Received 7 February 2023, accepted 17 February 2023, date of publication 3 March 2023, date of current version 20 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3252170

RESEARCH ARTICLE

Classification of LoRa Signals With Real-Time Validation Using the Xilinx Radio Frequency System-on-Chip

COLIN HORNE, (Member, IEEE), NIAL J. PETERS, (Member, IEEE), AND MATTHEW A. RITCHIE, (Senior Member, IEEE)

Department of Electronic and Electrical Engineering, University College London, WC1E 6BT London, U.K.

Corresponding author: Colin Horne (uceehor@ucl.ac.uk)

This work was supported in part by the PETRAS National Centre of Excellence for IoT Systems Cybersecurity funded by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/S035362/1, and in part by the Defence and Security Accelerator (DASA) through funding ARESTOR System Development.

ABSTRACT This paper demonstrates a real-time LoRa Internet of Things (IoT) signal classification technique that runs on Xilinx Radio Frequency System-on-Chip (RFSoC) hardware. IoT signals are being used for wider arrays of applications and therefore awareness of their presence is important for cyber security and infrastructure protection as well as battlefield situational awareness. Within this research a dataset of LoRa waveforms is captured using the RFSoC which bounds the possible combinations of waveform parameters. Offline algorithms are tested against this data to evaluate how to extract the centre frequency, bandwidth and spreading factor. The algorithms are then adapted to run natively on the Xilinx RFSoC to enable real-time classification of waveforms from non-cooperative LoRa transmitters with a high degree of classification success.

INDEX TERMS IoT, electronic surveillance, Hough transform, digital signal processing, RFSoC.

I. INTRODUCTION

Internet of Things (IoT) devices are ubiquitous, with the number and diversity of devices ever growing. Typically they require wireless connectivity to operate resulting in many radio-frequency (RF) signal transmissions within an IoT-equipped environment. They are generally used in low data-rate and low power-consumption roles. Hence they transmit signals in-frequently, with long duration transmits with the aim of being robust to long ranges and interference. Various types of IoT signals exist, these include Zigbee [1], SigFox [2], LoRa [3] and Bluetooth LE [4] to name a few.

Awareness of IoT signals within a given area of interest and in particular within your own infrastructure is a key security related capability. To understand the significance of the IoT signals which exist in the environment we must first monitor it with a view to classifying what is found. This paper describes the development of a signal processing chain which can successfully identify the characteristics of over-the-air captured LoRa signals in a PC based software environment. We further describe the results of applying a modified, real-time capable, version of the processing chain using the multi-role RF sensor platform, ARESTOR [5]. Specifically we monitor the environment for LoRa signals [6], and then classify them in terms of their signal characteristics such as centre frequency, bandwidth and spreading factor. This research is part of a portfolio of projects connected to IoT Cyber Security in a program titled the PETRAS hub. There are three key strands to the work presented here. In the first strand, ARESTOR was used to capture LoRa signals with varying characteristics from a cooperative transmitter in order to create a database of test signals. This database was used to inform the design and test of a software algorithm which could detect and classify the signals in terms of their key parameters. Finally, the software algorithm was adapted to run on ARESTOR, making use of hardware acceleration to achieve real-time operation.

The majority of prior research in the this area has focused on fingerprinting particular hardware transmitters, a process called Radio Frequency Fingerprint Identification (RFFI),

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaolong Li.

that are producing LoRa signals. For example, [7] used supervised machine learning and zero-shot learning to identify LoRa devices. They were able to demonstrate very high success rates in distinguishing devices with different chipsets, and moderate success on devices with the same chipsets. Another paper, [8], uses the carrier frequency offset of specific LoRa devices to perform RFFI and suggests its use as an additional security layer in LoRaWAN systems. In that article the signals from six separate LoRa transmitters were captured and high classification success rates were achieved at close ranges and high SNRs.

In [9] three deep-learning methods for LoRa RFFI were compared using an experimental dataset collected using 25 different LoRa devices. A CNN based approach was found to have the best classification success. A larger number of LoRa devices (60) were used in a different RFFI study [10], which again showed good performance in identifying rogue devices using a k-nearest neighbours approach. The robustness of deep-learning RFFI techniques for LoRa devices against changes in the device deployment settings are discussed in [11]. The findings showed that classification accuracy was severely impacted by changes in LoRa configuration (e.g. spreading factor) and also by changes to the channel.

Very little can be found in the literature with similar objectives to ours, that is, the detection and characterisation of LoRa signals. This objective was investigated for WiFi signals in [12] using a down-scaled version of the Faster R-CNN framework. However, the most comparable publications to our work can be found in [13] and [14]. The study in [13] predominantly looked to identify LoRa signals using the YOLO neural network processing architecture. A similar piece of work, [15], also uses YOLO but extends the study to a wider range of modulations. Both [13] and [15] focused on processing pre-recorded and simulated data offline to classify the signal parameters. The work in [13] also tested a single over-the-air example using a Lora bandwidth of 125 kHz and spreading factor of 9 over a range of transmitter-receiver separations. The work in [14] used over-the-air captured signals and employed compressive sampling to reduce the sample rate, achieving 100% successful LoRa parameter identifcation down to a level of 0 dB SNR.

All the RFFI related works discussed, except [11], use a known, fixed LoRa configuration (centre frequency, spreading factor etc.) and none address the issue of localising signals in the time-frequency domain. The works presented which address the identification of LoRa signal characteristics mainly use off-line machine learning approaches to identifying LoRa transmission parameters.

The unique contributions of this article are an algorithmic approach to the LoRa signal identification problem and modifications to the algorithms allowing a straightforward hardware implementation which provides real-time characterisation of LoRa waveforms on an edge device. By creating such an implementation the concepts have been increased in Technology-Readiness-Level (TRL) compared to prior literature. This work represents the first publication of using an RFSoC system in this role, one to which it is clearly well suited. Instead of synchronised co-operative captures of RF signals of interest as used in previous studies (e.g. [13] and [15]) our hardware implementation is able to continuously monitor the RF spectrum and only store instances when events of interest occur. This data reduction is vital when monitoring wide bandwidths of the RF spectrum.

The paper is structured in the following way, Section II shows the background theory and structure of LoRa signals. Section III then describes the implementations of signal classification algorithms and the database of signatures that has been created. Section IV discusses the hardware used to transmit and capture the LoRA IoT signals. Section V details how the algorithms were then implemented on ARESTOR, facilitating real-time detection and characterisation of LoRa signals. Finally Section VI concludes by discussing the key results that have been shown within this article and comments on future directions of the work.

II. LoRa BACKGROUND THEORY

LoRa is a proprietary physical-layer standard, however, many of its details have been reverse-engineered and published in the open literature [16], [17], and open-source software implementations are available [18]. A LoRa signal consists purely of rising and falling linear frequency modulation (LFM) waveforms, or chirps. For a specific LoRa configuration the LFM signal covers a constant range of frequencies which it traverses at a constant rate.

