Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

ELSEVIER

Towards an SDR implementation of LoRa: Reverse-engineering, demodulation strategies and assessment over Rayleigh channel

Alexandre Marquet [*], Nicolas Montavont, Georgios Z. Papadopoulos

IMT Atlantique, IRISA, F-35576 Cesson Sévigné, France

ARTICLE INFO

Keywords:
LoRa
Internet of Things (IoT)
Chirp Spread Spectrum (CSS)
Reverse engineering
Low-Power Wide Area Networks (LPWAN)
Soft-value decoding
Rayleigh channel
Software Defined Radio (SDR)
Cognitive radio

ABSTRACT

LoRa is a popular low-rate, Low-Power Wide Area Network (LPWAN) technology providing long range wireless access over unlicensed sub-GHz frequency bands to the Internet of Things (IoT). It has been used in many applications ranging from smart building to smart agriculture. LoRa is a patented modulation. However preliminary reverse-engineering efforts documented parts of it. In this article, we detail the different stages of LoRa transceivers: channel (de)coding, (de)whitening, (de)interleaving and (de)modulation with reverse-engineering in mind. Closed-form expressions for each of these stages are given, and different demodulation and decoding strategies are presented. This allows for a complete modeling of LoRa, which enables Software Defined Radio (SDR) implementations, as well as performance assessment under various channel conditions. These simulations show that LoRa systems have good properties for time and/or frequency selective channels (especially for the latter), thanks to the robustness of its underlying Chirp-Spread Spectrum (CSS) modulation.

1. Introduction

The Internet of Things (IoT) is a general concept of connecting “things” to a global network using standardized protocols and following the principles of the Internet. A thing, or an object, is usually a sensor and/or an actuator that can gather specific data or control a system. There a many applications in which connected objects are used, ranging from entertainment to home automation and industrial applications. For example, for smart buildings, objects can be used to perform metering (water, electricity, etc.), or light and temperature automatic control. In the Industry 4.0, the IoT can be used in closed-loop systems, in order to monitor and possibly control production chains.

While each application has specific requirements, we can still enumerate constraints that are common to most scenarii, such as low energy consumption, low processing capabilities and low memory usage. Thus, objects use wireless communication standards dedicated to IoT. They usually are lossy by design, low data rate, and consume less energy compared to standards of the classical Internet. Several options are available, and the choice is not straightforward. On the one side, there are technologies for short range radios, that allows for communications within a limited area, such as a few tens of meters. On the other side, there are technologies for long range transmissions, able to cover several kilometers of distance. These long range technologies are especially well fitted for smart-city applications, where a whole city can be covered using only a few antennas. LoRa is one of these technologies. It has the advantage of operating within non-licensed bands, which facilitates its deployment.

LoRa is a long range, low data rate and low power technology designed for constrained devices, based on a new modulation which is referred as Chirp Spread Spectrum (CSS) [1]. On top of this physical layer, the LoRaWAN protocol was developed to provide access to the medium. In LoRaWAN, the network is organized in a star topology, with a gateway at the center. The gateway is then connected to a core network via traditional Internet links. The medium access is based on the ALOHA protocol, where devices directly transmit their traffic to gateways, without spectrum sensing nor dedicated time/frequency slots.

LoRa is however a closed physical layer, which prevents studying the technology, and developing various optimizations likely to be proposed by the community. Disposing of a proper description of the modulation chain could enable the research and development of better receivers to achieve better range or higher data rate. Furthermore the ISM band, used by many consumer devices ranging from WiFi-enabled products to wireless doorbell and LoRa IoT devices, is an overcrowded physical resource. In this kind of environment, cognitive radio (*i.e.*, smart radios, able to cooperatively share the spectrum, and minimize energy consumption) is a promising technology. Now, cognitive radios need to be able to (i) sense their environment (spectrum, mainly), (ii) take decisions and (iii) reconfigure their physical layer [2]. While taking decisions relies entirely on software, environment sensing and

[*] Corresponding author.
E-mail addresses: alexandre.marquet@imt-atlantique.fr (A. Marquet), nicolas.montavont@imt-atlantique.fr (N. Montavont), georgios.papadopoulos@imt-atlantique.fr (G.Z. Papadopoulos).

https://doi.org/10.1016/j.comcom.2020.02.034
Received 18 October 2019; Received in revised form 30 January 2020; Accepted 11 February 2020
Available online 14 February 2020
0140-3664/

physical layer reconfiguration typically involve both software and hardware (the radio). This drives a need for very flexible radios, that can be provided with Software Defined Radios (SDR). Indeed, the SDR paradigm aims to perform as much signal processing as possible in software. As a result, within the frequency band reachable by the analog front-end of an SDR, transmitting, receiving and sensing any signal is only a matter of processing power and software, which are very flexible. As a result, a proper SDR implementation of a LoRa transceiver would help research and development of smart radios in the ISM band (either to sense LoRa transmissions, or to communicate using it).

The LoRa physical layer is not described in a standard, and is not published anywhere outside the Semtech patent, that only covers CSS modulation, interleaving (which is close, but not identical to the one actually used in LoRa), and preamble synchronization [1]. To the best of our knowledge, the most significant progress towards a complete characterization of the LoRa physical layer was published by Knight et al. [3] and Robyns et al. [4]. Both provided open-source SDR implementations of LoRa receivers. However, the demodulator adopted in [4] requires high Signal-to-Noise Ratios (SNR), while the implementation of [3] does not work in LoRa implicit mode (no header), and is not maintained since 2017. In both cases, the articles are really focused on implementation. Thus, they do not provide any radio-agnostic evaluation, using either simulations or theoretical development. We believe that only a rigorous modeling will allow research efforts to identify adequate use cases, strong and weak points of the LoRa modulation. Such a modeling will also pave the way for integration of CSS improvements proposed in the literature (*e.g.,* collision mitigation [5,6], spectral efficiency improvement [7]) into later revisions of LoRa, or into competitor systems.

In this article, we propose an analysis of the LoRa physical layer and a proper decomposition of the modulation and demodulation steps, each described using closed-form input–output relationships. To the best of our knowledge, this is the first time that such a detailed description of LoRa transceivers architecture is presented opening, thus, possibilities for independent simulations and implementations, as well as propositions and evaluation of potential improvements (*e.g.,* better channel coding, joint modulation and coding, etc.). We demonstrate the potential of such modeling, through the proposition of several possible demodulation chains for LoRa, and comparison of performance when M-FSK is used instead of CSS. Furthermore, our description of the discrete-time complex baseband equivalent of CSS signals and, more generally, discrete time descriptions of every algorithm of this article makes it a reference for digital implementations using, for instance: SDR, Digital Signal Processor (DSP) or Field Programmable Gate Array (FPGA). In more details, the main contributions of this article are:

- A detailed analysis of the LoRa physical layer, formally describing all the algorithms involved in LoRa transmission chains.
- Methods to reverse-engineer the key-parameters of every algorithms used for LoRa communications.
- Assessment of different receiver chains for LoRa, under various channel conditions (time selective, frequency selective, Rayleigh).
- Comparison of CSS and M-FSK (M-arry Frequency Shift Keying) with various channels.

Performance assessment shows that CSS (and thus, LoRa) is especially good, compared to M-FSK, when faced with frequency-selective channels without mobility. In every other scenarii (AWGN, time-selective, Rayleigh channel), it retains the performance of M-FSK.

The remainder of this article is organized as follows. In Section 2, we first summarize the recent works regarding LoRa reverse-engineering, performance assessment and SDR implementations. Section 3 formally details the LoRa architecture, gives methods to reverse-engineer individual components of the transceiver, to implement them, and then derives a discrete-time complex baseband equivalent of CSS modulated signals. Then, Section 4 gives closed-form expressions for various CSS demodulation strategies. Finally, simulations in Section 5 allow to compare demodulation strategies, as well as to assess CSS and LoRa performance over AWGN, time selective, frequency selective and Rayleigh fading channels.

2. Related work

Most of the literature related to LoRa is actually focused on upper layers (*e.g.,* LoRaWAN), applications and security [8,9]. A fair amount of the research effort is also directed towards the CSS modulations (receiver improvements, new modulations based on it, etc.). However, most likely due to the lack of publicly available hardware, only a few studies have been conducted to describe and analyze, from a theoretical point of view, the LoRa transmission chain. Indeed, between CSS (as the underlying modulation), and LoRaWAN, as the preferred Media Access Control (MAC) protocol, LoRa includes a number of physical layer techniques, namely: channel coding, whitening, and interleaving, that have a strong impact on LoRa system performance. It is worth analyzing and discussing the algorithms used in LoRa, in order to know its actual performance and behavior in details, as well as to propose evolutions or contenders to the technology.

Knight et al. [3] and Robyns et al. [10] wrote the only two reports, to the best of our knowledge, which document a reverse engineering work on LoRa and provide open source implementations. However, Robyns et al. [10] provided a sub-optimal CSS demodulator which requires high SNR. The implementation of Knight et al. [3] uses an optimal demodulator. However, as pointed out by [10] (based on their experiments), Knight et al. wrongly assumed that dewhitening should be performed prior to deinterleaving. Note that this does not prevent them to actually receive and decode LoRa message, thanks to the linearity of whitening and interleaving operations used in LoRa. It does, however, make the retro-engineering process more difficult (for instance, in Knight et al. implementation, different whitening sequences are needed depending on the spreading factor, which is not necessary when deinterleaving is done prior to dewhitening). Other implementations exist, but are not properly documented and hard to set up (*e.g.*, [11]).

