ (row element) and $\delta\tilde{f}$ (column element): $\boldsymbol{A}_{\delta\tilde{n},\delta\tilde{f}} = \tilde{a}$. Elements are only added to this matrix if the combination of $\delta\tilde{n}$ and $\delta\tilde{f}$ satisfies $\tilde{a}_{\text{pilot}} = \tilde{a}_{\text{sfd}} = 0$. The size of the matrix depends on the resolution of $\delta\tilde{n}$ and $\delta\tilde{f}$, which, as mentioned earlier can be chosen freely. The actual symbol value can be selected as the modal value of $\boldsymbol{A}$. The contents of this matrix are illustrated for an arbitrary symbol level of 7 in Fig. 14. For every LoRa symbol in the payload, this procedure can be repeated. Note that selecting the modal value of $\boldsymbol{A}$ as the detected symbol actually allows for the omission of the performance indicator optimization part of the synchronization step presented in Section IV-B4, as eventually no decision is taken for $\delta\tilde{n}$ and $\delta\tilde{f}$.

V. Simulated LoRa SDR Performance

When evaluating the performance of the LoRa receiving algorithm introduced in Section IV, it is important to keep in

mind the computational cost (and hence, power consumption) of the algorithm. To this end, four variations, consisting of different combinations of the synchronization and detection strategies presented in Sections IV-B and IV-C, are compared to the performance of LoRa signal detection under perfect CSI conditions (perfect time and frequency synchronization point is known).

A. Algorithm Variations

For the first variation of the receiving algorithm, the advanced synchronization and detection steps introduced to counteract the DFT bin mismatch mechanism described in Section IV-B4 and illustrated in Fig. 13 are fully omitted. As a result, this variation, which is labeled as direct synchronization (DS), is expected to have a very low computational cost. The second algorithm variation uses the advanced synchronization procedure proposed in Section IV-B4, without calculating a performance indicator (hence, the optimal values for $\delta\tilde{n}$ and $\delta\tilde{f}$ are estimated as the mean values of all combinations of $\delta\tilde{n}$ and $\delta\tilde{f}$ that lead to points in the optimization space illustrated in Fig. 14) and omits the advanced detection procedure presented in Section IV-C. This variation is labeled as advanced synchronization with estimated optimum (AS-EO). The third variation is very similar, however, now the energy in the strongest frequency bin when demodulating and processing the chirps in the preamble is chosen as a performance indicator in the last synchronization step and optimal values for $\delta\tilde{n}$ and $\delta\tilde{f}$ are determined based on this metric. Consequently, this variation is labeled as advanced synchronization with computed optimum (AS-CO). Finally, the fourth variation is fully based on the procedure presented in Section IV-C. As previously mentioned, no estimates for $\delta\tilde{n}$ and $\delta\tilde{f}$ are calculated, but symbol values are determined by taking the mode of all candidate symbol values. Therefore, this variation is labeled as a mode of successive detection (MSD). While the MSD variation is expected to be the most accurate way of detecting any given packet, it is also expected to be the most computationally expensive.

B. Bit Error Rate

To analyze the bit error performance of the algorithm variations under study, a communication channel was simulated by adding white Gaussian noise to synthesized LoRa packets (payload = 100 random symbols, B = 125 kHz, and SF = 12). Each packet sequence was preceded by a random number ($\in \mathbb{Z} \cap [15N, 25N]$) of noise samples, shifted by a random frequency offset ($\in \mathbb{R} \cap [-5, 5]$ kHz) and fed into one of the receiving algorithms. Shifting the frequency of the packets simulates a realistic CFO. The BERs obtained through this procedure are shown in Fig. 16. For each data point, at least 100 packet errors were simulated. For the lowest BER values, this corresponds to simulating up to 10^8 packet receptions for each data point. The results are also compared to the BER that would be achieved by a system with perfect CSI.

