Daniel Szafranski and Andreas Reinhardt. “ELORA: Even Longer Range Sensor Networking Through Modulated Concurrent LoRa Transmissions.” In: *Proceedings of the 1st IEEE Workshop on Wireless outdoor, Long-Range and Low-Power Networks (WoLoLo)*. 2023, pp. 388–393.

ELORA: Even Longer Range Sensor Networking Through Modulated Concurrent LoRa Transmissions

Daniel Szafranski
Department of Informatics
Clausthal University of Technology, Germany
daniel.szafranski@tu-clausthal.de

Andreas Reinhardt
Department of Informatics
Clausthal University of Technology, Germany
reinhardt@ieee.org

***Abstract*— LoRa is a widely used physical layer for Low Power Wide Area Networks (LPWANs). Its comparably low data rates still suffice for many IoT applications, while communication ranges of several kilometers can be accomplished. Due to their wireless nature, LoRa data transmissions become unreliable when the distance between sender and receiver approaches the limit at which the signal-to-noise ratio falls below the receiver's sensitivity. Furthermore, reliability can also be hampered by rough environmental conditions, like heavy rain or high humidity levels, both of which can have attenuating effects on the wireless signal propagation. While the Forward Error Correction in LoRa can compensate for sporadic bit errors, its overall efficacy is limited. To prevent data loss even in adverse weather conditions, alternative methods need to be found to maintain the reliability of communications. One such approach is the use of Constructive Interference, i.e., the concurrent transmission of identical LoRa packets by multiple transmitting stations. As the required clock synchronization is hard to accomplish, however, fully constructive LoRa data transmissions are virtually impossible in practice. In this paper, we hence present two modulation techniques (based on On–Off Keying and Pulse Width Modulation) that take advantage of constructive interference periods occurring in concurrent transmissions to increase the range and thus make a Wireless Sensor Network better usable in emergency scenarios. We present their implementation on actual LoRa transceivers and analyze the resulting signal's characteristics. Our first experimental results prove the viability of our concept, and show that data can be modulated and demodulated without the knowledge of the actual LoRa message.**

***Index Terms*—LoRa, Long Range, Constructive Interference, Concurrent Transmission, Modulation, Software Defined Radio, On-Off Keying, Pulse Width Modulation, Collisions**

I. INTRODUCTION

Thanks to its energy efficiency and long-range capabilities, the LoRa PHY is widely used as the physical layer for LPWANs. LoRa-based Wireless Sensor Networks (WSNs) are often deployed in outdoor scenarios where the distance between individual nodes may approach the physical limit of the transmission range. Outdoor deployments also expose WSNs to the (possibly harsh) physical environment. Communicating across long distances under rough environmental conditions typically has a negative effect on the packet reception rate (PRR). In very challenging cases, it may even not be possible to exchange data at all. If a WSN is, e.g., used for the analysis or prediction of weather extremes, yet rough environmental conditions like heavy rain impede the wireless transmissions, important data cannot be received anymore, even though environmental data under changing weather conditions is of particularly high importance. Hence reliability has a high priority in WSNs, especially when used in mission-critical applications.

In order to increase the reliability in LoRa-based WSNs, several well known methods have already been researched and published (cf. Section II). One very promising approach is the concept of Constructive Interference (CI), which is based on the superposition of simultaneously transmitted Radio Frequency (RF) signals, leading to an increase in received signal amplitude and thus a greater transmission range. Previous works (e.g., [1–4]) have investigated the application of this concept in LoRa. It was found that CI can indeed lead to an increased signal amplitude and thus a potentially larger reception range. However, it was also determined that this is not reliably achievable in real world scenarios due to the imperfect clock synchronization between devices. Thus, constructively overlapping waveforms always occur in alternation with destructive interference, leading to interference patterns that are referred to as the *beating effect* in related literature [2]. Due to this interference effects, the potential of CI cannot be fully utilized, and moreover the demodulation of the resulting LoRa packets with pulsating signal strength becomes more difficult, error-prone, or even impossible.