Regarding the CSS modulation, Goursaud et al. gave a continuous-time expression of CSS signals in [12]. Vangelista et al. [13] derived the optimal receiver over Additive White Gaussian Noise (AWGN), and Elshabrawy et al. [14] found out that CSS modulation is an instance of orthogonal signaling, and used this result to give closed-form formulas of bit-error probability over AWGN and Rayleigh fading channels. More recently Ghanaatian et al. [15] studied the impact of Carrier Frequency Offset (CFO) and Sampling Frequency Offset (SFO) on CSS signals, using analytical models and simulations over AWGN. They also described methods for estimating CFO and SFO based on the LoRa preamble, as well as compensation techniques to cancel these impairments. Several improvements and variations of CSS have been proposed, mostly focused on spectral efficiency improvements. Bomfin et al. [16] proposed to encode information in the phase of CSS symbols, which is similar to coplanar M-FSK proposed by Roth et al. [17]. Kim et al. [18] proposed to use both up- and down-chirps to encode more information. Vangelista et al. [7] proposed a technique to double the spectral efficiency of CSS, at the price of a 3 dB penalty in SNR.

3. System architecture and reverse-engineering

This section details all the algorithms involved when transmitting and receiving data over LoRa, as originally described in [10] (see Fig. 1). We also provide instructions on how to reverse-engineer key parameters for these algorithms.

The message consists in L bits, and is modeled as a binary vector of independent and identically distributed bits: $\mathbf{b} \in \{0;1\}^L$.

Channel code is introduced to improve reliability to noise. This step introduces redundancy, and yields a (longer) binary sequence denoted $\mathbf{b}^c \in \{0;1\}^{L/\eta}$, with $0 < \eta < 1$ the coding efficiency.

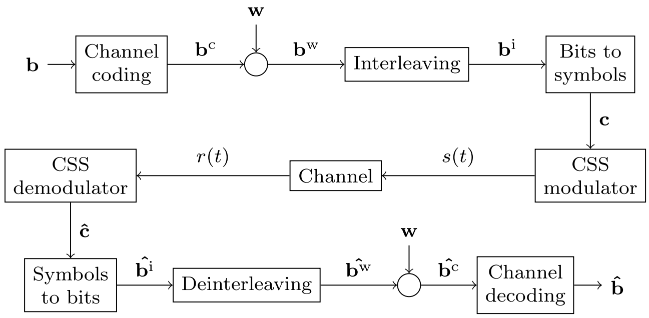


Fig. 1. Architecture of a LoRa transceiver.

Channel coding introduces correlation between bits, which is usually unwanted for designing the receiver. Reducing this correlation is the goal of the whitening algorithm, which computes an exclusive-OR of $\mathbf{b}^{\mathrm{c}}$ with a whitening sequence $\mathbf{w} \in \{0;1\}^{L/\eta}$, which possesses good statistical properties. This results in a sequence of whitened bits $\mathbf{b}^{\mathrm{w}} \in \{0;1\}^{L/\eta}$.

Then, the interleaver scatters the coded sequence in time, in order to reduce the impact of impulsive noise [19, Chap. 14.6]. The output of the interleaver is a deterministic shuffle of the bits in $\mathbf{b}^{\mathrm{w}}$, which is denoted as $\mathbf{b}^{\mathrm{i}} \in \{0;1\}^{L/\eta}$.

Next, groups of SF bits are made in order to produce symbols taking their values in $\mathcal{A} = [0; 2^{\mathrm{SF}} - 1] \subset \mathbb{N}$, yielding a sequence of symbols $\mathbf{c} \in \mathcal{A}^{L/(\mathrm{SF}.\eta)}$. Note that, in the context of CSS systems, SF is referred as the Spreading Factor: the higher the SF, the higher the dispersion of the information carried by each symbol in the time–frequency plane (hence, at fixed bandwidth, a higher SF means a longer time on air). The correspondence between groups of SF bits and values in $\mathcal{A}$ follows a Gray mapping, so that adjacent symbol values encode binary values that only differ by one bit.

The transmission chain ends with the CSS modulator, that translates $\mathbf{c}$ into a physical signal $s(t)$, which is observed as a corrupted signal $r(t)$.

The receiver basically revert every steps done in the transmitter. Individual components of the receiver either produce estimates, denoted with a hat (for example, $\hat{b^{\mathrm{c}}}$ is an estimate of b^{c}), or bitwise Log-Likelihood Ratios (LLR), denoted with an L (for example, $L(b^{\mathrm{c}})$ is the LLR of b^{c}). Let $b \in \{0;1\}$ be a bit, then its LLR $L(b)$ is defined as:

$$L(b) = \ln(\mathbb{P}\{b = 0\}) - \ln(\mathbb{P}\{b = 1\}), \tag{1}$$

where $\ln(x)$ stands for the natural logarithm of $x \in \mathbb{R}^{+}$, and $\mathbb{P}\{A = a\} \in [0;1]$ denotes the probability that a random variable A takes the value a. Note that the LLR of b can be converted back to bit-probability using:

$$\mathbb{P}\{b = x\} = \frac{e^{-xL(b)}}{\left(1 + e^{-L(b)}\right)}, \quad \forall x \in \{0;1\}. \tag{2}$$

In the following subsections, more information on how each step actually works is given, along with the methods used to reverse-engineer their parameters (if needed). Section 3.1 covers channel coding and decoding, and explains how the codes used in LoRa can be retrieved. Two methods of decoding are proposed: hard-value and soft-value decoding, with the later being able to correct more errors, at the price of a higher computational complexity. Next, Section 3.2 gives input–output relationships for the whitening block, whose goal is to transform the output binary sequence of the encoder into a sequence whose values are less correlated. On the receiver side, dewhitening algorithms for hard and soft-values are given, to accommodate the two possible decoding strategies. Here also, a method to discover the whitening sequences used in the LoRa modulation is presented. Just before modulation, the whitened bit-sequence is interleaved in order to reduce the impact of impulsive noise and fadings. The interleaving process, previously described in a graphical way by [10], is given in Section 3.3. The input–output relationships for interleaving and deinterleaving are also formally given, using equations. Similarly to the dewhitening process, the deinterleaving process, in the receiver, is slightly different depending on the use of hard or soft bit values (which, ultimately, define the kind of decoding strategy used). Finally, continuous-time and discrete-time signal models for CSS are given in Section 3.4. Different demodulation strategies will be presented later, in Section 4.

3.1. *Channel coding and decoding*

The goal of any channel code is to make transmissions more robust to noise, introduced by communication channels, using redundancy. In LoRa, the amount of redundancy is controlled by the “coding rate” parameter [20], denoted as $\mathrm{CR} \in [1;4]$ in the following. In [3] and [10], authors have determined that LoRa uses systematic block codes. It produces codewords of $(4+\mathrm{CR})$ bits, using input blocks of 4 bits, giving a code efficiency $\eta = 4/(\mathrm{CR}+4)$.

This subsection is divided into four parts. First, we describe the coding process. Then, we give a method to reverse-engineer the key-parameter of the code: its generator matrix. The last two parts describe hard-value and soft-value decoding strategies, with the former being less performant, but also less computationally demanding than the latter.

3.1.1. *Coding*

Let us denote $\mathbf{b_k}$ the kth block of 4 bits in $\mathbf{b}$, and $\mathbf{b}^{\mathrm{c}}_{\mathbf{k}}$ the associated codeword, containing CR + 4 bits. Then, the coding process can be described by the following equation:

$$\mathbf{b}^{\mathrm{c}}_{\mathbf{k}} = \mathbf{G}\mathbf{b_k}, \tag{3}$$

where $\mathbf{G}$ is the binary generator matrix. Note that (3) and following equations in this part use arithmetics over GF(2), which means that additions use exclusive-OR and multiplications use logical AND. Given that the code is systematic, we have:

$$\mathbf{G} = \begin{pmatrix} \mathbf{P} \\ \mathbf{I}_4 \end{pmatrix}, \tag{4}$$

where $\mathbf{I}_4$ is the (4×4) identity matrix, and $\mathbf{P}$ is a $(\mathrm{CR}\times4)$ binary matrix to be found.

3.1.2. *Reverse-engineering*

Given the small size of $\mathbf{P}$, its value can be easily brute-forced by testing every possible matrix $\mathbf{P}$ against a received sequence $\hat{\mathbf{b}^{\mathrm{c}}}$. Note that this strategy needs working implementations of the dewhitening and the deinterleaving, and assumes an error-free transmission.

3.1.3. *Hard decoding*

With $\mathbf{G}$ completely determined, hard-value maximum likelihood decoding can be efficiently achieved using a syndrome decoder [19, Chap. 7.5].

3.1.4. *Soft decoding*

It is well known that soft-decision decoding (that is, taking into account confidence about bit values) can give substantial processing gain, up to 2 dB [19, Chap. 5.2].

As the LoRa block code acts on blocks of only 4 bits (16 possible codewords), extensive search can be implemented to achieve maximum-likelihood decoding:

$$\hat{\mathbf{b_k}} = \underset{\bar{\mathbf{b}} \in \{0;1\}^4}{\mathrm{argmax}}\, \mathbb{P}\left\{\hat{\mathbf{b}^{\mathrm{c}}_{\mathbf{k}}} = \mathbf{G}\bar{\mathbf{b}}\right\}. \tag{5}$$

Assuming that elements of $\hat{\mathbf{b}^{\mathrm{c}}}$ are independent, we get:

$$\hat{\mathbf{b_k}} = \underset{\bar{\mathbf{b}} \in \{0;1\}^4}{\mathrm{argmax}}\, \mathit{\Pi}_{l=0}^{3+\mathrm{CR}} \mathbb{P}\left\{\hat{b^{\mathrm{c}}}_{(4+\mathrm{CR})k+l} = \left[\mathbf{G}\bar{\mathbf{b}}\right]_l\right\}, \tag{6}$$

where $[\mathbf{Gb}]_l$ denote the lth element of vector $\mathbf{Gb}$. Using formula (2) to convert LLRs to probabilities, discarding the denominator as a constant with respect to $\bar{\mathbf{b}}$, and translating into logarithmic domain, Eq. (6) reduces to:

$$\hat{\mathbf{b}_{\mathbf{k}}} = \underset{\bar{\mathbf{b}}\in\{0;1\}^4}{\operatorname{argmin}} \sum_{l=0}^{3+\mathrm{CR}} \left[\mathbf{G}\bar{\mathbf{b}}\right]_l L\left(b^{\mathrm{c}}_{(4+\mathrm{CR})k+l}\right). \tag{7}$$

Note that implementation of (7) is computationally more involved compared to a syndrome decoder, as operations with floating-point numbers require either more time or more energy than operations over binary values.