The DS curve in Fig. 16 clearly illustrates how the absence of an advanced time and frequency synchronization step has a detrimental impact on the BER through the DFT bin mismatch mechanism described in Section IV-B4 and illustrated in Fig. 13. However, despite this deficiency caused by the accuracy of the synchronization procedure, the DS version of the algorithm could still be very useful when an adequate error-correcting code is applied. A significant improvement w.r.t. DS is observed for AS-EO. Yet, the best performing algorithm versions are AS-CO and MSD, which exhibit very similar performance. For AS-CO, this shows that even for the lowest SNR values, the optimization space of $\delta\tilde{n}$ and $\delta\tilde{f}$ can be sufficiently explored to determine the best time and frequency correction. Applying the MSD procedure yields further improvements, which are, however, limited. In average SNR performance, the theoretical system with perfect CSI outperforms the simulated MSD algorithm by only 1.6 dB. The BERs presented in Fig. 16 also show that the CFO applied to the packets is successfully mitigated by the synchronization efforts applied in the best performing algorithms.

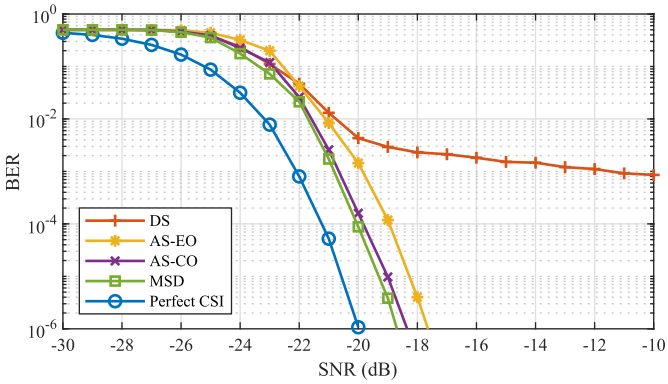


Fig. 16. BERs for all algorithm variations presented in Section V-A, compared to the BER under perfect CSI conditions, as shown in Fig. 4.

C. Computational Cost

To evaluate the computational cost of the algorithm variations under study, a LoRa link with sufficiently high SNR to allow the system to operate without significant packet loss was simulated on a 2.1 GHz Intel Core i7 8650U processor with 16 GB of memory. In practice, each version of the algorithm was presented with 100 randomized LoRa packets, consisting of N_s symbols, the packets were once again preceded by a random number ($\in \mathbb{Z} \cap [15N, 25N]$) of noise samples. AWGN noise was added to each of the packets to achieve SNR levels uniformly distributed between -20 and 0 dB. With the exception of impacting the amount of signal presence detection attempts that are made to detect the packet, the exact SNR of a received packet has no significant influence on its processing time. Fig. 17 shows the average computation times needed to find, synchronize, and process a single LoRa packet with a given packet length.

This figure clearly shows the computational impact of the synchronization steps taken in the MSD variation of the algorithm and also indicates why using DS could still be a desirable

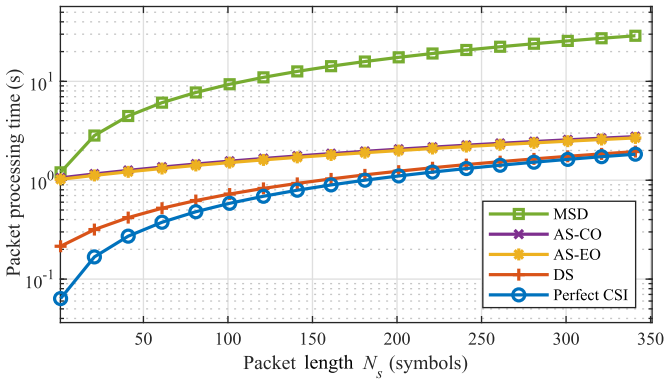


Fig. 17. Processing time needed to find, synchronize, and detect a single LoRa packet with a given length N_s for all algorithm variations presented in Section V-A.