The data payload portion of the signal package is contained in a number of rising frequency LFM segments, the information payload is encoded in the frequency offset of the chirp as referenced to the base frequency of the chirp. When the chirp reaches its maximum frequency for the configuration employed, it wraps to the base chirp frequency, and continues rising for the remainder of the chirp period.

The start of a LoRa message consists of a preamble containing a number, from zero upwards, of rising frequency chirps, the actual number being a user configurable value. These rising LFM chirps are followed by two additional rising LFM chirps, and two and a quarter falling LFM chirps which signify the start of the signal payload.

The spectrogram of an example LoRa signal transmitted by a FiPy module, see Section IV-B, and captured using ARESTOR, Section IV-A, is shown in Figure 1. The spectrogram is annotated to highlight the specific regions in the preamble of the LoRa signal and the payload. Specifically in this example there are 8 additional up-chirps prior to the 2 mandatory up-chirps, followed by 2.25 down-chirps, and the message payload.

Decoding of the LoRa transmission may be achieved by synchronised multiplication of the received symbols with the complex conjugate of the base up-chirp signal. This process produces specific ‘beat’ frequencies for each symbol value which might be identified by frequency domain analysis.

The physical layer transmission characteristics of the LoRa signal as defined for European use which are of interest in

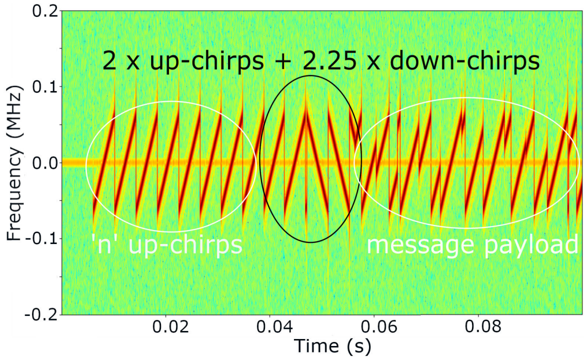


FIGURE 1. LoRa chirp signal preamble and payload regions.

TABLE 1. LoRa signal definition parameters.

Parameter	Notation	Range
Bandwidth	B	125 MHz, 250 MHz
Spreading Factor	SF	7, 8, 9, 10, 11, 12
Carrier Frequency	f_c	863 MHz ... 870 - B MHz

this work are described by the parameters shown in Table 1. Other parameters of the LoRa protocol such as the coding rate change how the captured data are encoded and interpreted by the receiver but do not change the transmission signal itself.

Rigorous mathematical descriptions of the LoRa modulation can be found in [19] and [20]. A basic representation of the LoRa payload chirp signal is given below.

The time extent T_c in seconds of each chirp signal is

$$T_c = \frac{2^{SF}}{B} \tag{1}$$

For a single conventional up-chirp signal with frequency transitioning linearly between 0 Hz and B Hz and modulated on to a carrier the output signal can be represented by:

$$c(t) = \begin{cases} e^{j\theta(t)}, & 0 \leqslant t \leqslant T_c \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\theta(t)$ is the accumulated phase of the modulated chirp signal, defined as

$$\theta(t) = 2\pi f_c t + \pi \frac{B}{T_c} t^2, \quad 0 \leqslant t \leqslant T_c \tag{3}$$

for a carrier frequency f_c.

This result can be generalized for the alternative frequency starting points of the LoRa chirp which represent the various symbols encoded on to the chirp signal, the transmitted signal $s(t)$ becomes:

$$s(t) = \begin{cases} e^{j\phi(t)}, & 0 \leqslant t \leqslant T_c \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\phi(t)$ is the accumulated phase of the coded chirp signal, defined as

$$\phi(t) = \begin{cases} 2\pi f_c t + \pi \frac{B}{T_c} t^2 + 2\pi f_n t, & 0 \leqslant t < T_{corr} \\ 2\pi f_c t + \pi \frac{B}{T_c} t^2 + 2\pi f_n t - 2\pi B t, & T_{corr} \leqslant t < T_c \end{cases} \tag{5}$$

where f_n is the frequency offset for symbol number $n \in \{0..M-1\}$ where M is the number of possible symbols. f_n is calculated as:

$$f_n = \frac{n}{M} B \tag{6}$$

and T_{corr} is the time at which the chirp frequency wraps back to zero offset from the carrier upon reaching the maximum chirp frequency:

$$T_{corr} = T_c\left(1 - \frac{n}{M}\right) \tag{7}$$

Figure 2 shows spectrograms of a captured LoRa signal, the mixing signal which might be employed to decode it, and the results of mixing the captured signal with the decoding signal. The required decoding signal is known *a priori* and is synchronised with the captured signal. Where the captured signal corresponds to the base up-chirp signal the mixing process produces a zero frequency output. For cases where the chirps are offset from the base chirp then two constant frequencies are output, one for the region before the data chirp folds back in frequency and one after, which provides an estimate of the frequency offset between the captured chirp signal and the reference mixing signal, and therefore an indication of the payload data encoded into the chirp.

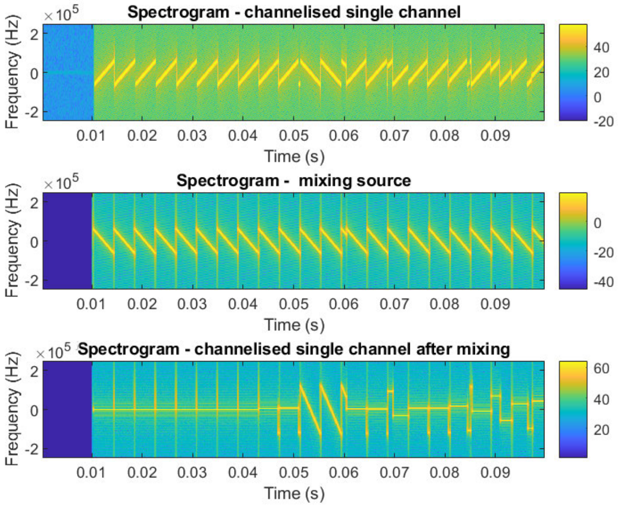


FIGURE 2. Spectrograms of captured LoRa signal (upper), mixing signal (middle) and mixing results (lower).

In this work we do not carry out the decoding process on captured data to extract the payload information, instead we concentrate on the physical layer, characterising the parameters associated with each identified LoRa signal. To allow data decoding conventional LoRa receivers need knowledge of the LoRa parameters being used in the transmission. Our solution is agnostic to the transmission parameters so provides the ability to operate across any encoded LoRa transmission without prior knowledge. In addition alternative IoT protocols could also be sensed and potentially characterised by the system given suitable algorithm development.