3.2. Whitening and dewhitening

By design, a channel encoder adds redundancy bits, which introduces correlation between bits of the coded sequence. This correlation may cause issues with other algorithms in the transmission chain, as they may assume Independent and Identically Distributed (IID) input sequences. Thus, the goal of the whitening operation is to transform the coded sequence, such that individual bits in the whitened sequence are weakly correlated.

This subsection is arranged into four parts. The first one describes the whitening algorithm. Then we provide a method to reverse-engineer the whitening sequences used in the LoRa modulation. Finally, the last two parts describe hard-value (acting on bits) and soft-value (acting on LLR) dewhitening. The latter being required if soft-value decoding is used (see Section 3.1).

3.2.1. Whitening

The whitening operation adds (in $GF(2)$) a known pseudo-random sequence to the output of the channel encoder (coded bits $\mathbf{b}^{\mathrm{c}}$). The input–output relationship is thus given by:

$$\mathbf{b}^{\mathrm{w}} = \mathbf{w} + \mathbf{b}^{\mathrm{c}}. \tag{8}$$

If the whitening sequence $\mathbf{w}$ is carefully chosen, very correlated input sequences (such as the all-one sequence) will result in a pseudo-ransom sequence. The worst case scenario happens when $\mathbf{b}^{\mathrm{c}} = \mathbf{w}$. In order to reduce the probability of this scenario, long whitening sequences should be used.

3.2.2. Reverse-engineering

The whitening sequence can be easily recovered by exploiting the linearity of the channel code. Indeed, according to (3), the codeword associated to the all-zeros block is the all-zeros codeword. So, when sending $\mathbf{b} = \mathbf{0}_{L\times 1}$, the whitened sequence reduces to the whitening sequence: $\mathbf{b}^{\mathrm{w}} = \mathbf{w}$. In this case, assuming that we have a working implementation of the deinterleaver, and an error-free transmission, the whitening sequence can be retrieved as $\mathbf{w} = \hat{\mathbf{b}^{\mathrm{w}}}$.

Note that having error-free transmissions is hard to obtain, which explains why open-source implementations found different whitening sequences [3,10,11].

3.2.3. Hard dewhitening

Thanks to the property that $a + a = 0\ \forall a \in GF(2)$, whitening and dewhitening are similar operations. The only difference being that the later takes a whitened sequence (estimates of the whitened sequence $\hat{\mathbf{b}^{\mathrm{w}}}$, in the case of LoRa) and outputs the original sequence (estimates of the coded sequence $\hat{\mathbf{b}^{\mathrm{c}}}$, here).

3.2.4. Soft dewhitening

By definition of (1), one can notice that the sign of an LLR refers to the value of the estimated bit, while its absolute value refers to the confidence level of that estimation. This means that dewhitening is performed by reversing the sign of LLR at indexes where the bit values of the whitening sequence equals to one.

In practice, on many hardware platform, reversing the sign of a floating-point number is done by flipping the first bit of its binary representation, which means that implementing soft dewhitening has a limited impact compared to hard dewhitening, in terms of complexity.

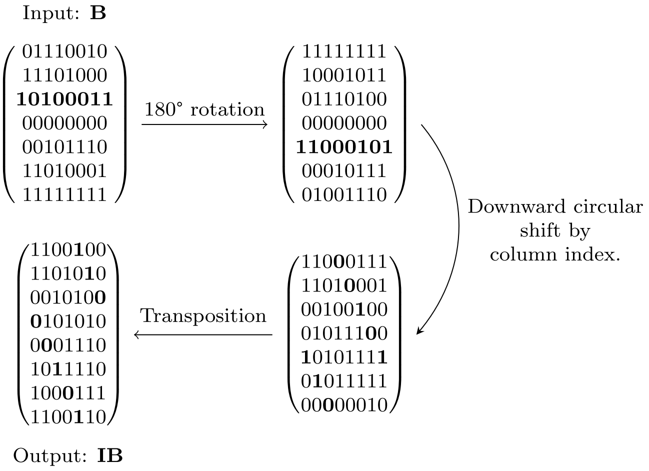


Fig. 2. Decomposition of the interleaving process. Example with SF = 7 and CR = 4.

3.3. Interleaving and deinterleaving

LoRa is an instance of Bit-Interleaved Coded Modulations (BICM), known to be able to provide robust performance under both fading and AWGN channels. With this kind of modulation, interleaving is used to scatter the individual bits of each codeword in time, providing better immunity to noise bursts (or, equivalently, fadings) [19, Chap. 14.6]. The LoRa interleaver belongs to the family of diagonal interleavers. Such interleavers act on blocks of data represented as a matrix. The name comes from the principle of operations: values present on a line of the input matrix end up being scattered on a diagonal of the output matrix.

This subsection is divided into three parts. The interleaving algorithm is given in the first one. Then deinterleaving, which basically consists in re-arranging the received bits back to their initial positions, is described for hard (bits) and soft-values (LLR). Similar to dewhitening, a soft-deinterleaver is necessary to enable soft decoding (see Section 3.1).

3.3.1. Interleaving

The interleaver used in LoRa have successfully been retro-engineered in [10], and is described below. It acts on a block of SF whitened codewords (with size CR + 4 bits), and outputs a block of (CR + 4) × SF interleaved bits, ready to be mapped to CR + 4 symbols. We denote $\mathbf{B}$ the kth block of SF whitened codewords, defined as:

$$\mathbf{B} = \begin{pmatrix} b^{\mathrm{w}}_{k} & \cdots & b^{\mathrm{w}}_{k+(\mathrm{CR}+4)-1} \\ \vdots & \cdots & \vdots \\ b^{\mathrm{w}}_{k+(\mathrm{SF}-1)(\mathrm{CR}+4)} & \cdots & b^{\mathrm{w}}_{k+\mathrm{SF}(\mathrm{CR}+4)-1} \end{pmatrix}, \tag{9}$$

with b^{w}_{k} the kth element of $\mathbf{b}^{\mathrm{w}}$. From there, denoting $[\mathbf{B}]_{i,j}$ the element of $\mathbf{B}$ at line index i and column index j, the kth block of interleaved bits $\mathbf{IB}$ is defined element-wise by:

$$[\mathbf{IB}]_{i,j} = [\mathbf{B}]_{\mathrm{SF}-1-((j-i) \bmod \mathrm{SF}),\,(\mathrm{CR}+4)-1-i} \qquad \forall i \in [0;(\mathrm{CR}+4)-1], j \in [0;\mathrm{SF}-1]. \tag{10}$$

Finally, each line of $\mathbf{IB}$ contains SF interleaved bits to be mapped to symbols later on:

$$\left(b^{\mathrm{i}}_{k+l.\mathrm{SF}} \cdots b^{\mathrm{i}}_{k+(l+1).\mathrm{SF}-1}\right) = \delta_l.\mathbf{IB} \qquad \forall l \in [0;(\mathrm{CR}+4)-1], \tag{11}$$

where δ_l is a line vector of size (CR + 4) whose lth entry has value 1, and every other entry has value 0. This process can be decomposed in several elementary operations, as explained in Fig. 2.

3.3.2. Hard deinterleaving

The deinterleaving process is formally described as follows. The kth matrix of estimated interleaved bits $\hat{\mathbf{IB}}$ is constructed as follows:

$$\hat{\mathbf{IB}} = \begin{pmatrix} \hat{b}^{\mathrm{i}}{}_{k} & \cdots & \hat{b}^{\mathrm{i}}{}_{k+\mathrm{SF}-1} \\ \vdots & \cdots & \vdots \\ \hat{b}^{\mathrm{i}}{}_{k+(\mathrm{CR}+4-1)\mathrm{SF}} & \cdots & \hat{b}^{\mathrm{i}}{}_{k+(\mathrm{CR}+4)\mathrm{SF}-1} \end{pmatrix}. \tag{12}$$

From there, the kth block of SF whitened codewords $\hat{\mathbf{B}}$ is defined element-wise by:

$$\left[\hat{\mathbf{B}}\right]_{i,j} = \left[\hat{\mathbf{IB}}\right]_{(\mathrm{CR}+4)-1-j,\ [\mathrm{SF}-1-i+(\mathrm{CR}+4)-1-j] \bmod \mathrm{SF}} \quad \forall i \in [0; \mathrm{SF}-1], j \in [0; (\mathrm{CR}+4)-1]. \tag{13}$$

Where each line of $\hat{\mathbf{B}}$ is an estimated whitened codeword:

$$\left(\hat{b^{\mathrm{w}}}_{k+l.(\mathrm{CR}+4)} \cdots \hat{b^{\mathrm{w}}}_{k+(l+1).(\mathrm{CR}+4)-1}\right) = \delta_l.\hat{\mathbf{B}} \quad \forall l \in [0; \mathrm{SF}-1]. \tag{14}$$

This deinterleaving process corresponds to the same steps as in Fig. 2, but in reverse.

3.3.3. Soft deinterleaving

Soft deinterleaving is done exactly as hard deinterleaving, the only difference being that $\hat{\mathbf{IB}}$ is constructed using LLR $L\left(b^{\mathrm{i}}\right)$ instead of estimated bits $\hat{b}^{\mathrm{i}}$.

Soft deinterleaving can be done with the same complexity as hard deinterleaving, as long as they use copy-free implementations.