In this paper, we present two alternative approaches of utilizing CI by taking advantage of the signal periods with higher amplitudes but without trying to demodulate the actual content of the LoRa packet. The main idea behind our approach is, that even if it is impossible to demodulate (constructively or destructively) interfering LoRa packets, we can determine (a) whether a LoRa signal is present or not, and (b) the durations of the signals as well as the time in-between them. Using this information we can build a modulator and demodulator based on the well known On–Off Keying (OOK) and Pulse Width Modulation (PWM) techniques. Our approach is intended to be used as a fallback modulation scheme with limited data rate, when no direct communication via LoRa is possible.

II. RELATED WORK

Due to the potential of CI, concurrent transmissions have become a popular approach in previous works to increase performance metrics like the PRR or communication range of WSNs. *Glossy* [5] presents a network flooding architecture for IEEE 802.15.4 networks which takes advantage of CI by concurrent transmissions of the same data, and has been reported to accomplish PRRs up to 99.99%.

A. Concurrent LoRa transmissions

The concept of concurrent transmissions has also been investigated for LoRa-based WSNs. For example, Liao et al. examine concurrent LoRa transmissions in [2]. They conclude that due to an unpredictable carrier frequency offset (CFO), even packets with identical contents collide on the wireless channel. The authors find that LoRa can tolerate such packet collisions from concurrent transmissions when the *capture effect* occurs, meaning that only the packet with a higher power level is being demodulated. Ultimately, the demodulation process is analogous to the situation when collisions are caused by transmissions with different data content. The demodulation of colliding LoRa packets is also a well researched topic area, and discussed in *mLoRa* [6] and *FTrack* [7]. Both works use individual features from the physical layer of collided packets to separate them and thus increase the throughput by a factor of approx. $3\times$. In [3], Eletreby et al. present *Choir*, a system to demodulate multiple parallel LoRa transmissions with different as well as identical content by using features from hardware imperfections, like time, phase and frequency offsets, of every individual transmitter. It was also discovered that packets with identical content which are transmitted by multiple nodes are received with greater power. On the one side, this leads to both increased data rates as well as a larger transmission range. Specifically, they were able to increase the reception range by a factor of $2.65\times$ when using 30 concurrently transmitting nodes. On the other side, these collisions of cause require a complex demodulation approach for correct collision resolution to retrieve the transmitted data content.

B. Alternative means of LoRa transmission

Researchers have also focused on alternative means for data transmissions beyond the limits of the standard LoRa PHY. For example, [8] investigates the effect of LoRa transmitters with low signal-to-noise ratios (SNRs). One important insight therein is that LoRa transmissions, which have an insufficient SNR for proper demodulation, can still interfere with other transmissions. The authors take advantage of this and present a *falcon transmitter* that overhears weak transmissions and selectively interferes with them in a manner that a receiver gateway is able to decode the interfered signal. In [9], Xu et al. present *Ostinato* which transforms the original data packet into a pseudo packet with repeating symbols. The use of repeating symbols concentrates the energy and increases the SNR. The authors in [10] introduce *CloakLoRa*, an Amplitude modulation (AM) scheme to add an additional covert channel to LoRa. The orthogonal AM is achieved on hardware level by switching an transistor that adds an impedance load to the antenna output, effectively changing the amplitude of the transmitted signal.

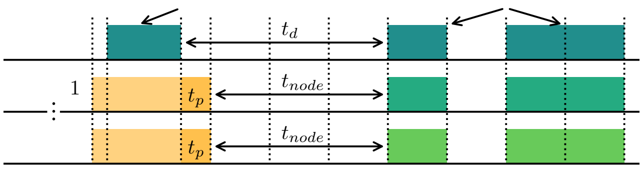


Fig. 1. Synchronization of nodes for CI transmissions.

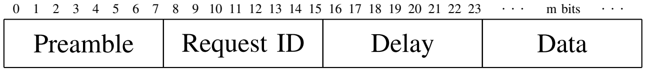


Fig. 2. Packet structure of a CI Request.

In our work, we combine two different well known modulation techniques on top of standard LoRa, that do not rely on successful demodulation of LoRa packets, but rather on their simple presence or absence. This way, our approach can take full advantage of CI periods occurring in concurrent transmissions of LoRa packets with increased amplitudes, without having to handle the complicated demodulation process of collisions. We also do not need any modification on the transmitter side and can use Commercial off-the-shelf (COTS) LoRa transmitters.