III. ALGORITHM DEVELOPMENT

A. LoRa SIGNAL DATABASE

The first stage of algorithm development involved using the ARESTOR platform as a receiver for capturing a wide selection of example raw LoRa signals generated by the FiPy IoT Transceiver, see Section IV for details of the hardware setup. This process allowed the creation of a labelled database of over-the-air captured LoRa signals from across the range of user selectable LoRa parameters. This database was subsequently used to inform the design of the processing chain, and testing of the algorithms against real LoRa signals.

Scripts running on ARESTOR and the FiPy configure each platform such that both systems are synchronised regarding the parameters of the signals being transmitted or received allowing a labelled database of known signals to be created. ARESTOR captures and saves the digitised signal, after mixing down to baseband, to a file named to specify the LoRa parameters employed to create the signal.

To capture known over-the-air LoRa IoT signals and to label the data appropriately it is also necessary to synchronise in time the ARESTOR platform capturing the over-the-air transmission and the FiPy device which is transmitting the signal. This is achieved by means of a digital signal generated by the ARESTOR platform on a general purpose input/output pin a short time after ARESTOR is enabled to capture data from its ADCs. This signal is fed to an interrupt input on the FiPy and triggers the transmission of the appropriate LoRa signal. This technique guarantees the known signal appears within the relatively short capture period setup in ARESTOR specifically for the purpose of the database creation. The joint operation of ARESTOR and the FiPy module for labelled database creation is termed co-operative mode. When the algorithm is operating in the real-time context, see Section V, this additional synchronisation between the FiPy and ARESTOR is no longer necessary, the two components running totally independently.

The parameters of the LoRa signal examples captured by ARESTOR are signal bandwidth, spreading factor, coding rate, output power level and transmit centre frequency. An additional signal attenuation option was also included which allowed the transmitted signal to be externally attenuated by 0 dB or by an additional 10 dB. The range of options for the specified parameters are shown in Table 2. For this work we concentrate on the 863 MHz - 870 MHz frequency band only. The resulting database is comprised of 2160 example LoRa signals for the offline algorithm to be tested against. This includes measurements of two separate FiPy devices.

TABLE 2. LoRa signal database parameters.

Parameter	Range
Bandwidth	125 MHz, 250 MHz
Spreading Factor	7, 8, 9, 10, 11, 12
Coding Rate	4_5
Output Power	2 dBm, 8 dBm, 14 dBm
Centre Frequency	15 spot frequencies in 863 MHz – 870 MHz band (500 kHz spacing)
Additional Attenuation	0, 10 dB (external to FiPy)
RF Sample Frequency	4.08 GHz
Decimation	512x
Baseband Sample Frequency	7.968750 MHz

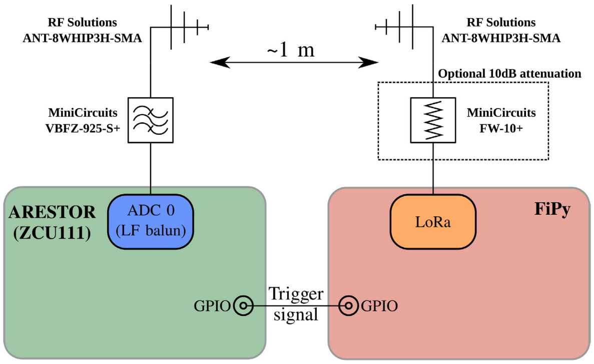


FIGURE 3. Illustration of ARESTOR platform and FiPy device layout.

An illustration of the co-operative mode configuration of ARESTOR and the FiPy for the capture of the database signals is shown in Figure 3. This shows the FiPy transmitter with optional external attenuators feeding a whip antenna. The ARESTOR system is shown on the left receiving via a whip antenna and a bandpass filter.

B. IDENTIFYING LoRa SIGNAL CHARACTERISTICS

The identification of LoRa signal parameters is based on detection of the two full bandwidth down-chirps, as shown in Figure 1, which are always part of the LoRa protocol preamble. This region of the signal is selected as the basis for LoRa signal identification as it represents the only fully defined pattern which is always present within the protocol. The initial up-chirps in the protocol preamble, which at first sight appear to be ideal candidates for identification of the signal, are not guaranteed to be included in the transmission, and are therefore not a suitable identifier. The two remaining up-chirps in the header are also not ideal for identification purposes as they carry configuration information encoded by the starting frequency in the same way as the payload, so do not begin and end at the extremes of the message bandwidth.

The processing flow for the detection and characterisation of LoRa signals is illustrated in the block diagram shown in Figure 4. The processing steps shown contain both the blocks required for the off-line version of the identification process and additions required specifically for the real-time implementation. The multiple processing chains are not applied in the offline solution but are implemented within the real-time processing shown in Section V.

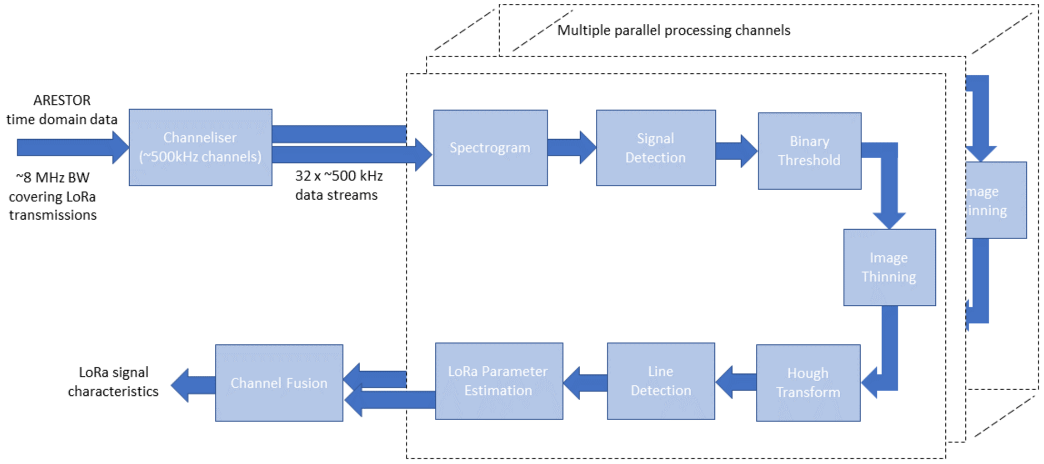


FIGURE 4. Block diagram of processing chain for LoRa protocol characteristics identification.

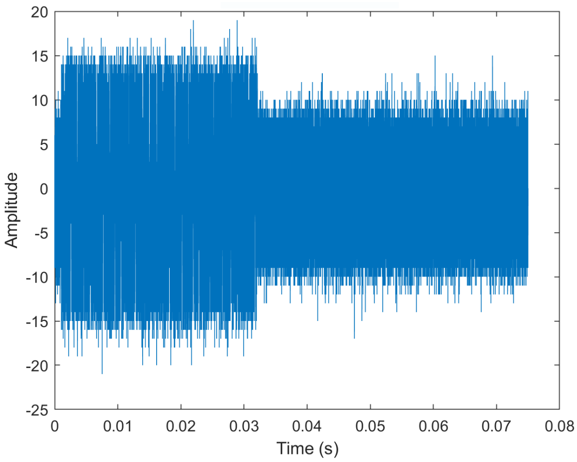


FIGURE 5. Time domain LoRa signal with 5 dB SNR.