3.4. Chirp spread spectrum modulation

CSS uses linear chirps (see Fig. 3) to spread information carried by each symbol in the time–frequency plane. This allows good resilience to time (as implied, for instance, by Doppler shifts) and frequency selective channels (as implied, for example, by multipath signal propagation) [21]. It is an index-based modulation, which uses the initial frequency shift of the chirp carrying a symbol to encode its value (see Fig. 3) [12].

Let us denote B the bandwidth of the signal, and T the time to transmit a symbol, then $\log_2(B.T) = \mathrm{SF}$ is the spreading factor. As for Direct Sequence Spread Spectrum (DSSS) systems, SF characterizes the increase in bandwidth occupation for a fixed value of T. As seen in the previous subsections, in the case of the LoRa modulation, SF also gives the number of bits carried by a single symbol, meaning that a symbol can take $M = B.T = 2^{\mathrm{SF}}$ values.

Using these notations, we define the expression of the complex baseband equivalent of the base chirp, that is the chirp that encodes the symbol 0. This base chirp $g_0(t)$ has a frequency that linearly grows from $-B/2$ at $t = -T/2$ to $B/2$ at $t = T/2$ (as to span the whole available bandwidth B during the time of a symbol T):

$$g_0(t) = e^{j\pi\frac{B}{T}t^2}\Pi_T(t), \quad \forall t \in \mathbb{R}, \tag{15}$$

where $\Pi_T(t)$ is the rectangular window of size T: $\Pi_T(t) = 1$ for all $t \in [-T/2; T/2[$, and $\Pi_T(t) = 0$ otherwise. In order to modulate a symbol $c \in \mathcal{A}$, the initial frequency of $g_0(t)$ is shifted by $c.B/M = c/T$ Hz, and extra steps are taken to make sure that the chirp is wrapped between $[-B/2; B/2]$ as illustrated in Fig. 3:

$$g_c(t) = \Pi_T(t).e^{j2\pi(\frac{B}{2T}t+\frac{c}{T})t}.e^{-j2\pi lBt}, \quad \forall t \in \mathbb{R}, l \in \mathbb{Z} \quad \text{s.t. } \frac{B}{T}t + \frac{c}{T} \in \left[lB - \frac{B}{2}; lB + \frac{B}{2}\right]. \tag{16}$$

Finally, the modulated signal consists in multiple symbols sent sequentially, with a time spacing of T seconds:

$$s(t) = \sum_{n=0}^{L/(\mathrm{SF}.\eta)} g_{\mathbf{c}_n}(t - nT), \quad \forall t \in \mathbb{R}, \tag{17}$$

where $\mathbf{c}_n \in \mathcal{A}$ denotes the nth element of $\mathbf{c}$.

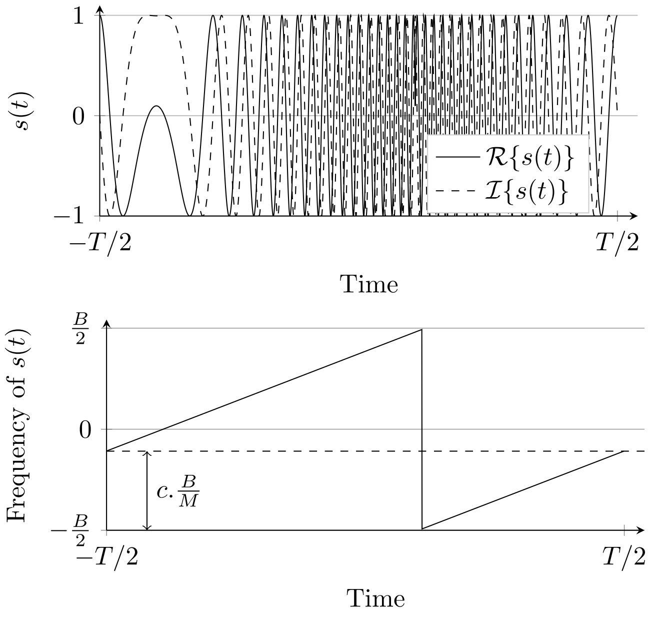


Fig. 3. Transmission of one chirp corresponding to symbol $c = 50$, with M = 128. $\mathcal{R}\{\cdot\}$ and $\mathcal{I}\{\cdot\}$ denote the real part, and the imaginary part, respectively.

To obtain a discrete-time expression for $s(t)$, we sample at the Shannon–Nyquist rate: $F_e = 1/T_e = B$. Consequently, using the relation $B.T = M$, we also have $T = M.T_e$ and $T_e/T = 1/M$. This gives the following discrete-time expression for $s(t)$:

$$s[k] = s(kT_e) = \sum_{n=0}^{L/(\mathrm{SF}.\eta)} g_{\mathbf{c}_n}(kT_e - nT) \tag{18}$$

$$= \sum_{n=0}^{L/(\mathrm{SF}.\eta)} g_{\mathbf{c}_n}\left((k - nM)T_e\right) \tag{19}$$

$$= \sum_{n=0}^{L/(\mathrm{SF}.\eta)} g_{\mathbf{c}_n}[k - nM], \quad \forall k \in \mathbb{Z}. \tag{20}$$

Let us now derive the discrete-time expression of a modulated chirp:

$$g_c[k] = g_c(kT_e) = \Pi_T(kT_e).e^{j2\pi(\frac{B}{2T}kT_e+\frac{c}{T})kT_e}.e^{-j2\pi lBkT_e} \quad \forall k, l \in \mathbb{Z} \quad \text{s.t. } \frac{B}{T}kT_e + \frac{c}{T} \in \left[lB - \frac{B}{2}; lB + \frac{B}{2}\right]. \tag{21}$$

where $\Pi_T(kT_e) = \Pi_{\frac{T}{T_e}}(k) = \Pi_M(k) = \Pi_M[k]$, and $e^{-j2\pi lBkT_e} = e^{-j2\pi lk} = 1\ \forall l, k \in \mathbb{Z}$. Thus, the expression of $g_c(t)$ is simpler in the discrete-time domain:

$$g_c[k] = \Pi_M[k].e^{j2\pi(\frac{B}{2T}kT_e+\frac{c}{T})kT_e} \tag{22}$$

$$= \Pi_M[k].e^{j2\pi\frac{T_e}{T}(\frac{BT_e}{2}k+c)k} \tag{23}$$

$$= \Pi_M[k].e^{j2\pi\frac{1}{M}(\frac{1}{2}k+c)k} \tag{24}$$

$$= \Pi_M[k].e^{j\pi\frac{k^2}{M}}e^{j2\pi\frac{c}{M}k}, \quad \forall k \in \mathbb{Z}, c \in \mathcal{A}. \tag{25}$$

Now, incorporating (25) in (20), we obtain:

$$s[k] = \sum_{n=0}^{L/(\mathrm{SF}.\eta)} \Pi_M[k - nM].e^{j\pi\frac{(k-nM)^2}{M}} .e^{j2\pi\frac{\mathbf{c}_n}{M}(k-nM)} \tag{26}$$

$$= \underbrace{e^{j\pi\frac{k^2}{M}}}_{\text{Chirp}} \underbrace{\sum_{n=0}^{L/(\mathrm{SF}.\eta)} \Pi_M[k - nM].e^{j2\pi\frac{\mathbf{c}_n}{M}k}}_{M\text{-FSK}}, \quad \forall k \in \mathbb{Z}. \tag{27}$$

This shows that CSS signals are basically chirped versions of M-arry Frequency Shift Keying (M-FSK) signals.

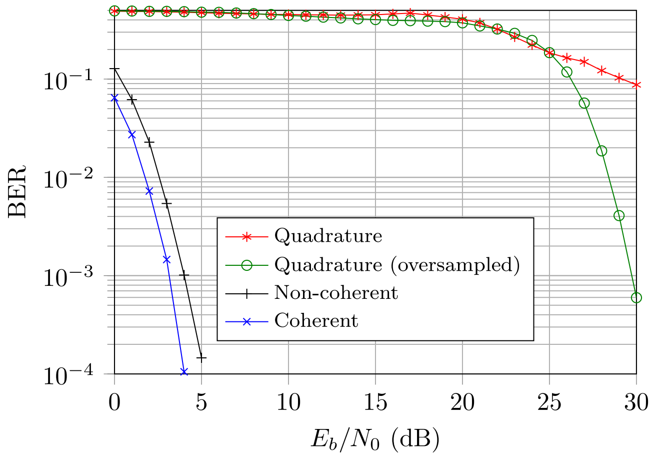


Fig. 4. Comparison of coherent, non-coherent, and quadrature receivers over AWGN with SF = 9.

4. Chirp spread spectrum demodulation

The aim of this section is to rigorously define three demodulations strategies for CSS-modulated signals: quadrature (as used in [7]), coherent and non-coherent receivers. An analysis of their expected behaviors over AWGN, from a theoretical standpoint, will also be given. Also, because the previous section highlighted the potential gains of soft information decoding, techniques for computing bit LLR from coherent and non-coherent demodulators will be given. These strategies will be assessed later on, in Section 5.

In order to do that, this section will be divided in four subsections. The first three sections will cover quadrature demodulation (4.1), coherent demodulation (4.2), and non-coherent demodulation (4.3). Then, Section 4.4 will detail how to extract soft information from coherent and non-coherent demodulators.

In order to present the three demodulation strategies of this part, we consider an AWGN channel:

$$r[k] = s[k] + z[k], \quad \forall k \in \mathbb{Z}, \tag{28}$$

where $z[k]$ is a zero-mean, complex-circular white Gaussian noise.

The three following demodulators use the same dechirping operation, described as:

$$r_d[k] = r[k]e^{-j\pi\frac{k^2}{M}}, \quad \forall k \in \mathbb{Z}, \tag{29}$$

where $r_d[k]$ is the dechirped observed signal. This operation does not change the statistical properties of the noise, thus this system model, involving $r_d[k]$ instead of $r[k]$ is equivalent to M-FSK over AWGN.