III. SYNCHRONIZATION PROTOCOL

To take advantage of Constructive Interference, all transmitting nodes need to be time-synchronized, such that packet transmissions can start simultaneously. We hence implemented a protocol that is capable of synchronizing a various number of nodes for concurrent data transmission upon a request from an initiator node. Figure 1 shows the timing diagram of our implemented procedure which uses a specific packet structure as depicted in Figure 2. The packet structure is embedded in the payload field of a regular LoRa packet, such that every standard LoRa receiver is capable of receiving it. It starts with a *Preamble* field, that identifies a CI request. This is followed by an *Request ID* that helps to identify different CI requests. The next field is the *Delay* which specifies the temporal delay before the receiving nodes shall start their transmissions. At the end comes the *Data* field containing m bits of data to be transmitted concurrently.

The algorithm works as follows: First, an initiator node prepares the CI request packet. Then, the packet is sent via standard LoRa to all nodes within reach. On reception of a CI request packet, a node waits an individual time $t_{d,node}$ which is derived from the specified time delay t_d in the packet header. This timing correction is necessary to compensate for different device processing speeds and to ensure that all subsequent transmissions start synchronously. Typically there are two types of delays, the systematic and predictable processing delays t_p, and unpredictable and random delays t_r that occur due to to physical imperfections, quantization errors, and environmental conditions. Since typically t_p is

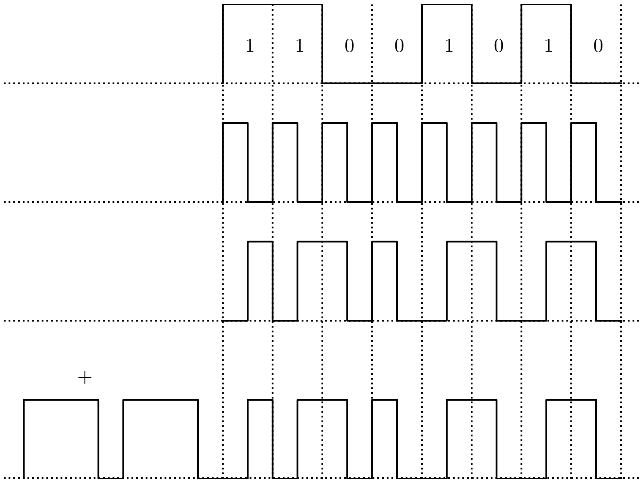


Fig. 3. Our OOK implementation uses a preamble and Manchester encoding of the data bits.

many times larger than t_r, the approximation in Equation (1) is precise enough for our use case.

$$t_{d,node} = t_d - t_p - t_r \approx t_d - t_p \tag{1}$$

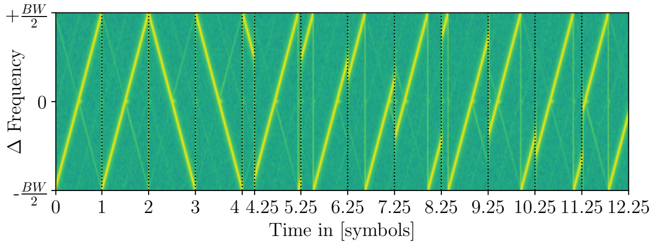
After the time delay, all nodes (including the initiator node) begin to transmit the requested data simultaneously, leading to desired interference effects, including alternating constructive and destructive interference. During the constructive periods that happen multiple times during each transmission, the resulting signal's amplitude increases.

IV. MODULATION

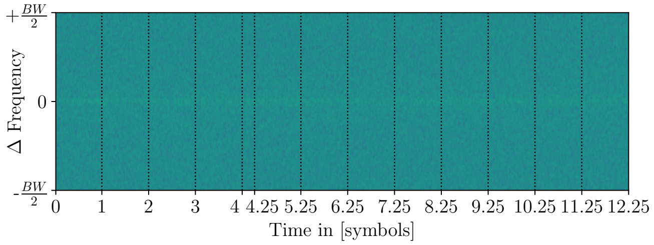
We use two well known modulation techniques on top of the CI-enhanced LoRa PHY, namely On–Off Keying and Pulse Width Modulation. The two modulations are used to take advantage of the partially increased amplitude values (due to CI) occurring in real-world concurrent LoRa transmissions. They also do not require the successful demodulation of LoRa packets for data communications, but instead modulate and demodulate data based on the presence and absence of the RF carrier waves. Since LoRa uses a chirp modulation, were refer to signals within the configured Bandwidth (BW) around the center frequency. In our implementation, we use transmissions or no transmissions of short LoRa packets for the modulation, allowing us to use COTS LoRa transmitters.