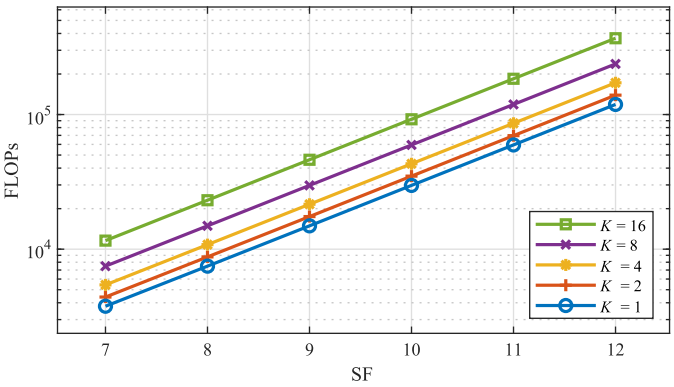


Fig. 18. Estimation of the amount of FLOPs needed to perform symbol detection at different values for the SF and oversampling factor (K).

TABLE II
PERCENTAGE OF COMPUTATIONAL RESOURCES CONSUMED BY THE MAIN OPERATIONS IN THE DETECTION PROCEDURE

Operation	Equation	Share
Decimating oversampled signal	(10)	49.6 %
Dechirping and Fourier transforming LoRa signal	(13)	24.2 %
Filtering oversampled signal	(9)	12.4 %
Applying frequency correction	(6)	8.5 %
Retrieving symbol value from DFT bins	(14)	4.6 %

option when employing the receiving algorithm in real-world LoRa applications, especially when an appropriate coding strategy would be used. AS-EO and AS-CO clearly strike a sound balance between excellent BER performance and a manageable computational cost, with AS-CO emerging as the better choice because of its superior error performance, if either the computational cost or power consumption of the receiver is an issue.

Additional diagnostics reveal that for each version of the algorithm, 98% of the computational resources are spent on detecting symbols (either when performing trial detections in the synchronization procedure or when executing the final symbol detection steps). Hence, it is interesting to zoom in on the computational cost of the detection procedure. To this end, Table II describes the percentage of computational resources consumed by each part of this procedure.

Among others, it is shown that both filtering and decimating the oversampled signal account for a large portion of the computational resources consumption. It should also be noted that the computational cost of symbol detection is heavily impacted by the values for the SF (hence, by M) and the oversampling factor (K). Based on a MATLAB implementation of the detection procedure, the amount of floating-point operations (FLOPs) needed to detect one symbol can be estimated. An overview of such estimations for different SFs and oversampling factors is provided in Fig. 18.

VI. COMPARISON TO COMMERCIAL HARDWARE

To be able to demonstrate the practical improvements achieved by the LoRa receiving algorithm variations presented in this work, these were implemented on a bladeRF 2.0 micro XA4 SDR unit [49]. In this section, the performance of these solutions is compared to that of a commercial Semtech SX1276 LoRa transceiver [50]. The SX1276 chip is integrated on a compact LoRa sensor node with extended dynamic range, which was specifically developed and calibrated for performing channel measurements [15].

A. Measurement Setup

To compare the performance of both systems, a second SX1276-based LoRa sensor node was used to generate LoRa packets (carrier frequency $f_c = 868$ MHz, payload $N_s = 13$ symbols, $B = 125$ kHz, SF $= 12$, and CR $= 4/5$), which contain a unique 16-bit packet number as payload. To obtain each data point, 100 of these packets were sent to either system through a switched attenuator, with a transmission power of 0 dBm. The size of this experiment is limited by the computational resources needed to store and process the vast amount of data samples generated in practical SDR measurements with LoRa modulation. However, as simulating 100 packet receptions results in a maximum PRR standard deviation of 1/20 (at PRR $= 50$ %), this is considered sufficient to draw meaningful conclusions. A Rohde and Schwarz FSV40 Spectrum Analyzer was used to verify the signal levels after cable loss. As the encrypting and coding scheme of the commercial transceiver is proprietary, the packets received by the SDR were error checked by remodulating these packets and presenting the result to another SX1276-receiver in a repeater configuration. Hence, for each packet received by the SDR, symbols are detected using one of the algorithm variations presented earlier. Next, based on the received symbol values, packets are reconstructed and transmitted to a second SX1276 at the SDR's maximum power level. At this power level, additional symbol errors are extremely rare. Hence, the PRR experienced by the second SX1276 is essentially the same as the PRR that would result from the SDR receiver. Identical to the calibration procedure presented in [15], the devices under test were connected using highly shielded RF cables with the transmitting LoRa sensor node being placed in an anechoic chamber and the receivers being located in a shielded control room to provide sufficient isolation between both systems. Both receivers