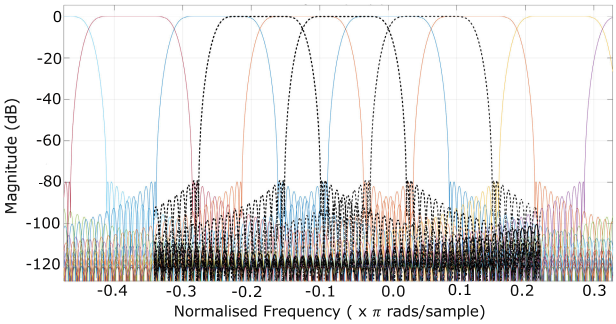


FIGURE 6. Response of overlapping channelising filters.

As previously stated we are are working with LoRa transmissions in the 863 MHz – 870 MHz band. The LoRa transmission can reside anywhere within this licence exempt frequency range. We are not considering out of band use of the LoRa protocol.

The input to the signal characterisation processing is time domain data pre-decimated to reduce the data rate and bandwidth to cover the region of interest. An example of time domain LoRa data captured with ARESTOR is shown in Figure 5 for a signal with approximately 5 dB of signal to noise ratio (SNR). Specifically, the sample rate into the channeliser is 7.96875 MHz.

We employ a channelising filter to divide the ∼8 MHz bandwidth into 32 overlapping bands of approximately 500 kHz each. The bands overlap by 50% ensuring a LoRa signal fits fully within at least one channel making the signal characterisation more straightforward, at the expense of increasing the number of processing channels, and the signal potentially being detected across multiple channels, albeit with reduced bandwidth. Figure 6 shows part of the normalised frequency response of the channelising filter with the non-overlapping channels shown by solid colour lines, and three overlapping channels shown in black dashed lines. The remaining overlapping channels are not shown for clarity

In addition to providing individual channels of 500 kHz each, the channeliser also decimates the data providing a final rate of ∼500 k samples per second of complex data to the spectrogram block. Subsequent processing is applied separately to each channel.

In the offline processing implementation only a single channel from the channeliser is processed further as there is *a priori* knowledge of the frequency of the signal of interest being used for testing purposes. In contrast the real-time implementation processes all channeliser outputs to characterize waveforms across the whole digitized bandwidth of interest.

Following channelisation the time domain signal is converted to a time-frequency representation (spectrogram) using the short-time Fourier Transform (STFT) with a Hamming window applied. Other time-frequency transforms have been shown to be superior for parameter estimation of LFM waveforms [21], however, the STFT can be readily implemented in hardware and so was a suitable choice for this work. The overlap between spectrogram samples, the zero

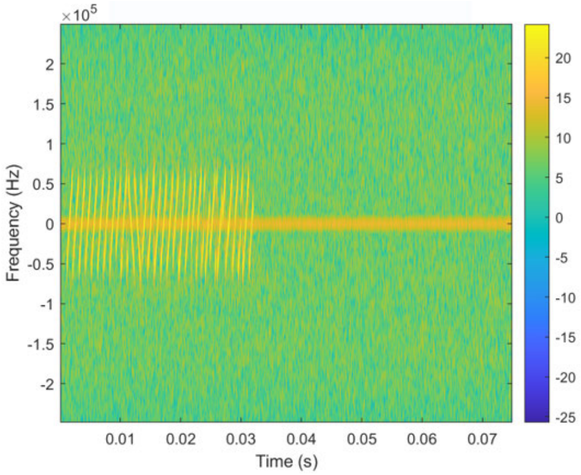


FIGURE 7. Spectrogram image of LoRa transmission captured over the air.

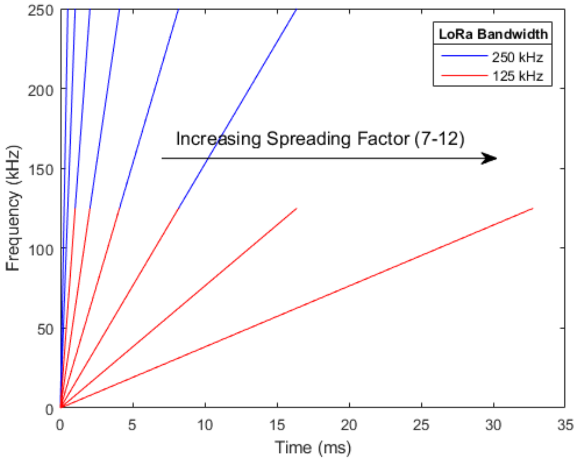


FIGURE 8. Illustration of symbol time-frequency extent for each spreading factor.

padding and spectrogram size were determined empirically. An overlap of 90% was selected with a sample set of 64 samples from the input data, zero padded to 256 samples as input to the spectrogram. This process creates one set of frequency samples for each 6 input time samples.

A spectrogram image from a single processing channel containing a LoRa signal centred on the channel centre frequency can be seen in Figure 7. The chirp nature of the transmission is clearly visible. The RF signal represented in Figure 7 and all subsequent frequency based plots are down converted to baseband.

The primary criterion for selecting the time extent of the processes following the spectrogram is ensuring the longest chirp expected to be seen within the LoRa signal is contained in a single spectrogram image allowing the full extent of the chirp in time to be determined. The longest duration LoRa chirp will be created when the transmitter is configured with a bandwidth setting of 125 kHz and a spreading factor of 12. Other LoRa parameters such as coding rate have no effect on the chirp time extent. These parameters produce a maximum chirp time of ∼33 ms. The time extent of spectrogram images provided to subsequent processing stages is fixed at 75 ms.

Figure 8 illustrates the time-frequency extent of symbols for each available spreading factor. The six blue plots represent the 250 kHz bandwidth coded symbols and the red plots the 125 kHz bandwidth. It can be seen that there is a wide variation in the symbol time duration from ∼0.5 ms up to ∼33 ms.

The spectrogram image is fed to the Signal Detection block whose function is to detect the presence of a signal of interest, enabling further processing steps The signal detection process is not required in the off-line version.

Spectrogram images containing signals of interest are subjected to a median filter to reduce the noise content, and a binary threshold using the Matlab 'ibinarize' function employing the default Otsu's method [22] to produce a black and white image as shown in Figure 9.