A. On–Off Keying

On–Off Keying is a simple binary modulation scheme that interprets the presence of a RF carrier wave (On) as a logical 1 and the absence (Off) as a logical 0. Our implementation is shown in Figure 3. In the first step, we use Manchester encoding to generate a self-clocking signal. Thereby, each logical 1 bit is replaced by a $[0, 1]$ bit sequence and each logical 0 bit by $[1, 0]$. This encoding scheme can be easily implemented using a clock signal and logical XOR operations. This way we can add temporal information to our data and compensate timing drifts due to long sequences of equal data bits. In addition, the end of a packet can be also reliably detected by an absence of logical bit changes. To transmit the Manchester encoded data, we use very small LoRa packets (12.25 symbols) for a logical 1. A logical 0 is encoded by an equally long time without transmission. Figure 4 shows this implementation. The packet size also determines the achievable data rates which can be derived as follows. The LoRa symbol period can be calculated based on Spreading Factor (SF) and BW as shown in Equation (2) [11].



(a) OOK representation of the binary value $(1)_2$.



(b) OOK representation of the binary value $(0)_2$.

Fig. 4. Our OOK implementation uses fixed sized LoRa Packets for transmission in on periods, like shown in (a). During off periods no packet is transmitted as shown in (b). A period length is 12.25 LoRa symbols long.

$$T_{lora,s} = \frac{2^{SF}}{BW} \tag{2}$$

Since we use Manchester encoding and LoRa packets with 12.25 symbols, Equation (3) can be used to obtain our effective bit period.

$$T_{ook,bit} = 2 * 12.25 * T_{lora,s} = 2 * 12.25 * \frac{2^{SF}}{BW} \tag{3}$$

Using this, we can calculate the data rate with Equation (4).

$$DR_{ook} = \frac{1}{T_{ook,bit}} = \frac{BW}{2 * 12.25 * 2^{SF}} \tag{4}$$

Table I shows the LoRa symbol times, the OOK bit times and data rates for different SFs. Here it should be noted, that lowering the SF or increasing the BW shortens the duration of LoRa symbols and the OOK bit periods, which leads to

TABLE I
ACHIEVABLE DATA RATE USING OUR OOK MODULATION

LoRa SF	**7**	**8**	**9**	**10**	**11**	**12**
$T_{lora,s}$ [ms]	1.024	2.048	4.096	8.192	16.384	32.768
T_{bit} [ms]	25.09	50.18	100.35	200.7	401.41	802.82
DR [bit/s]	39.86	19.93	9.96	4.98	2.49	1.25

(raw data without preamble; BW = 125 kHz; CR = 4/5)

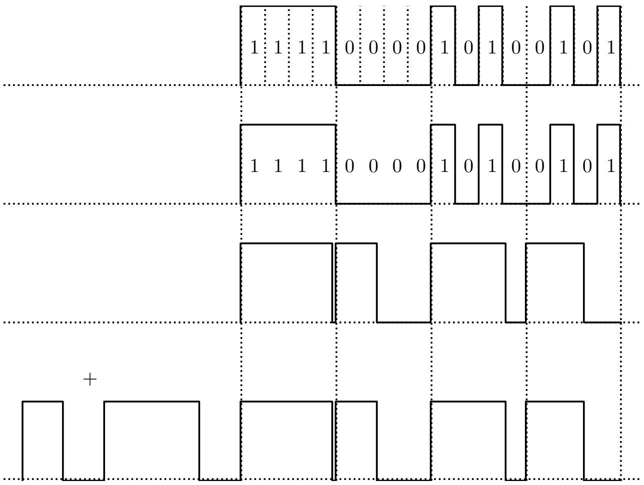


Fig. 5. Our PWM implementation uses a preamble, is based on bit grouping and has a min. pulse length > 0 and max. pulse length $<$ PWM period.

higher timing requirements in the demodulation process. To increase reliability and simplify demodulation, we decided to include a preamble prior to our data. The preamble consists of symbols that cannot be used for data encoding to simplify distinction of preamble and data. It consists of two sequences of 1.5 period long logical 1s, followed by short 0.5 period long logical 0s.