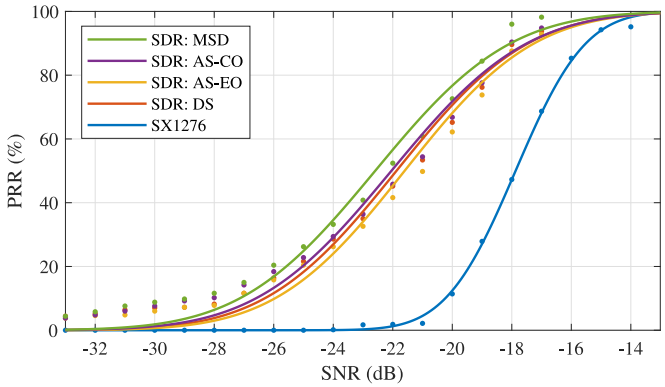


Fig. 19. PRRs measured when increasing the attenuation between a LoRa transmitter node and the two devices under test.

logged the SNR of received packets, either based on the I and Q samples in the SDR, or by extracting this information from the SX1276 transceiver in the commercial receiver. The accuracy of the SNR measurements made by the SX1276 is not publicly available. However, it is known that for fixed noise levels, 1 dB of signal strength increase does not exactly correspond to 1 dB of SNR improvement as reported by this LoRa module [15], [51]. This is corrected by applying the calibration presented in [15].

B. Results

The results of this experiment are shown in Fig. 19. This figure shows lognormal cumulative density functions fitted to the PRR data points as a function of the measured SNR. Lognormal distributions are found to agree relatively well with the data.

Fig. 19 clearly illustrates the significant performance gain achieved by the SDR system and the packet reception algorithm variations presented in this work when compared to the SX1276 transceiver. On average, an SNR performance improvement of 4.7 dB is measured for the SDR system, which means that (on average) the same PRR is observed at SNR levels 4.7 dB lower than for the commercial LoRa system. Furthermore, the 10 % outage probability occurs at an SNR level of -18.0 dB. This is a significant gain when compared to the -15.5 dB measured for the commercial system. When higher outage probabilities are considered, this difference is even larger. The average 4.7 dB performance gain corresponds to a sizable 172% range improvement in free space for receivers with the same noise floor. Less obvious performance differences are observed between the different algorithm variations running on the SDR. However, these do agree with the relative BER levels presented in Fig. 16.

VII. Conclusion

An advanced LoRa packet reception algorithm was developed and implemented on SDR. LoRa chirp and symbol formulations were presented and extended to include frequency offset and oversampling capabilities. Additionally, a symbol detection procedure for detecting oversampled LoRa symbols was constructed and tested. A unique LoRa receive algorithm was conceived, applying a state-of-the-art signal presence detection strategy, rigorous packet synchronization methods, and an advanced data detection procedure to yield the best possible link quality. Every part of the algorithm was described in detail, keeping in mind various mechanisms that could significantly degrade bit error performance. Different algorithm variations were presented and explored by examining the BERs obtained from AWGN channel simulations. Those algorithm variations that actively correct small timing and frequency synchronization errors were shown to perform particularly well, at best leaving only 1.6 dB of SNR performance to be gained when compared to BERs obtained by a theoretical system with perfect CSI. Moreover, the proposed algorithm variations have been shown to accurately correct a possible CFO. The comparison between all presented algorithm variations also clearly illustrated the pertinent trade-off between link performance and computational cost. When considering the best performing algorithm version's implementation on SDR, average SNR performance improvements up to 4.7 dB were observed when compared to industry standard LoRa hardware.