4.1. Quadrature demodulation

Quadrature demodulation is a technique that is usually applied to analog Frequency Modulated (FM) signals in order to extract frequency variations. It was retained by Robyns et al. [10] in their SDR implementation, because of its low-complexity. Its use in M-FSK demodulation comes from the fact that these modulations can be seen as frequency modulated M-arry Amplitude Shift Keying (ASK) modulations. Thus, it is actually a two-stage demodulation process. The first stage performs quadrature detection of the frequency variation in $r_d[k]$, and the second performs M-ASK demodulation on this frequency variation.

Quadrature detection reads as follows:

$$\mathcal{Q}\{r_d[k]\} = \arg\left\{r_d[k]r_d^*[k-1]\right\}, \quad \forall k \in \mathbb{Z}, \tag{30}$$

where $(\cdot)^*$ denotes the complex conjugation. As frequency is the derivative of the phase, this detector corresponds to the digital derivative of the phase of the received signal. After quadrature detection, the signal is analogous to an M-ASK modulated signal:

$$\mathcal{Q}\{r_d[k]\} = \frac{2\pi}{M}\sum_{n=0}^{L/(\mathrm{SF}.\eta)} \Pi_M[k-nM]\mathbf{c}_n + z'[k], \quad \forall k \in \mathbb{Z}, \tag{31}$$

but corrupted by a noise $z'[k]$ that has no guarantee to be Gaussian nor white.

Nevertheless, symbols $\mathbf{c}$ can be demodulated with matched filtering:

$$\begin{aligned}\hat{\mathbf{c}}_n &= \frac{1}{2\pi}\left(\Pi_M * r_d\right)[n] && (32)\\ &= \sum_{n'=0}^{L/(\mathrm{SF}.\eta)}\sum_{k=-\infty}^{+\infty} \Pi_M[k-nM]\Pi_M[k-n'M]\mathbf{c}_{n'} && (33)\\ &\quad + (\Pi_M * z')[n] \\ &= M.\mathbf{c}_n + (\Pi_M * z')[n], && (34)\end{aligned}$$

where $\Pi_M[k]$ is the matched filter of itself as it is real and symmetrical. Hard decisions can be taken from $\hat{\mathbf{c}}_n$, by comparing its amplitude to $M-1$ thresholds.

This technique is suboptimal due to the fact that (i) quadrature detection is not an optimal frequency detector in presence of AWGN, and (ii) ASK demodulation is performed using matched filtering (which is a maximum likelihood detector in presence of AWGN) even though the noise $z'[k]$ is not AWGN [19, Chap. 4.4]. However, transposing the demodulation problem from M-FSK to M-ASK allows for the use of standard and well studied techniques for various channel impairments, such as clock drift, CFO and channel selectivity.

4.2. Coherent receiver

Coherent receivers are used when the channel induces no phase impairments, or when such phase impairments are considered perfectly estimated and corrected. Over perfect AWGN channels, it allows for maximum likelihood detection. Its use is mandatory for demodulating derivative of CSS that use the phase to encode symbols (see, *e.g*, [16]).

A maximum likelihood demodulation strategy for M-FSK in presence of AWGN is to find the symbol $c \in \mathcal{A}$ whose associated signal $e^{2\pi\frac{c}{M}k}$ correlates the best with the observed signal $r_d[k]$ [19, Chap. 4.4]. This is expressed as follows:

$$\hat{\mathbf{c}}_n = \underset{c\in\mathcal{A}}{\operatorname{argmax}}\,\mathcal{R}\left\{\sum_{k=-\infty}^{+\infty} \Pi_M[k]e^{-2\pi\frac{c}{M}k} r_d[k+nM]\right\}. \tag{35}$$

One can observe that the term in the real part is actually the M point Discrete Fourier Transform (DFT), denoted as $\mathcal{F}_M\{\cdot\}[\cdot]$, of the received signal:

$$\hat{\mathbf{c}}_n = \underset{c\in\mathcal{A}}{\operatorname{argmax}}\,\mathcal{R}\left\{\mathcal{F}_M\{r_d[k+nM]\}[c]\right\}, \tag{36}$$

which can be efficiently implemented using the Fast Fourier Transform (FFT) algorithm [22].

Expanding the correlation term in (36) allows to understand why M-FSK modulations are effective in low SNR conditions, especially as M grows bigger:

$$\begin{aligned}&\mathcal{F}_M\{r_d[k-nM]\}[c'] && (37)\\ &\quad = \mathcal{F}_M\{s[k-nM]\}[c'] + \mathcal{F}_M\{z[k]\}[c'] \\ &\quad = \sum_{n'=0}^{L/(\mathrm{SF}.\eta)} \mathcal{F}_M\{\Pi_M[k-(n-n')M].e^{j2\pi\frac{\mathbf{c}_n}{M}k}\}[c'] && (38)\\ &\qquad + \mathcal{F}_M\{z[k]\}[c'] \\ &\quad = M.\delta_{n-n'}\delta_{\mathbf{c}_n-c'_n} + \mathcal{F}_M\{z[k]\}[c'], && (39)\end{aligned}$$

where $\delta_n = 1$ if $n = 0$ and zero otherwise. As we can see, the correlator output corresponding to the symbol that was sent has its amplitude augmented by M, as opposed to the other correlators. This means that this modulation, along with coherent demodulation gets more robust to noise as M increases. However, as $B.T = M$, this comes at the expense of either an increased bandwidth or a decreased symbol rate.

4.3. Non-coherent receiver

While effective, the coherent receiver is very sensitive to channel impairments other than white noise. In particular, due to the real part in (36), any channel impairment that translates to a phase shift in (37), results in a loss of reliability.

Non-coherent detection is a computationally efficient way to tackle this issue, as it only replaces the real part operator by the magnitude operator in (36):

$$\hat{\mathbf{c}}_n = \underset{c\in\mathcal{A}}{\operatorname{argmax}} \left|\mathcal{F}_M\{r_d[k+nM]\}[c]\right|. \tag{40}$$

However, this comes at the expense of a reduced noise immunity.

4.4. Soft-decisions for coherent and non-coherent receivers

In order to leverage the potential gains of soft-decision decoding (see Section 3.1), the demodulator must be altered, as to compute LLR for each bit of a codeword:

$$L\left(b_n^{\mathrm{i}}\right) = \ln \frac{\mathbb{P}\left\{b_n^{\mathrm{i}} = 0|r_d(t)\right\}}{\mathbb{P}\left\{b_n^{\mathrm{i}} = 1|r_d(t)\right\}}. \tag{41}$$

Under the assumption that transmitted symbols are IID, assuming that b_n^{i} is carried by the lth bit of the nth symbol, and using Bayes rule, we get:

$$L\left(b_n^{\mathrm{i}}\right) = \ln \frac{\sum_{c\in\mathcal{A}\text{ s.t. }b_l(c)=0} \mathbb{P}\left\{r_d(t)|c_n = c\right\}}{\sum_{c\in\mathcal{A}\text{ s.t. }b_l(c)=1} \mathbb{P}\left\{r_d(t)|c_n = c\right\}}, \tag{42}$$

where $b_l(c)$ denotes the value of the lth bit mapped to symbol c. On an AWGN channel, likelihoods $\mathbb{P}\left\{r_d(t)|c_n = c\right\}$ are proportional to $e^{\mathcal{R}\{\mathcal{F}_M\{r_d[k+nM]\}[c]\}/\sigma^2}$, where σ^2 denotes noise power [17]. Thus, LLR for coherent detection are finally expressed as:

$$\begin{aligned} L\left(b_n^{\mathrm{i}}\right) = &\max^*_{c\in\mathcal{A}\text{ s.t. }b_l(c)=0} \mathcal{R}\left\{\mathcal{F}_M\{r_d[k+nM]\}[c]\right\}/\sigma^2 \\ &- \max^*_{c\in\mathcal{A}\text{ s.t. }b_l(c)=1} \mathcal{R}\left\{\mathcal{F}_M\{r_d[k+nM]\}[c]\right\}/\sigma^2, \end{aligned} \tag{43}$$

where $\max^*$, defined as $\max^*_{c\in\mathcal{A}} c = \ln\left(\sum_{c\in\mathcal{A}} e^c\right)$, has several efficient implementations, and can be approximated as $\max^* \approx \max$, in which case σ^2 can be omitted from (43) [23].

Similar to Section 4.3, the real part in (43) results in a loss of reliability for any channel impairment that translated to a phase shift in (37). This can be efficiently circumvented using the magnitude operator instead of the real part:

$$\begin{aligned} L\left(b_n^{\mathrm{i}}\right) = &\max^*_{c\in\mathcal{A}\text{ s.t. }b_l(c)=0} \left|\mathcal{F}_M\{r_d[k+nM]\}[c]\right|/\sigma^2 \\ &- \max^*_{c\in\mathcal{A}\text{ s.t. }b_l(c)=1} \left|\mathcal{F}_M\{r_d[k+nM]\}[c]\right|/\sigma^2, \end{aligned} \tag{44}$$

which also comes at the expense of a reduced noise immunity.

Computing LLR from quadrature detection would also be possible. Although the non-gaussianity of the noise at the output of the matched filter (32) would make the computation of exact LLR difficult, it would still be possible to get an approximation using standard techniques used for M-ASK over AWGN. However, given the very poor performance of quadrature detection compared to coherent and non-coherent demodulation of CSS signals (see Fig. 4), we chose not to study its performance in the context of LoRa demodulation chains.

5. System simulations

In the following, we assess the performance of CSS, M-FSK, LoRa and FSK-LoRa over four channels: AWGN, frequency selective, time selective and Rayleigh. FSK-LoRa designates a LoRa system (as described in Fig. 1) in which the CSS modulator/demodulator blocks are replaced by M-FSK ones.