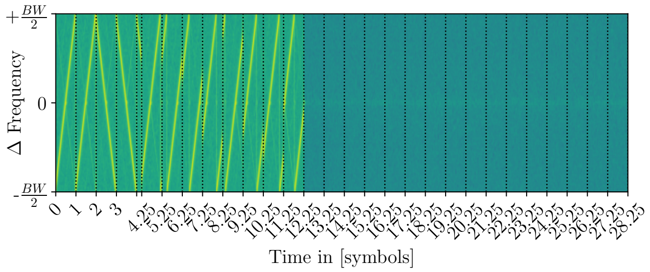
B. Pulse Width Modulation

Pulse Width Modulation is a digital modulation that modulates data based on the presence and absence duration of the RF carrier wave. This modulation scheme interprets the presence duration of the carrier wave as a logical value, proportional to the duration. The value range and thus the amount of data bits per period can be set up individually depending on the application specific needs. Our implementation of this modulation scheme is shown in Figure 5. Since PWM can modulate more than one bit in a single period, the data bits are first split into groups of n bits, resulting in a data range from 0 to $2^n - 1$. The actual PWM takes place in the next step, where the grouped data bits are transformed into a pulse width. Since LoRa Packets have a minimal length, the smallest pulse width must also be at least this size. We set the smallest pulse width, representing the value 0 as $t_{min} = 12.25\ symbols$. The remaining PWM period is divided into 2^n entities. Since LoRa offers a variable preamble length and the smallest entity in LoRa is one symbol, we decided to use the preamble length for varying the pulse width in our implementation. This way the PWM period length can be expressed as shown in Equation (5).

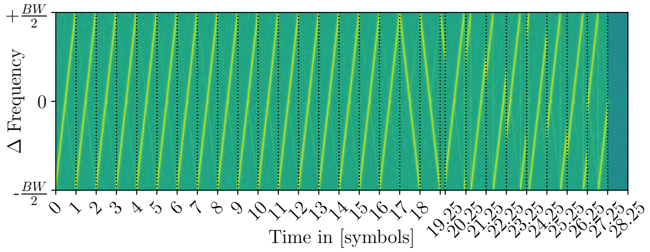
$$t_{pwm} = t_{min} + 2^n = 12.25 + 2^n\ symbols \tag{5}$$

We set the maximum pulse length to be one entity smaller than the PWM period length, Equation (6) can be used for it's calculation.

$$t_{max} = t_{pwm} - 1 = 12.25 + 2^n - 1\ symbols \tag{6}$$



(a) PWM representation of the decimal value $(0)_{10} = (0000)_2$.



(b) PWM representation of the decimal value $(15)_{10} = (1111)_2$.

Fig. 6. Our PWM implementation uses different sized LoRa Packets for generating pulses with variable length. (a) and (b) show the PWM modulated pulse lengths for different data values, grouped by 4 bits (nibbles).

Figure 6 shows our PWM implementation using bit groups of $n = 4$, where Figure 6a shows the representation of the lowest value, a decimal 0, consisting of $12.25\ symbols$ transmission time and a $16\ symbols$ duration off time. The maximum value, a decimal 15, is represented by $27.25\ symbols$ transmission time and $1\ symbol$ off time, as shown in Figure 6b. Using a minimum pulse width > 0 and a maximum width $<$ PWM period, the modulation includes temporal information and becomes self-clocking. This way it can also compensate drifts in timing occurring in long sequences of minimal and maximal data bits. This method also allows us to determine the end of a packet by detecting the absence of PWM data. The individual pulse length for a value x in groupings of n bits can be calculated using Equation (7).