In future work, FPGA technology or advanced ARM architectures could be employed to parallelize certain parts of the algorithm presented in this work, limiting the time needed to process packets while fully taking advantage of the excellent performance demonstrated in this article. Parts of the algorithm that qualify for parallelization are signal presence detection and frame synchronization (where in both cases different delay steps could be processed in parallel), and advanced synchronization and detection (which both feature a lot of trial detections at different time and frequency offsets). The presented LoRa receiving algorithm could in theory be implemented on any SDR platform. Consequently, this implementation of LoRa enables more in depth research on the physical performance of LoRa modulation as it makes the physical layer fully available to researchers. When integrated in existing LoRa networks, the proposed algorithm could yield significant improvements in range and/or reliability for both base stations and remote nodes with sufficient computational power.

References

[1] SigFox Foundation. *SigFox*. Accessed: Apr. 20 2021. [Online]. Available: https://www.sigfox.com/en

[2] GSM Association. *NarrowBand—Internet of Things (NB-IoT)*. Accessed: Apr. 20 2021. [Online]. Available: https://www.gsma.com/iot/narrow-band-internet-of-things-nb-iot/

[3] LoRa Alliance. *LoRa*. Accessed: Apr. 20 2021. [Online]. Available: https://www.lora-alliance.org/

[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, p. 1466, 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 *Proc. 19th ACM Int. Conf. Model. Anal. Simulat. Wireless Mobile Syst.*, 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 *Proc. Int. Conf. Adv. Technol. Signal Image Process. (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 Commun.*, vol. 23, no. 5, pp. 60–67, Oct. 2016.

[8] L. Vangelista, "Frequency shift chirp modulation: The LoRa modulation," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1818–1821, Dec. 2017.

[9] M. Chiani and A. Elzanaty, "On the LoRa modulation for IoT: Waveform properties and spectral analysis," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8463–8470, Oct. 2019.

[10] L. Gregora, L. Vojtech, and M. Neruda, "Indoor signal propagation of LoRa technology," in *Proc. 17th Int. Conf. Mechatronics (Mechatronika)*, Dec. 2016, pp. 1–4.

[11] P. Neumann, J. Montavont, and T. Noël, "Indoor deployment of low-power wide area networks (LPWAN): A LoRaWAN case study," in *Proc. IEEE 12th Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, 2016, pp. 1–8.

[12] 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 *Proc. IEEE Conf. Antenna Meas. Appl. (CAMA)*, Dec. 2017, pp. 381–384.

[13] 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," in *Proc. ETFA*, 2017, pp. 1–8.

[14] J. Petäjäjärvi, K. Mikhaylov, R. Yasmin, M. Hämäläinen, and J. Iinatti, "Evaluation of LoRa LPWAN technology for indoor remote health and wellbeing monitoring," *Int. J. Wireless Inf. Netw.*, vol. 24, no. 2, pp. 153–165, 2017.

[15] 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, 2018. [Online]. Available: http://dx.doi.org/10.3390/s18072137

[16] T. Ameloot, P. Van Torre, and H. Rogier, "Indoor body-to-body LoRa link characterization," in *Proc. 9th IEEE-APS Topical Conf. Antennas Propag. Wireless Commun.*, Sep. 2019, pp. 42–47.

[17] 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 *Proc. 14th Int. Conf. ITS Telecommun. (ITST)*, Dec. 2015, pp. 55–59.