We implemented channel coding, whitening and interleaving as described in previous sections using GNURadio. The associated source code is available in [24].

Most of the simulations presented here present performance assessed with respect to the per-bit SNR of the transmission, denoted as E_b/N_0. SNR and per-bit SNR are related by the spectral efficiency α (in bits/s/Hz) of the modulation:

$$\mathrm{SNR} = \alpha E_b/N_0. \tag{45}$$

In the case of CSS and M-FSK, spectral efficiency is given by $\alpha = \mathrm{SF}/M$. In the case of LoRa, spectral efficiency is given by $\alpha = \eta\mathrm{SF}/M = 4.\mathrm{SF}/(M.(\mathrm{CR}+4))$. On the one hand, using E_b/N_0 allows for fair comparison of different modulations, and demodulation methods, as it considers the amount of energy dedicated to the transmission of each bit [19, Chap. 4.2.9]. On the other hand, SNR is more meaningful when one wants to evaluate the performance (the range, in particular) of actual systems, as typical transmitters are limited in terms of signal power (not in terms of per-bit energy).

5.1. Demodulation methods

We first compare the three demodulation techniques presented in Section 4 on Fig. 4. As expected, quadrature demodulation has the worst performance, which can however be improved by oversampling the received signal. However, even with an oversampling factor of 2, quadrature demodulation suffers from an E_b/N_0 difference as high as 23 dB for a BER of 10^{-4} with respect to coherent demodulation.

The difference between coherent and non-coherent demodulation is much less, in the order of magnitude of 1 dB.

5.2. Performance over AWGN channel

AWGN channel is an idealized channel where propagation would be perfect, and only additive noise (mainly thermal noise in the receiver electronics) would affect the emitted signal. It is nevertheless a commonly used base for comparison of different modulations.

As CSS and M-FSK belong to the same modulation family (orthogonal modulations), they should perform exactly the same over an AWGN channel, which is confirmed by Fig. 5.

Fig. 5b shows that coherent demodulation and soft-decoding allow for processing gain of approximately 1 dB and 1.5 dB, respectively. This means 2.5 dB of difference between the most complex and less complex implementation of the receiver.

Comparing Figs. 5a and 5b we notice that the extra processing steps introduced by LoRa, with respect to CSS and M-FSK, does not significantly improve the efficiency of the transmission, at least for $\mathrm{SF} = 9$ and $\mathrm{CR} = 4$, on an AWGN channel.

5.3. Performance over frequency-selective channels

Frequency-selective channels are representative of multipath propagation, where the propagation paths does not vary with time, or when this variation is negligible during the time of the transmission. Rural scenarii where both the transmitter and the receiver are not mobile are well modeled with frequency-selective channels.

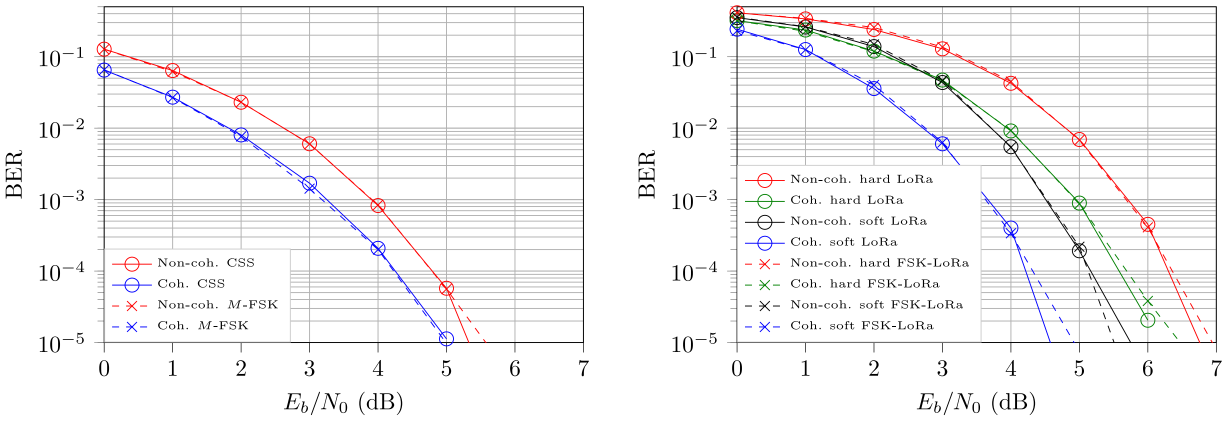
In this subsection, we consider the Proakis B [19, chap. 9.4] static, frequency-selective channel with impulse response $h[k] = 0.407\delta_{k-1} + 0.815\delta_k + 0.407\delta_{k+1}$, (with $\delta_k = 1$ if $k = 0$ and $\delta_k = 0$ otherwise). The received signal is given by:

$$r[k] = \sum_{l=-\infty}^{\infty} h[l]s[k-l] + z[k]. \tag{46}$$

This channels is non-invertible, which means that part of the information carried by $s[k]$ is lost during the transmission (see Fig. 7).

As we can see on Fig. 6 CSS performs much better than M-FSK on this channel, due to its better spreading of the information in frequency.

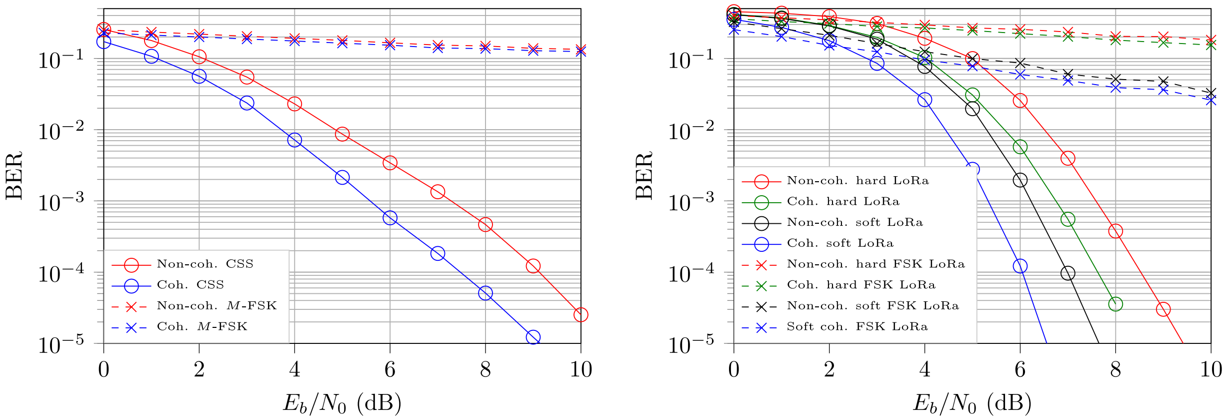
Comparing Figs. 6a and 6b, we observe that the LoRa processing chain allows for a steeper slope of the BER curve. This allows LoRa to outperform CSS at $E_b/N_0 = 5$ dB with soft decoding (coherent demodulation, 6 dB for non-coherent) and $E_b/N_0 = 8$ dB with hard decoding (for both coherent and non-coherent demodulation).



(a) Benefits of coherent demodulation on CSS and M-FSK.

(b) Benefits of coherent demodulation and soft decoding on LoRa and FSK-LoRa, with CR = 4.

Fig. 5. Simulations over an AWGN channel, with SF = 9. CSS and M-FSK are exactly equivalent in this scenario.



(a) Benefits of coherent demodulation on CSS and M-FSK.

(b) Benefits of coherent demodulation and soft decoding on LoRa and FSK-LoRa, with CR = 4.

Fig. 6. Simulations over a Proakis B frequency-selective channel, with SF = 9. Here, CSS performs much better than M-FSK, thanks to the symbol-level frequency diversity induced by the chirp.

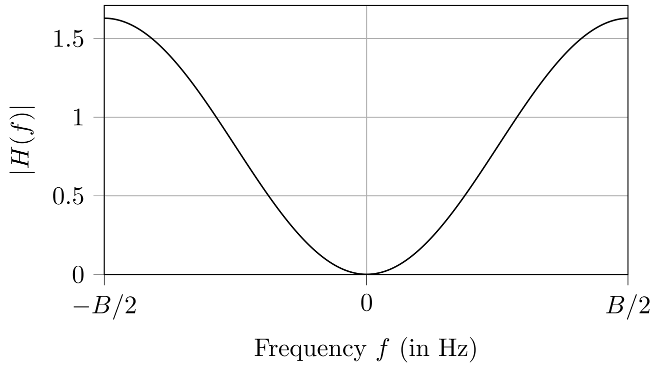


Fig. 7. Frequency-domain response of the Proakis B channel [19, chap. 9.4]. Signal is completely lost at the center of the occupied bandwidth, and suffers from important attenuation around it.

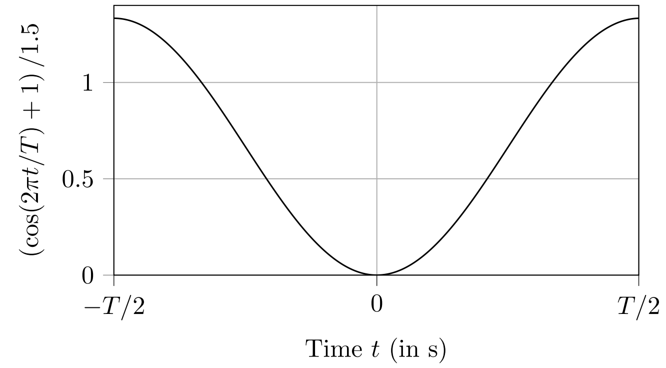


Fig. 8. Time-domain response of the time-selective channel used in this article. The central portion of each symbol suffers from very high attenuation, while signal at the exact center of it is completely lost.