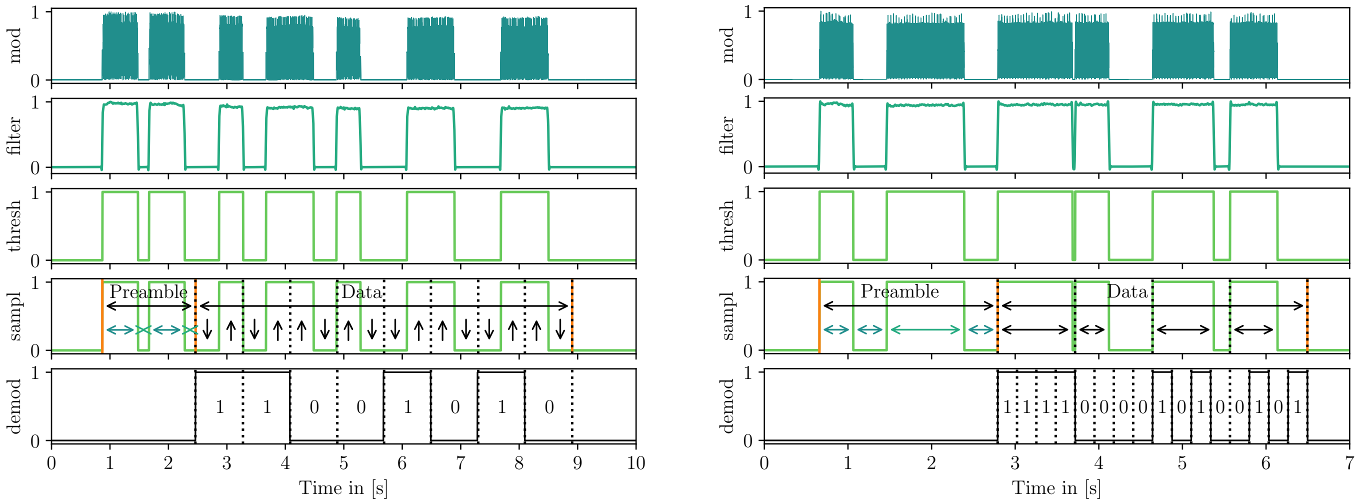
$$t_{pulse}(x) = t_{min} + \frac{x}{2^n - 1} * (t_{max} - t_{min}) \tag{7}$$

Table II shows the LoRa symbol times, the PWM bit times and data rates for different SFs. Here it should be also noted, that just like lower SFs or higher BWs, increasing the number of bits per group also leads to higher timing requirements in the demodulation process. Instead of using only a single symbol as the smallest entity, it is also possible to increase the symbol count to also increase the distinctiveness of pulse widths. Also, like in our OOK implementation, we decided

TABLE II
ACHIEVABLE DATA RATE USING OUR PWM IMPLEMENTATION

LoRa SF	**7**	**8**	**9**	**10**	**11**	**12**
$T_{lora,s}$ [ms]	1.024	2.048	4.096	8.192	16.384	32.768
T_{bit} [ms]	7.23	14.46	28.93	57.86	115.71	231.42
DR [bit/s]	138.27	69.14	34.57	17.28	8.64	4.32

(raw data without preamble; BW = 125 kHz; CR = 4/5; 4 bit grouping)



(a) OOK demodulation of the value $(11001010)_2$.

(b) PWM demodulation of the value $(1111\ 0000\ 1010\ 0101)_2$.

Fig. 7. Our signal processing pipeline, composed of a low pass filter for pre-filtering data and the use of thresholding to binarize data. Subfigure (a) shows the demodulator for our OOK implementation, while (b) shows the demodulator for our PWM implementation. The modulated data (mod) was captured from two concurrently transmitting nodes. As it can be seen from the timelines, the PWM implementation offers higher data rates compared to OOK.

to use a specific preamble in front of the actual payload that is not replicable within the data part to simplify preamble and data separation. The preamble starts with an t_{min} pulse, followed by a pause with the same length. Next, a pulse with the length of a whole PWM period t_{pwm} is transmitted, again followed by a t_{min} pause.