[18] 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," *Int. J. Distrib. Sens. Netw.*, vol. 13, no. 3, pp. 1–16, 2017. [Online]. Available: https://doi.org/10.1177/1550147717699412

[19] O. Iova *et al.*, "LoRa from the city to the mountains: Exploration of hardware and environmental factors," in *Proc. Int. Conf. Embedded Wireless Syst. Netw.*, 2017, pp. 317–322. [Online]. Available: http://dl.acm.org/citation.cfm?id=3108009.3108091

[20] T. Ameloot, P. Van Torre, and H. Rogier, "LoRa base-station-to-body communication with SIMO front-to-back diversity," *IEEE Trans. Antennas Propag.*, vol. 69, no. 1, pp. 397–405, Jan. 2021.

[21] T. Ameloot, P. Van Torre, and H. Rogier, "Experimental parameter optimization for adaptive LoRa modulation in body-centric applications," in *Proc. 14th Eur. Conf. Antennas Propag. (EuCAP)*, 2020, pp. 1–5.

[22] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?" *IEEE Wireless Commun. Lett.*, vol. 6, no. 2, pp. 162–165, Apr. 2017.

[23] K. Mikhaylov, J. Petäjäjärvi, and T. Haenninen, "Analysis of capacity and scalability of the LoRa low power wide area network technology," in *Proc. 22th Eur. Wireless Conf.*, May 2016, pp. 1–6.

[24] B. Reynders, W. Meert, and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, 2016, pp. 1–6.

[25] M. Cattani, C. A. Boano, and K. Römer, "An experimental evaluation of the reliability of LoRa long-range low-power wireless communication," *J. Sens. Actuator Netw.*, vol. 6, no. 2, p. 7, 2017. [Online]. Available: http://www.mdpi.com/2224-2708/6/2/7

[26] C. A. Boano, M. Cattani, and K. Römer, "Impact of temperature variations on the reliability of LoRa—An experimental evaluation," in *Proc. 7th Int. Conf. Sens. Netw.*, vol. 1, 2018, pp. 39–50.

[27] T. Ameloot, P. Van Torre, and H. Rogier, "Periodic LoRa signal fluctuations in urban and suburban environments," in *Proc. 13th Eur. Conf. Antennas Propag. (EuCAP)*, 2019, pp. 1–5.

[28] B. Sikken. *Decoding LoRa*. Accessed: Apr. 20 2021. [Online]. Available: https://revspace.nl/DecodingLora

[29] J. Blum. *LoRa SDR Project*. Accessed: Apr. 20 2021. [Online]. Available: https://github.com/myriadrf/LoRa-SDR

[30] RTL-SDRangelove. *Github Project Page*. Accessed: Apr. 20 2021. [Online]. Available: https://github.com/hexameron/rtl-sdrangelove

[31] A. Marquet, N. Montavont, and G. Z. Papadopoulos, "Towards an SDR implementation of LoRa: Reverse-engineering, demodulation strategies and assessment over Rayleigh channel," *Comput. Commun.*, vol. 153, pp. 595–605, Mar. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366419314665

[32] R. Ghanaatian, O. Afisiadis, M. Cotting, and A. Burg, "LoRa digital receiver analysis and implementation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 1498–1502.

[33] J. Tapparel, O. Afisiadis, P. Mayoraz, A. Balatsoukas-Stimming, and A. Burg, "An open-source LoRa physical layer prototype on GNU radio," in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2020, pp. 1–5.

[34] P. Robyns, P. Quax, W. Lamotte, and W. Thenaers, "A multi-channel software decoder for the LoRa modulation scheme," in *Proc. 3rd Int. Conf. Internet Things Big Data Security*, vol. 1, 2018, pp. 41–51.

[35] P. Edward, A. Muhammad, S. Elzeiny, M. Ashour, T. Elshabrawy, and J. Robert, "Enhancing the capture capabilities of LoRa receivers," in *Proc. Int. Conf. Smart Appl. Commun. Netw. (SmartNets)*, 2019, pp. 1–6.