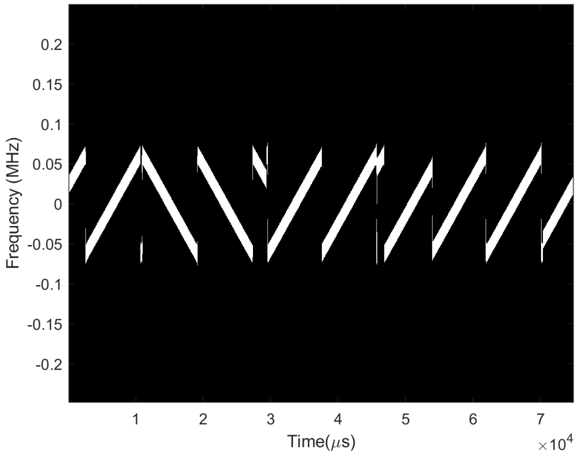


FIGURE 9. Spectrogram image of LoRa transmission after binary threshold application.

A thinning process is applied to the black and white image to reduce the detection of erroneous lines within each downchirp. This process iteratively removes threshold crossings at black-white boundaries using the Matlab 'bwmorth' function with the 'thin' parameter set, resulting in narrow lines remaining in the image. Figure 10 shows the results of the thinning process on the binary thresholded image.

The primary technique used in the LoRa signal characterisation process is the Hough transform [23], [24]. This is used to identify the lines representing the down-chirps in the thinned monochrome spectrogram image. Use of the Hough transform is commonplace in electronic surveillance, especially in the radar community where the waveforms of interest are often LFMs [25], [26], [27], [28], [29] which are represented by straight lines in the spectrogram image space. Conventionally the Hough transform attempts to identify lines within an image at all possible gradients. This is not necessary in the case of LoRa chirp detection as the lines in the image representing true LoRa LFM signatures can only occupy a small number of gradients due to the limited

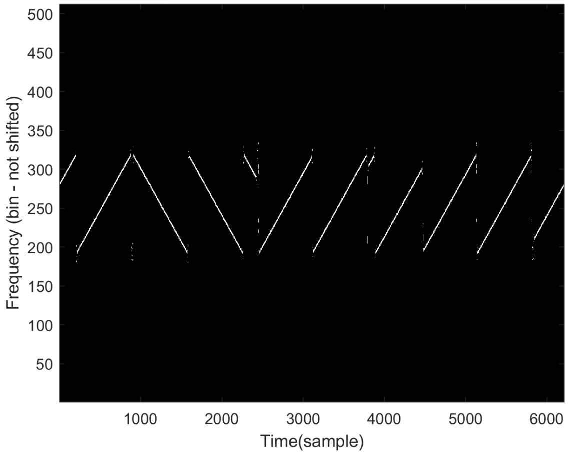


FIGURE 10. Spectrogram image of LoRa transmission after thinning.

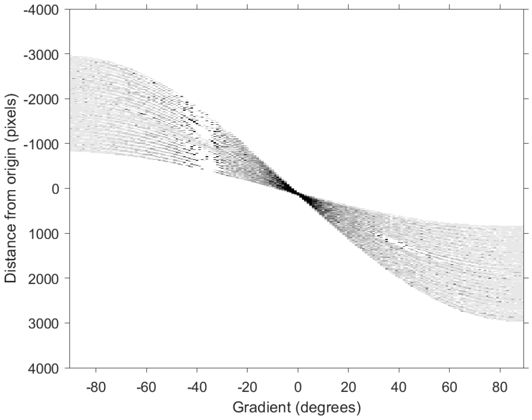


FIGURE 11. Hough space example for near vertical line in the image space illustrating intersecting sine curves.

combinations of spreading factors and bandwidths available. Only eight gradient angles across the two bandwidths need to be assessed, significantly reducing the processing requirements for the Hough transform. The actual gradients to be considered depends on the sampling rate, spectrogram sample size, sample overlap and zero padding used in the spectrogram formulation. The gradients applicable to the parameters chosen in this study can be seen in Table 3.

The Hough transform maps pixels in the input image into the 'Hough space', where they are represented by sine curves. Where multiple pixels lie on a straight line, their sine curve representations intersect in Hough space. Given sufficient intersecting curves, the intersection point can be determined as representing a line in the image. See [23] and [24] for further details. An example of the result of mapping image space to Hough space is shown in Figure 11.

An intersection identified in Hough space may be transformed back to image space to identify the line it represents. At this point there is no indication of the length of the line, and therefore no indication of the time extent or frequency extent of the signal. The final stage of the line detection involves matching pixels containing threshold crossings in the image space with the definition of the lines indicated from the Hough transform to isolate the line length. This process is carried out by the Matlab function 'houghlines'.

Due to the detection process there might be gaps within the pixel threshold crossings for the line under investigation which might lead to multiple short lines being identified. To overcome this issue we define the maximum size of any threshold crossing gaps, in pixels, which are still accepted as a continuous line, allowing gaps up to a certain length to be bridged. There is also the possibility that noise induced threshold crossings may be bridged to form spurious lines. To alleviate this problem a line length threshold, in pixels, is defined below which line segments are not considered to be valid. These two values are the 'FillGap' and 'MinLength'parameters of the 'houghlines' function.

On investigation it was found that both of these parameters require to be set according to the gradient of the line being processed due to the density of threshold crossings being dependent on the line gradient. The thresholds are found empirically. The values used for the two parameters are summarised in Table 3.

TABLE 3. Summary of 'houghlines' function FillGap and MinLength parameters for each gradient angle. (Gradient angles in degrees, parameter values in pixels).

Gradient	71.7	56.7	37.5	20.8	10.7	5.4	2.7	1.35
FillGap	30	30	30	30	30	40	50	100
MinLength	75	80	85	90	95	100	105	110

Knowledge of the location of the line in the image allows the centre frequency and bandwidth of the associated chirp signal to be estimated from the vertical (frequency) spacing of the line end points. The LoRa spreading factor can be estimated from the calculated bandwidth and the horizontal extent (time) of the identified lines.

Figure 12 shows successful identification of the LoRa signal by overlaying the identified chirp parameters for two full down-chirps over the original thinned image.

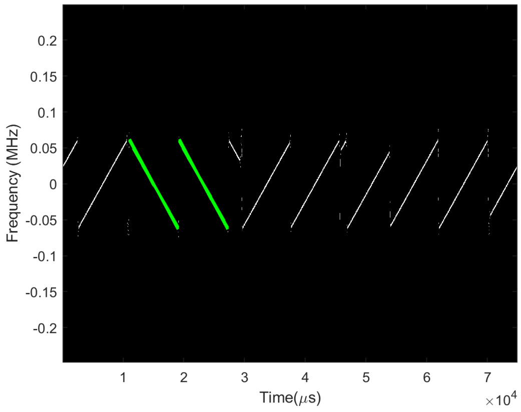


FIGURE 12. Characterisation parameters overlaid on source image.