V. EVALUATION

For the evaluation we used two COTS Arduino Boards with Dragino LoRa Shields based on the SX1276 RF Chip from Semtech [12] on the transmitter side. Since we want to use the modulation schemes for emergency scenarios only and thus focus on reliable transmissions rather than high data rates, we decided to configure a SF of 12, a BW of 125 kHz and Error correction rate (CR) = $4/5$. We used our synchronization algorithm presented in Section III to synchronize the two LoRa transmitters and enable concurrent transmissions. On the receiver side, we used an USRP B205mini-i Software defined radio (SDR) from Ettus Research [13]. The signal processing, analysis and demodulation is done in software using the programming language Python. The concept for our signal processing pipeline and the demodulator implementations for OOK and PWM are shown in Figure 7 and work as follows: First, to enable data reception, we configured our SDR with a center frequency of 868 Mhz and a bandwidth of 1 Mhz, this left enough room for all the possible bandwidth configurations of the LoRa transmitters. The SDR uses IQ sampling and thus provides the complex samples in the format $c = a + i * b$. Since the used modulations are based on the presence or absence of data, we only consider the signal magnitude for further analysis. As shown in Equation (8), the magnitude can be easily retrieved from the real and imaginary part of the complex sample values.

$$|c| = |a + i * b| = \sqrt{a^2 + b^2} \tag{8}$$

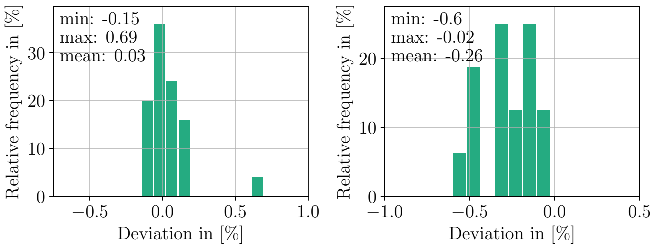
After calculating the magnitude values, we apply a low pass filter and decide based on a threshold value, whether a value is interpreted as a logical high or low value. In the next step, we sample the binary data and demodulate it according to the corresponding modulation scheme.

A. *On–Off Keying*

In the OOK modulation scheme we first search for the preamble. Therefore we compare two subsequent logical high periods for equality in length and check whether there are logical low values after both, that are only one third in duration. After finding a preamble we start to demodulate the subsequent data. Since we already have the timing of the preamble, we can average the lengths and determine the period length for sampling. We sample two times per period to capture the level changes originating from the Manchester encoding. Transitions from low to high level are interpreted as logical 1s and changes from high to low as logical 0s. The end of a transmission is detected by the absence of logical level changes. In our evaluation we analyzed the timing deviation of our demodulator implementation. Therefore, we measured the timing errors as the difference between the expected and real duration of On and Off periods in relation to the smallest bit entity, namely T_{bit} for OOK demodulation. Since, the bit period in our OOK implementation is comparatively large, we were able to achieve results of min/max and mean timing deviations well below 1 %. The complete results are shown in Figure 8a.

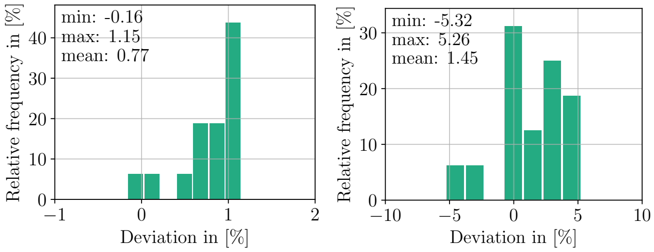
B. *Pulse Width Modulation*

In the PWM scheme we also start to search for a preamble. Therefore we compare two subsequent logical high periods. Both periods must be followed by logical low periods with the same length as the first logical high period. The length relation



(a) OOK Demodulation.

(b) PWM (2-bit grouping).



(c) PWM (3-bit grouping).

(d) PWM (4-bit grouping).

Fig. 8. Demodulation timing error for SF = 12; BW = 125 kHz; CR = 4/5

between both high periods is determined by the number of bits per group and the smallest symbol entity. For example, $\frac{12.25}{28.25}$ when using 4 bit grouping and one symbol as the smallest entity, since the shortest PWM pulse is 12.25 symbols long, and the PWM period is $12.25 + 2^4 = 12.25 + 16 = 28.25$ symbols long. After finding a preamble we start to demodulate the subsequent data. Since we already have the timing of the preamble, containing the minimum PWM pulse length and PWM period length, we have the necessary reference lengths for demodulation. In the next step, we measure the individual pulse lengths t_{pulse}. By comparing these pulse lengths with the minimum and maximum pulse length t_{min} and t_{max}, we can determine the demodulated value using Equation (9).