[36] A. A. Kherani and K. M. P. Maurya, "Improved packet detection in LoRa-like chirp spread spectrum systems," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, 2019, pp. 1–4.

[37] T. Elshabrawy, P. Edward, M. Ashour, and J. Robert, "On the different mathematical realizations for the digital synthesis of LoRa-based modulation," in *Proc. 25th Eur. Wireless Conf.*, 2019, pp. 1–6.

[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 Pers. Commun.*, vol. 120, pp. 2291–2306, Feb. 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, p. 4123, 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. 75721–75731, 2019.

[41] L. Fernandez, J. A. Ruiz-De-Azua, A. Calveras, and A. Camps, "Assessing LoRa for satellite-to-earth communications considering the impact of ionospheric scintillation," *IEEE Access*, vol. 8, pp. 165570–165582, 2020.

[42] S. Hassayoun, S. Lahouar, and K. Besbes, "SDR bridge for a secure wireless sensor network (WSN)," in *Proc. IEEE Int. Conf. Design Test Integr. Micro Nano-Syst. (DTS)*, 2020, pp. 1–5.

[43] C. Gavrila, C. Kertesz, M. Alexandru, and V. Popescu, "Reconfigurable IoT gateway based on a SDR platform," in *Proc. Int. Conf. Commun. (COMM)*, 2018, pp. 345–348.

[44] D. M. Molla, H. Badis, A. A. Desta, L. George, and M. Berbineau, "SDR-based reliable and resilient wireless network for disaster rescue operations," in *Proc. Int. Conf. Inf. Commun. Technol. Disaster Manage. (ICT-DM)*, 2019, pp. 1–7.

[45] R. Narayanan and S. Kumar, "Revisiting software defined radios in the IoT era," in *Proc. 17th ACM Workshop Hot Topics Netw.*, 2018, pp. 43–49. [Online]. Available: https://doi.org/10.1145/3286062.3286069

[46] T. Elshabrawy and J. Robert, "Closed-form approximation of LoRa modulation BER performance," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1778–1781, Sep. 2018.

[47] A. Rahmadhani and F. Kuipers, "When LoRaWAN frames collide," in *Proc. 12th Int. Workshop Wireless Netw. Testbeds Exp. Eval. Characterization*, 2018, pp. 89–97. [Online]. Available: https://doi.org/10.1145/3267204.3267212

[48] S. Ruder, "An overview of gradient descent optimization algorithms," 2016. [Online]. Available: arXiv:1609.04747.

[49] Nuand. *BladeRF 2.0 Micro xA4*. Accessed: Apr. 20 2021. [Online]. Available: https://www.nuand.com/product/bladerf-xa4/

[50] "137 MHz to 1020 MHz low power long range transceiver," Data Sheet SX1276, Semtech Corp., Camarillo, CA, USA, 2016.

[51] J. Gaelens, P. Van Torre, J. Verhaevert, and H. Rogier, "LoRa mobile-to-base-station channel characterization in the antarctic," *Sensors*, vol. 17, no. 8, p. 1903, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/8/1903

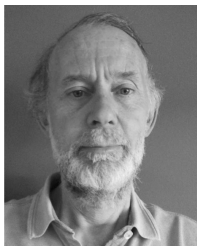
Thomas Ameloot received the M.Sc. degree in electronics and ICT engineering technology and the Ph.D. degree in electrical engineering from Ghent University, Ghent, Belgium, in 2017 and 2021, respectively.

Through his research on the physical performance of long-range, low-power wireless communication technologies, he contributed five journal and seven conference proceedings papers. He is currently active as a Postdoctoral Researcher with the Electromagnetics Group, Ghent University, continuing his research into sub-GHz long-range wireless sensor network technologies and software-defined radio applications.

Hendrik Rogier (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from Ghent University, Ghent, Belgium, in 1994 and 1999, respectively.