Additional cross-channel processing is necessary in the real-time implementation to confirm the actual frequency characteristics of the signal where the signal might be seen in adjacent channels. This is represented by the 'Channel Fusion' processing block seen in Figure 4.

C. ALGORITHM PERFORMANCE

The processing chain was evaluated against the complete database of captured signals. Of the 2160 captures processed 7 were found to contain corrupted data so were disregarded. A success rate of >99.9% in the identification of both bandwidth and spreading factor was obtained for the remaining captures. On further investigation the only error was found to be a single example of identifying only one of the two downchips expected in the LoRa signal for the case of 125 kHz bandwidth and spreading factor 7. The reason for this error requires further investigation. The results of the performance evaluation are summarised in Table 4.

TABLE 4. Summary of software algorithm performance against captured database.

Parameter	Value
Captures per Tx Device	1080
Number of Devices	2
Total Captures	2060
Total Valid Captures	2053
Downchirps Correctly Identified	4105
Success Rate	>99.9%
Mean Centre Frequency Error	382 Hz
Standard Deviation of Centre Frequency Error	1028 Hz

A further quantitative evaluation of the LoRa characterisation algorithm performance was performed to assess the robustness against variations in the signal to noise ratio (SNR). The SNR experiments are somewhat analogous to separating the transmitter and receiver by increasing distances, but without effects such as multi-path interference and obstructions between the systems. Separation experiments were not possible during this project. The time domain captured data is degraded by subjecting it to increasing degrees of additive white Gaussian noise before being processed through the characterisation processing chain. The performance is evaluated by this method at SNR intervals of 3 dB. As expected the results show varying performance with LoRa signal bandwidth and spreading factor due to the integration gain available in each combination. Figure 13 shows a summary of the results of the SNR investigation. The plot represents the SNR required, for each LoRa signal bandwidth, to achieve successful classification of 99% of LoRa database signals for each spreading factor. The lowest SNR which achieved successful characterisation of LoRa signals for both 125 kHz and 250 kHz bandwidths is −15 dB, however, that value is achieved with different spreading factors for each bandwidth.

The results shown previously have all contained the signal of interest centred on the centre frequency of the channelised data and close to the left hand edge of the spectrogram image. A qualitative evaluation of the processing chain has been performed to ensure that the location of the downchirp within the spectrogram image is not critical. The LoRa signal in the time-frequency domain is transformed to different locations within the image. The signal is not changed except in its location. The processing chain is run against this modified data to assess its success in alternative signal locations. The overlaid characterisation is shown in Figure 14 which illustrate successful operation in scenarios where the signal of interest resides in various time-frequency locations.

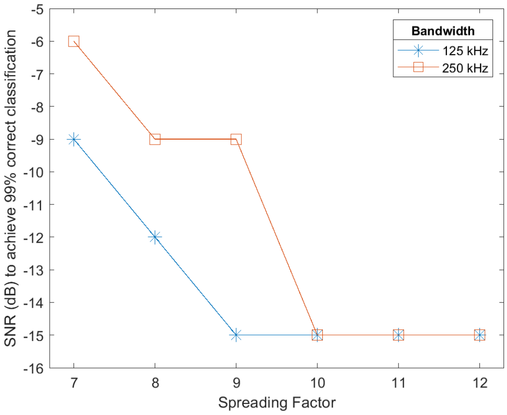


FIGURE 13. SNR required to achieve 99% successful parameter characterisation over LoRa signal bandwidth and spreading factor.

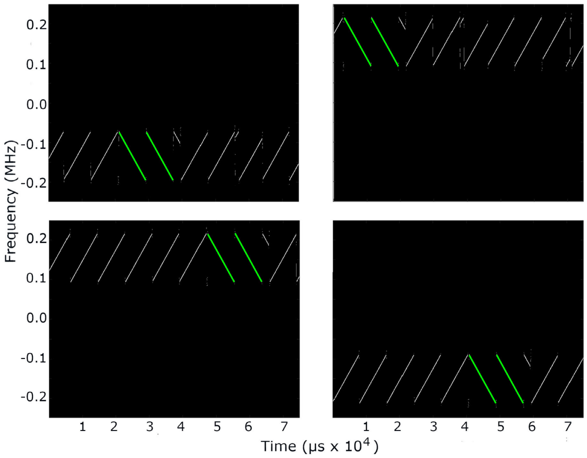


FIGURE 14. Characterisation parameters overlaid on source images with signal in different time-frequency locations.

An additional qualitative evaluation of the processing chain was carried out against a congested environment containing multiple LoRa signals overlapping in time and frequency. Given the prolific adoption of LoRa this is not an unlikely scenario, and will become very likely if parallel transmission strategies such as that presented in [30] are adopted. The source data for this test was synthetically created by combining three elements from the signal database each containing signals at similar centre frequencies and times. Two of the signals have bandwidths of 125 kHz, the third has a bandwidth of 250 kHz. The spectrogram created from this dataset is seen to be highly cluttered towards the top right hand side

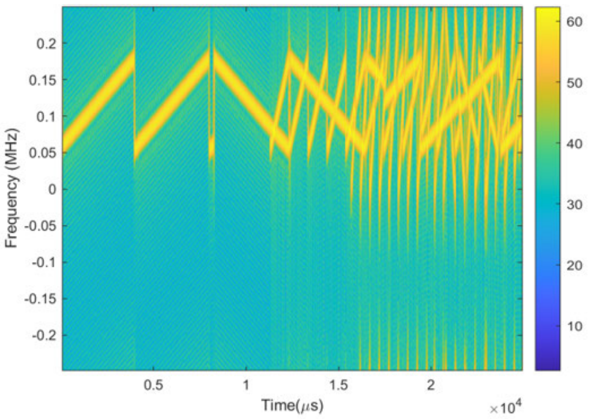


FIGURE 15. Spectrogram image of congested LoRa reception created by superimposing multiple over the air captures of individual LoRa signals.

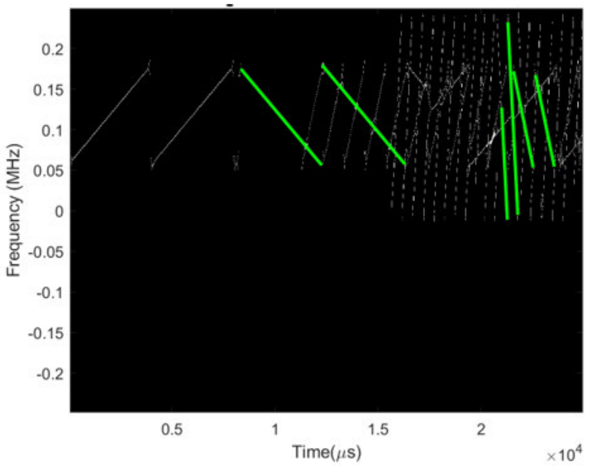


FIGURE 16. Characterisation parameters overlaid on source image of congested environment containing multiple overlapping LoRa signals.

of the image, and not easy to interpret by eye, Figure 15. The characterisation results are illustrated in Figure 16. The down chirps from all three LoRa signals were identified. One of the 250 kHz bandwidth chirps is seen to have been detected with a somewhat reduced bandwidth. This result shows that the techniques described in this paper have the ability to work successfully in congested scenarios.