$$demod(t_{pulse}) = \frac{t_{pulse} - t_{min}}{t_{max} - t_{min}} * (2^n - 1) \tag{9}$$

The end of a transmission is detected by the absence of PWM data. We also analyzed the timing errors for our PWM implementation. The results are shown in subfigures 8 (b)–(d) for different bit groupings of 2, 3 and 4 bits. The results show that the timing errors depend on the used bit grouping. When using groups of 2 bits, the min/max and mean timing errors are well below 1 %. These values increase with higher bit group count, rising on average from −0.26 % using 2 bits up to 1.45 % using 4 bits. This is expected, because more bits per group decrease the duration of the smallest entity in PWM, increasing the timing requirements and errors.

Nevertheless, both presented modulation schemes offer alternative methods of data transmissions that come with advantages (e.g., higher resilience) and disadvantages (e.g., lower data rate) that need to be taken into account when selecting one of them.

VI. CONCLUSION AND OUTLOOK

In this paper we have researched the combination of two well known modulation schemes, namely OOK and PWM on top of the LoRa PHY to take advantage of the signal amplitude increases that occur as a result of concurrent transmissions. The idea behind our approach is to shift the focus from trying to demodulate complicated LoRa collisions to only relying on the presence or absence of the RF carrier wave. This simplifies the demodulation step while taking advantage of the stronger signal due to the use of CI. Even though our approach only provides a very low data rate, in the order of several bits per second, it may prove helpful in emergency scenarios, where no reliable communication is possible via standard LoRa, yet multiple nodes can synchronize and send data cooperatively. In the future, we plan to validate our approach in a real world environment and with a larger number of LoRa nodes.

ACKNOWLEDGMENT

This work was supported by the German Federal Ministry of Education and Research (BMBF) within the scope of the project EXDIMUM.

REFERENCES

[1] D. Szafranski, M. Bouchur, and A. Reinhardt, "Investigating Constructive Interference for LoRa-based WSNs," in *Proceedings of the 19th GI/ITG KuVS Fachgespräch "Sensornetze" (FGSN)*, 2022, pp. 25–28.

[2] C.-H. Liao, G. Zhu, D. Kuwabara, M. Suzuki, and H. Morikawa, "Multi-Hop LoRa Networks Enabled by Concurrent Transmission," *IEEE Access*, vol. 5, pp. 21 430–21 446, 2017.

[3] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering Low-Power Wide Area Networks in Urban Settings," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*, 2017, pp. 309–321.

[4] P. Tian, X. Ma, C. A. Boano, Y. Liu, F. Yang, X. Tian, D. Li, and J. Wei, "ChirpBox: An Infrastructure-Less LoRa Testbed," in *Proceedings of the International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2021, pp. 115–126.

[5] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient Network Flooding and Time Synchronization with Glossy," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2011, pp. 73–84.

[6] X. Wang, L. Kong, L. He, and G. Chen, "mLoRa: A Multi-Packet Reception Protocol in LoRa networks," in *Proceedings of the 27th IEEE International Conference on Network Protocols (ICNP)*, 2019, pp. 1–11.

[7] X. Xia, Y. Zheng, and T. Gu, "FTrack: Parallel Decoding for LoRa Transmissions," *IEEE/ACM Transactions on Networking*, vol. 28, no. 6, pp. 2573–2586, 2020.

[8] S. Tong, Z. Shen, Y. Liu, and J. Wang, "Combating Link Dynamics for Reliable Lora Connection in Urban Settings," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2021, pp. 642–655.

[9] Z. Xu, P. Xie, J. Wang, and Y. Liu, "Ostinato: Combating LoRa Weak Links in Real Deployments," in *Proceedings of the 30th IEEE International Conference on Network Protocols (ICNP)*, 2022, pp. 1–11.

[10] N. Hou and Y. Zheng, "CloakLoRa: A Covert Channel over LoRa PHY," in *Proceedings of the 28th IEEE International Conference on Network Protocols (ICNP)*, 2020, pp. 1–11.

[11] Semtech Corporation, *AN1200.22 - LoRa Modulation Basics*, 2nd ed., 2015.

[12] Semtech Corporation, *SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver*, 4th ed., 2015.

[13] Ettus Research, *USRP B200mini Series - Product Overview, Features*, 2022.