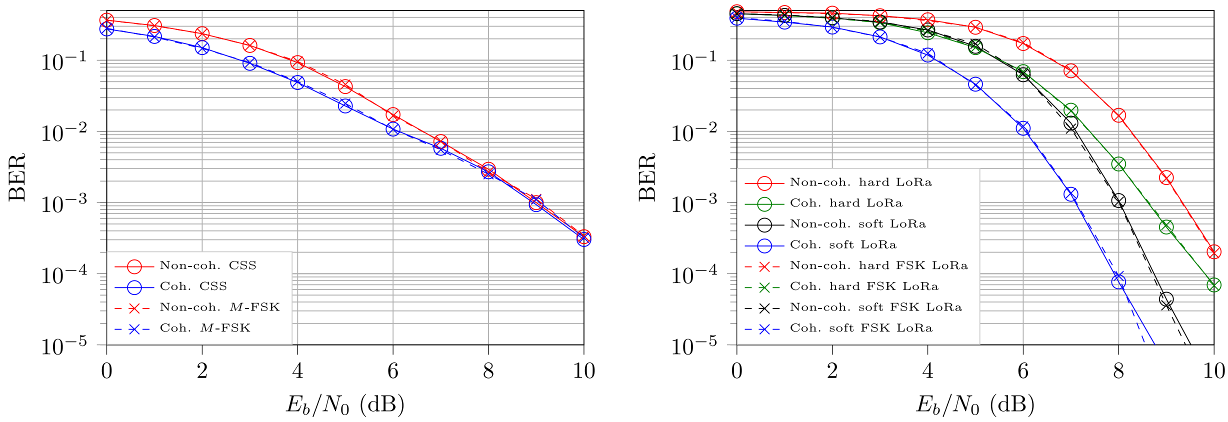
5.4. Performance over time-selective channels

Time-selective channels are representative of propagation suffering from Doppler only. That is, situation where the emitter and receiver are mobile, but there are no obstacle leading to multipath propagation. This is a very unlikely situation, and we study this channel mainly to assess the limits of LoRa and CSS when faced with mobility.

Here, we consider a cosined-shaped time-selective channel:

$$r[k] = \left[(\cos(2\pi k/M) + 1)/1.5\right] s[k] + z[k]. \tag{47}$$

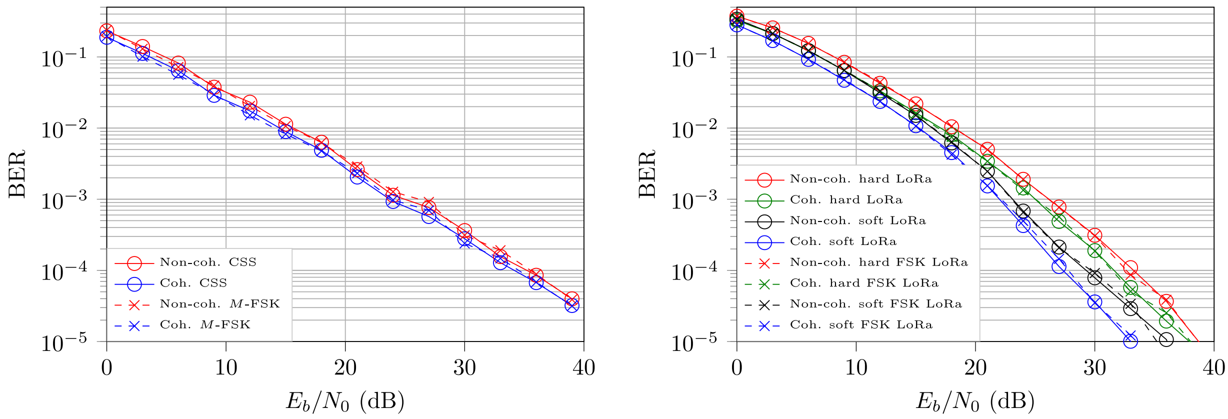
This channels is also non-invertible (see Fig. 8).



(a) Benefits of coherent demodulation on CSS and M-FSK.

(b) Benefits of coherent demodulation and soft decoding on LoRa and FSK-LoRa, with CR = 4.

Fig. 9. Simulations over a time selective channel, with SF = 9. Here, M-FSK and CSS performs equally. One can notice that the processing gain due to coherent receiving vanishes as E_b/N_0 goes higher.



(a) Benefits of coherent demodulation on CSS and M-FSK.

(b) Benefits of coherent demodulation and soft decoding on LoRa and FSK-LoRa, with CR = 4.

Fig. 10. Simulations over an ITU indoor to outdoor and pedestrian Rayleigh channel, with SF = 9. Here also, CSS and M-FSK have similar performances.

Fig. 9 shows that CSS (LoRa) and M-FSK (FSK-LoRa) are equivalent in this scenario. Interestingly, we notice on Fig. 9a that coherent and non-coherent demodulation tend to be equivalent as E_b/N_0 increases. We notice the same behavior on Fig. 9b, but it happens at higher values of E_b/N_0.

Comparing Figs. 6a and 6b, we observe that the LoRa processing chain allows for a steeper slope of the BER curve. This allows LoRa to outperform CSS at $E_b/N_0 = 6$ dB with soft decoding (coherent demodulation, 8 dB for non-coherent) and $E_b/N_0 = 9$ dB with hard decoding (coherent demodulation, 10 dB for non-coherent).

5.5. Performance over Rayleigh channels

Rayleigh channels are representative of propagation suffering from multipath and mobility, inducing random fadings in the received signal. Here, we use the ITU outdoor to indoor and pedestrian A channel, taken from [25, Table 4] (see Table 1). It models communication between a device located outdoor (a gateway, typically) and a device located indoor or in a street (a smart sensor, for instance).

To fit this description, we configured a maximum Doppler shift corresponding to a typical walking speed: $v = 3$ km/h ($v \approx 0.83$ m/s). The other parameters are chosen to match a LoRa transmission in the 868 MHz band, that is, a bandwidth $B = 125$ kHz, and a carrier frequency $f_0 = 868.1$ MHz. The normalized maximum Doppler frequency is then computed as $f_d = f_0.v/(c.B)$, with $c \approx 3.10^8$ m/s the speed of light in a vacuum. The received signal is expressed as:

$$r[k] = \sum_{l=-\infty}^{\infty} h_k[l]s[k-l] + z[k], \tag{48}$$

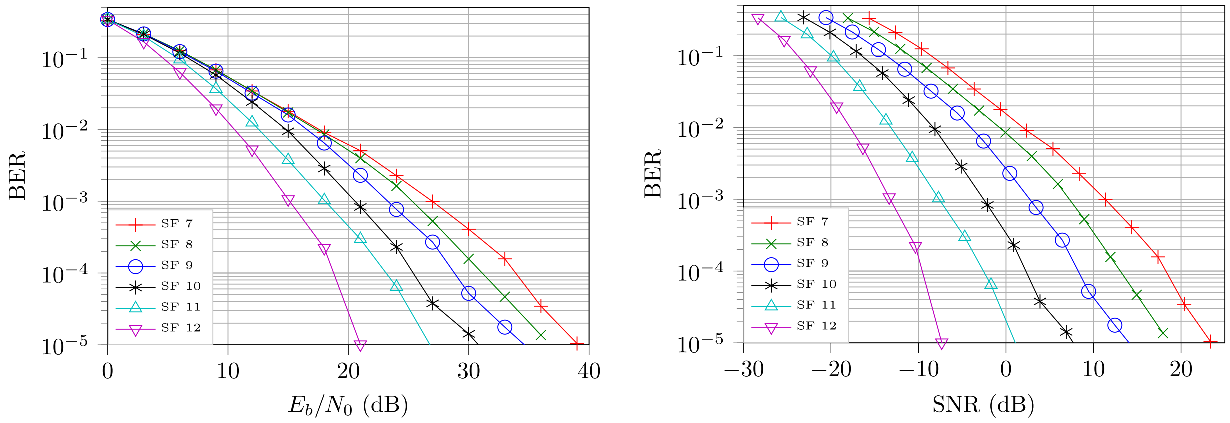
where $h_k[l]$ is the time-varying impulse response of the channel, given by:

$$h_k[l] = \sum_{n=0}^{N_{\text{taps}}-1} A[n]\alpha_k[n]\text{sinc}(\tau_N[n] - l), \tag{49}$$

with N_{taps} the number of path in the Rayleigh channel model, $\tau_N[n]$ $\forall n \in [0; N_{\text{taps}} - 1]$ is the relative normalized delay of path n, and $A[n]$ $\forall n \in [0; N_{\text{taps}} - 1]$ is the average power of path n. $\alpha_k[n]$ is the time-varying complex gain, and is computed using the sum-of-sinusoids technique (with 8 sinusoids) described in [26]. In our simulations, these computations are handled by the GNURadio block `selective_fading_model`.

Note that, for coherent demodulation to be effective in this scenario, an estimator of the phase shift introduced by the channel is necessary. As this article does not cover such an estimator, we computed this phase-shift with knowledge of the simulated channel (more precisely, we compare the phase of the signal before and after passing through the Rayleigh channel, before adding AWGN).