D. PERFORMANCE COMPARISON

As there is little in previous literature of a similar nature to this work, performance comparisons for our algorithm is limited.

The performance of the algorithm in [13] was specified in terms of percentage of mean Average Precision (mAP %) across all the simulation cases, or across the captured data cases, without any definition of the range of SNRs considered. For this reason no comparison of performance is possible.

A more meaningful performance comparison can be made against [14]. In this work close to 100% successful parameter estimation was achieved for SNR values of 0 dB and above, which is in line with our results. However, in [14] the authors state that successful classification degrades 'drastically' below SNRs of −5 dB, whereas, at least for higher spreading factors, our algorithm provides very high success rates down to SNRs of −15 dB.

IV. EXPERIMENTAL HARDWARE AND CONFIGURATION

A. THE ARESTOR RF PLATFORM

ARESTOR [5] is a multi-function RF sensor platform based on the Xilinx radio frequency system on a chip (RFSoC) [31]. It is based on the generation 1 RFSoC device, specifically it employs one or more ZCU111 development boards [32], also produced by Xilinx.

The characteristic which differentiates the RFSoC from other SoC devices is the inclusion of RF subsystems on-chip. In the case of the RFSoC on the ZCU111 board this consists of eight 6.554 GS/s digital to analogue converters (DAC), and eight 4.096 GS/s analogue to digital converters (ADC). These converters provide direct RF generation and capture capabilities which would usually require multiple additional devices.

The ARESTOR platform takes the raw RFSoC and provides a framework for multi-function RF sensing. An array of custom IP components and software have been developed to provide building blocks for multiple radar and electronic surveillance applications. These building blocks are combined, configured and built into different system solutions by a combination of TCL and Python scripting based on a high-level design specification file. Examples of the solutions created are a frequency modulated continuous wave (FMCW) radar, an FMCW radar capable of capturing full polarisation information, a multi-band FMCW radar operating in ISM bands at 2.4 GHz and 5.8 GHz [33], an active-passive radar featuring simultaneous capture of target returns from an FMCW active radar and a WiFi illuminator based passive radar [34], and a dual-function radar and communication (DFRC) device [35].

For this study, a new configuration of the ARESTOR platform was developed which targeted the digitisation and analysis of LoRa signals over their full parameter space. Details of this configuration are given in Section V. A single receiver channel was used on the ZCU111 with an ANT-8WHIP3H-SMA omni-directional antenna connected via a MiniCircuits VBFZ-925-S+ bandpass filter.

B. THE FiPy IoT TRANSCEIVER

The FiPy [36] is a Micropython based IoT development board capable of transmitting and receiving WiFi, Bluetooth, LoRa, Sigfox and LTE-M signals. FiPys were used as a source of LoRa signals for all the experiments conducted in this study. The FiPy was connected to a ANT-8WHIP3H-SMA omni-directional antenna. During all experiments, the FiPy was placed at a distance of ∼1 m from the ARESTOR system.

Figure 17 shows a photo of the hardware as configured for LoRa data capture with the ARESTOR platform on the left marked 'A' and the FiPy module on the right, 'B'.

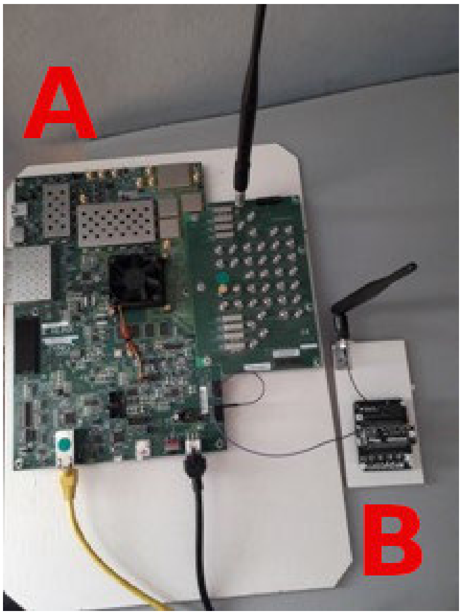


FIGURE 17. Image showing ARESTOR platform (A) and FiPy device (B).

V. DEVELOPMENT OF REAL-TIME LoRa CLASSIFICATION

Section III presented an algorithm based on the Hough transform for detecting LoRa signals and extracting their key parameters without decoding their message payload. A Matlab implementation of this algorithm was shown to be effective on real data even at low SNR and when multiple LoRa signals are overlapping in frequency and time. However, when run on an i7 laptop PC this implementation takes on average >0.5 seconds to process 75 ms of a single channelised channel of captured data, making it unsuitable for real-time applications. To address this, we adapted the algorithm to leverage the signal processing capabilities of our hardware platform, which provided sufficient acceleration to allow real-time operation. A further benefit of this ‘‘edge processing’’ approach is that very little data transfer is required compared to a more centralised processing technique.

As indicated previously, the hardware platform employed in this work is the RFSoC based ARESTOR platform, which had previously been developed as a multi-role RF sensing system [5]. The functionality of ARESTOR did not exactly match the requirements of this research, but the platform has been designed from the ground up with the ability to be flexible and extensible. This allowed the straightforward inclusion of new processing blocks to cater for the specific requirements of this project. For real-time classification of signals a non co-operative mode of operation between ARESTOR and the FiPy modules is used. This mode is similar to that illustrated in Figure 3, but excludes the trigger signal. This change decouples the two devices such that ARESTOR has no *a priori* knowledge of the transmitted signal. Results are verified offline by comparing the sequence of symbols sent by the FiPy with the sequence classified by ARESTOR.

A. REAL-TIME HARDWARE IMPLEMENTATION

Figure 18 shows an overview of the hardware-accelerated data processing chain for real-time LoRa classification. The different algorithm stages were partitioned across various elements of the RFSoC system depending on their suitability for different types of hardware acceleration. Details of each stage are discussed below. Overall, the algorithm implemented on hardware closely resembles the software implementation but with the addition of a noise rejection stage and using a peak-detect stage rather than a thinning stage.

The incoming signal from the antenna is continuously digitised at a sample rate of 4.08 GHz. The data are then down-mixed to baseband and decimated by a factor of 8, using the hardware available on the RF tile of the RFSoC. Further decimation by a factor of 64 is achieved using a decimation filter implemented in the FPGA fabric. This follows the same design as that described in [5]. The resulting complex baseband signal covers ∼8 MHz of bandwidth, the same as that recorded for the database described above, and sufficient to record the full LoRa band.