From 2003 to 2004, he was a Visiting Scientist with the Mobile Communications Group, Vienna University of Technology, Vienna, Austria. He is a currently a Senior Full Professor with the Department of Information Technology, Ghent University, a Guest Professor with the Interuniversity Microelectronics Centre, Ghent, and a Visiting Professor with the University of Buckingham, Buckingham, U.K. He has authored or coauthored over 185 papers in international journals and over 200 contributions in conference proceedings. His current research interests include antenna systems, radio wave propagation, body-centric communication, numerical electromagnetics, electromagnetic compatibility, and power/signal integrity.

Prof. Rogier was a recipient of the URSI Young Scientist Award (twice) at the 2001 URSI Symposium on Electromagnetic Theory and at the 2002 URSI General Assembly, the 2014 Premium Award for Best Paper in the *IET Electronics Letters*, the Best Paper Award First Place in the 2016 IEEE MTT-S Topical Conference on Wireless Sensors and Sensor Networks, the Best Poster Paper Award at the 2012 IEEE Electrical Design of Advanced Packaging and Systems Symposium, the Best Paper Award at the 2013 IEEE Workshop on Signal and Power Integrity, and the Joseph Morrissey Memorial Award for the First Best Scientific Paper at BioEM 2013. From 2017 to 2019, he was an Associate Editor of IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES. He is currently an Associate Editor of *IET Electronics Letters* and *IET Microwaves, Antennas and Propagation*. He acts as the URSI Commission B Representative for Belgium. He is a member of Technical Committee 24 on RFID Technology with the IEEE Microwave Theory and Techniques Society (MTT-S) and a member of the Governing Board of Topical Group MAGEO on Microwaves in Agriculture, Environment, and Earth Observation with the European Microwave Association, Leuven, Belgium.



Marc Moeneclaey (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from Ghent University, Ghent, Belgium, in 1978 and 1983, respectively.

He is a Professor Emeritus with the Department of Telecommunications and Information Processing, Ghent University. He has authored more than 500 scientific articles in international journals and conference proceedings. Together with Prof. H. Meyr (RWTH Aachen) and Dr. S. Fechtel (Siemens AG), he has coauthored the book *Digital Communication Receivers*, *Synchronization*, *Channel Estimation*, and *Signal Processing* (Wiley, 1998). His main research interests include statistical communication theory, carrier and symbol synchronization, bandwidth-efficient modulation and coding, spread spectrum, and satellite and mobile communication.

Prof. Moeneclaey was a co-recipient of the Mannesmann Innovations Prize 2000. From 1992 to 1994, he was an Editor of *Synchronization* and the IEEE TRANSACTIONS ON COMMUNICATIONS. He has served as a Co-Guest Editor for special issues of the *Wireless Personal Communications* (on Equalization and Synchronization in Wireless Communications) in 1998 and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (on Signal Synchronization in Digital Transmission Systems) in 2001.

Patrick Van Torre received the M.Sc. degree in electrical engineering from University College Ghent, Ghent, Belgium, in 1995, and the Ph.D. degree in electrical engineering from Ghent University, Ghent, in 2012.

After working for a number of years as a hardware and software designer in industry, he started as a Lecturer with University College Gent in 1999, where he was a pioneer in setting up project-oriented lab courses in a variety of different topics, ranging from analog electronics to digital signal processing and embedded systems. In 2011, his project concerning the development of didactical equipment was awarded the Queen Paola Award in Education. In combination with his ample educational responsibilities, he started his first research activities in 2008, including six journal and seven conference proceedings papers as well as a scientific award. He is currently active as a Professor with the Department of Information Technology, Ghent University and a Postdoctoral Researcher with imec, Leuven, Belgium. He is responsible for multiple courses in Electronics and ICT and is active as a Researcher in the field of analog electronics and wireless communications. He has authored or coauthored 113 publications in international journals and in conference proceedings. His current research interests include body-centric multiple-input–multiple-output and beam-forming systems as well as on software-defined radio and sub-GHz long-range wireless sensor networks.