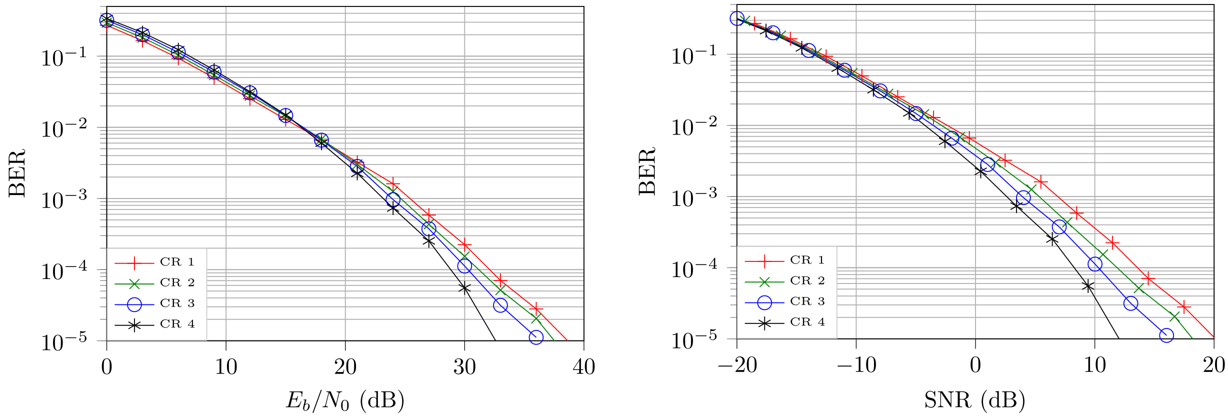
Fig. 10 shows that CSS (LoRa) and M-FSK (FSK-LoRa) are also equivalent in this scenario. On Fig. 10b, one can observe that the



(a) As a function of per-bit SNR.

(b) As a function of SNR.

Fig. 11. LoRa simulations over an ITU indoor to outdoor and pedestrian Rayleigh channel, with various SF and CR = 4.



(a) As a function of per-bit SNR.

(b) As a function of SNR.

Fig. 12. LoRa simulations over an ITU indoor to outdoor and pedestrian Rayleigh channel, with various CR and SF = 9.

Table 1
ITU outdoor to indoor and pedestrian A channel tapped-delay-line parameters (from [25, Table 4]). Normalized delays τ_N are computed from delays τ as $\tau_N = \tau/T_c = \tau.B$, with $B = 125$ kHz.

Tap n	Relative delay ($\tau[n]$, ns)	Relative normalized delay ($\tau_N[n]$)	Average power ($A[n]$, dB)
0	0	0	0
1	110	0.01375	−9.7
2	190	0.02375	−19.2
3	410	0.05125	−22.8

processing gains allowed by coherent demodulation and soft-decoding are up to 5 dB. Here LoRa outperforms CSS at $E_b/N_0 = 21$ dB with soft decoding (same as non-coherent) and $E_b/N_0 = 27$ dB with hard decoding (same as non-coherent).

Figs. 11 and 12 show the impact of the spreading factor and the coding rate, respectively, on the transmission robustness. Notice how increasing the spreading factor by one results in an SNR gain of approximately 5 dB (3 dB in terms of E_b/N_0). Similarly, consecutive CR introduces a gain of around 2 dB of SNR (1 dB in terms of E_b/N_0). Hence, in terms of BER performance, varying the spreading factor leads to the most significant gains. However, increasing SF by one doubles the time on air. Thus, when considering other constraints such as the time on air, and the duty cycle, as enforced in the ISM bands, varying the coding rate can be a better option, depending on the application. This shows how an increase in spreading factor and/or coding rate CR can help cover longer ranges. However, this comes at the price of a reduced spectral efficiency and, thus, reduced bitrate, for a fixed bandwidth).

6. Conclusion

In this article, we presented the LoRa modulation architecture. We discussed channel coding, whitening and interleaving with reverse-engineering in mind. In particular, we identified the challenges for determining correct whitening sequences, and gave closed-form expression for interleaving and deinterleaving. We derivated the discrete-time expression of the CSS modulated signal, and used it as a basis to express and study several demodulation strategies. By means of simulation, we showed that quadrature demodulation of CSS modulated signal is outperformed by coherent and non-coherent receivers. We confirmed that CSS modulation has good properties for transmissions over time and frequency selective channels (especially for the latter).

Finally, we observed that the three extra stages used in LoRa can only give tangible benefits if coherent demodulation and soft decoding are used. These are computationally more expensive, and may not be implemented in the most energy constrained devices. They are however very likely to be implemented in base stations.

All simulations have been implemented in GNURadio, which is an SDR framework [24]. However, this implementation is not able to reliably demodulate real LoRa signal yet. Indeed, LoRa demodulation still presents open challenges. In particular, there is no dedicated literature covering algorithms for time and frequency shift tracking for the CSS modulation. Without such algorithms, reliable decoding of LoRa transmissions, especially with average quality radios (such as the ones commonly used in SDR) and long packets, cannot be made reliably, even at high SNR.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.comcom.2020.02.034.

CRediT authorship contribution statement

Alexandre Marquet: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Nicolas Montavont:** Writing - original draft, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Georgios Z. Papadopoulos:** Writing - original draft, Writing - review & editing, Supervision, Funding acquisition.

References

[1] O. Seller, N. Sornin, Low power long range transmitter, 2014.

[2] C. Moy, High-level design approach for the specification of cognitive radio equipments management apis, J. Netw. Syst. Manage. 18 (1) (2010) 64–96, http://dx.doi.org/10.1007/s10922-009-9151-3.

[3] M. Knight, B. Seeber, Decoding LoRa: Realizing a Modern LPWAN with SDR, in: Proceedings of the GNU Radio Conference, Vol. 1 (1).

[4] P. Robyns, P. Quax, W. Lamotte, W. Thenaers, Gr-lora: An efficient lora decoder for GNU radio, 2017, http://dx.doi.org/10.5281/zenodo.892174.

[5] R. Eletreby, D. Zhang, S. Kumar, O. Yaundefinedan, Empowering low-power wide area networks in urban settings, in: Proceedings of the Conference of the ACM Special Interest Group on Data Communication, SIGCOMM '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 309–321, http://dx.doi.org/10.1145/3098822.3098845.

[6] U. Noreen, L. Clavier, A. Bounceur, Lora-like css-based phy layer, capture effect and serial interference cancellation, in: European Wireless 2018; 24th European Wireless Conference, 2018, pp. 1–6.

[7] L. Vangelista, A. Cattapan, A new lora-compatible modulation improving the lorawan network level performance, in: 2019 IEEE Latin-American Conference on Communications (LATINCOM), 2019, pp. 1–6, http://dx.doi.org/10.1109/LATINCOM48065.2019.8937880.

[8] J.P.S. Sundaram, W. Du, Z. Zhao, A survey on lora networking: Research problems, current solutions and open issues, IEEE Commun. Surv. Tutor. (2019) 1, http://dx.doi.org/10.1109/COMST.2019.2949598.

[9] The things network, 2019, URL https://www.thethingsnetwork.org/docs/lorawan/academic.html.

[10] P. Robyns, P. Quax, W. Lamotte, 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: IoTBDS, INSTICC, SciTePress, 2018, pp. 41–51, http://dx.doi.org/10.5220/0006668400410051.

[11] R.F. Myriad, Myriad RF an SDR lora implementation for R & D, commit 52ec64d, 2018, URL https://github.com/myriadrf/LoRa-SDR/.

[12] C. Goursaud, J.-M. Gorce, Dedicated networks for IoT : PHY / MAC state of the art and challenges, EAI endorsed transactions on Internet of Things http://dx.doi.org/10.4108/eai.26-10-2015.150597.

[13] L. Vangelista, Frequency shift chirp modulation: The lora modulation, IEEE Signal Process. Lett. 24 (12) (2017) 1818–1821, http://dx.doi.org/10.1109/LSP.2017.2762960.

[14] T. Elshabrawy, J. Robert, Closed-form approximation of lora modulation BER performance, IEEE Commun. Lett. 22 (9) (2018) 1778–1781, http://dx.doi.org/10.1109/LCOMM.2018.2849718.

[15] R. Ghanaatian, O. Afisiadis, M. Cotting, 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, http://dx.doi.org/10.1109/ICASSP.2019.8683504.

[16] R. Bomfin, M. Chafii, G. Fettweis, A novel modulation for iot: Psk-lora, in: 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), 2019, pp. 1–5, http://dx.doi.org/10.1109/VTCSpring.2019.8746470.

[17] Y. Roth, J.-B. Doré, L. Ros, V. Berg, Contender waveforms for low-power wide-area networks in a scheduled 4g OFDM framework, EURASIP J. Adv. Signal Process. 2018 (2018) 43, http://dx.doi.org/10.1186/s13634-018-0566-4.

[18] K. Kim, S. Lee, Y. Shin, Spectral efficiency improvement of chirp spread spectrum systems, in: 2019 International Conference on Information and Communication Technology Convergence (ICTC), 2019, pp. 1085–1087, http://dx.doi.org/10.1109/ICTC46691.2019.8939967.

[19] J. Proakis, M. Salehi, Digital Communications, McGraw-Hill International Edition, McGraw-Hill, 2008.

[20] Semtech, sx1276/77/78/79 datasheet, rev. 5 edition, 2016.

[21] Semtech, an1200.22 lora modulation basics, rev. 2 edition, 2015.

[22] R. Singleton, A short bibliography on the fast fourier transform, IEEE Trans. Audio Electroacoust. 17 (2) (1969) 166–169, http://dx.doi.org/10.1109/TAU.1969.1162040.

[23] P. Robertson, P. Hoeher, E. Villebrun, Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding, Eur. Trans. Telecommun. 8 (2) (1997) 119–125, http://dx.doi.org/10.1002/ett.4460080202.

[24] A. Marquet, Gnuradio implementation of a lora transceiver, 2019, http://dx.doi.org/10.5281/zenodo.3630182.

[25] International Telecommunication Union Radiocommunication Sector, Guidelines for evaluation of radio transmission technologies for IMT-2000, Recommendation, ITU-R, 1997.

[26] M. Patzold, C. Wang, B.O. Hogstad, Two new sum-of-sinusoids-based methods for the efficient generation of multiple uncorrelated rayleigh fading waveforms, IEEE Trans. Wireless Commun. 8 (6) (2009) 3122–3131, http://dx.doi.org/10.1109/TWC.2009.080769.