Subdividing the incoming data into overlapping channels of 500 kHz bandwidth is achieved using an oversampled polyphase filter bank, [37], implemented in the FPGA fabric. This is a resource-efficient method of channelising data often found in FPGA designs. Its key benefit being that data can be decimated prior to being filtered, allowing lower-throughput filter designs to be used. A good overview of the technique is given in [38]. In our implementation, we use a Hanning window with 256 taps as our prototype filter.

As with the software algorithm presented above, a spectrogram is produced for each of the 500 kHz channels using the Short-Time Fourier Transform (STFT) method. The same window size, overlap and zero padding factors are used as above. The high degree of parallelism of this process makes it well suited to acceleration using the FPGA. Creation of the spectrograms is split across four identical processing pipelines, each dealing with 8 channels in parallel. Figure 19 shows how each pipeline is implemented in hardware, using a block-RAM (BRAM) FIFO to buffer data so that it can be reused for the overlapping segment of the subsequent window. Following the data-overlap, a windowing function (a Hanning window in this case) is applied by multiplying the data by pre-computed windowing values read out of a single BRAM buffer, which is shared between all four pipelines. The data are then prepended with zeros up to the zero-padding amount and passed into a standard Xilinx FFT block for processing.

A crucial difference between the real-time system and the software algorithm described above is that the latter is fed with finite packets of data which are known to contain a LoRa signal whereas the former must analyse a continuous stream of data, much of which will only contain noise. The zero-padding and data overlap of the spectrogram creation results in a large increase in the quantity of data samples that must be passed into the remainder of the processing chain

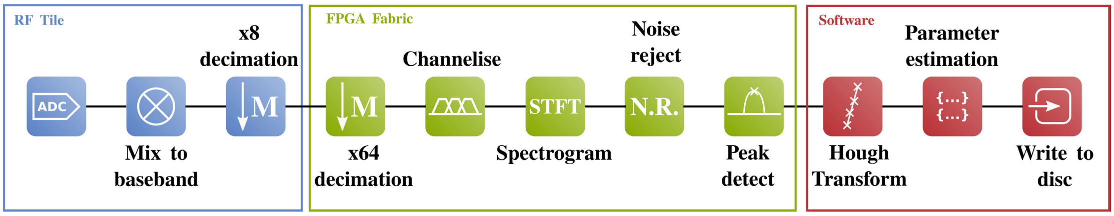


FIGURE 18. Block diagram of the real-time digital processing chain for LoRa classification showing the partitioning of processing steps between different elements of the RFSoC.

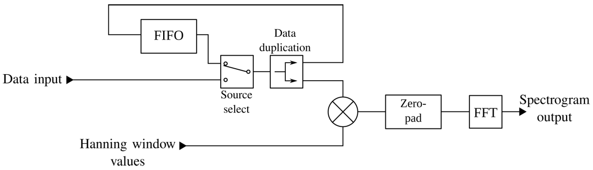


FIGURE 19. Block diagram showing the main elements of a single spectrogram processing pipeline. Each pipeline can process 8 channels of data in parallel. Four pipelines are used in the design to handle the 32 overlapping 500 kHz channels.

(256 output samples for every 6 input samples). An obvious way to reduce this, is to filter out samples which are just noise. Distinguishing noise from valid signal is achieved by comparing the frequency distribution of energy in each STFT time window to that of a pre-recorded reference signal which contained only noise. The comparison, Eq. 8, is made by considering the normalised cumulative distribution functions of the STFT time window under test (*Sig.*) and the reference (*Ref.*), and computing the maximum difference between the two, τ, (somewhat analogous to a Kolmogorov–Smirnov test) as shown in Fig. 20. Time windows with a maximum difference, τ, of greater than 0.19 are considered to contain signal. The threshold value of 0.19 was determined empirically.

$$\tau = \max_{0 \leq i \leq 255} (|F(Sig.)_i - F(Ref.)_i|)$$
$$F(x)_i = \frac{\sum_{j=0}^{i} x_j}{\sum_{j=0}^{255} x_j} \tag{8}$$

The software algorithm described in Section III employs a thinning stage to reduce the width of LoRa chirp signatures in the spectrogram prior to applying the Hough transform. Thinning algorithms are typically iterative and require random access to the spectrogram data, making them unsuitable for acceleration on the hardware available on the RFSoC. For the real-time system therefore, we do not apply thinning to the data and instead use a simple peak-detect (maximum value) to select the centre of the chirp signature in each spectrogram time window. Using a maximum value approach here is valid since the time slices are known to contain signal due to the previous noise-rejection processing. However, it assumes that there is only one valid signal present in each time window of any given channel, which in a highly congested/contested environment may not be the case (i.e. if two LoRa signals whose frequency ranges are spaced by less than 500 kHz occurred simultaneously, they would not be correctly detected). For this study, we have accepted this limitation in favour of simplified processing. Future work may look to leverage additional processing hardware (perhaps a Graphics Processing Unit, GPU) in conjunction with the RFSoC to implement a more advanced thinning algorithm in real-time.

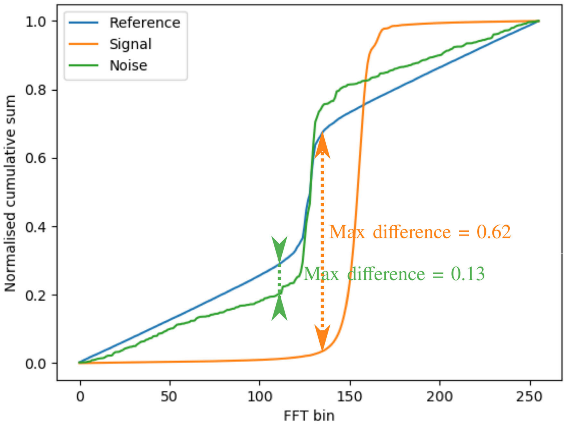


FIGURE 20. Plot showing the normalised cumulative distribution function for the noise reference, a noise sample and a sample containing signal. The annotation shows the maximum difference between the distributions. This is used as a metric to discriminate between signal and noise.

As shown in Fig. 21, the output from the peak-detect stage of processing is a series of coordinates in frequency-time space which represent the received LoRa chirps. Each coordinate is encoded as a 32 bit value, which also contains the channel number that the point was received on. The data from all channels is collated in the FPGA and transferred into RAM using a combination of direct memory access (DMA) and zero-copy driver software as described in [5]. The remainder of the processing is achieved in software running on the application processing unit (APU) under a Linux operating system.

The parallel nature of the Hough transform is exploited by implementing segments of the algorithm on the Neon co-processor which is present in the APU. This is a single instruction multiple data (SIMD) processor which greatly speeds up the computation. Combining this with the restricted angle subset approach described in Section III facilitates real-time operation. Parameters of the lines identified by the Hough transform are then extracted and matched to possible

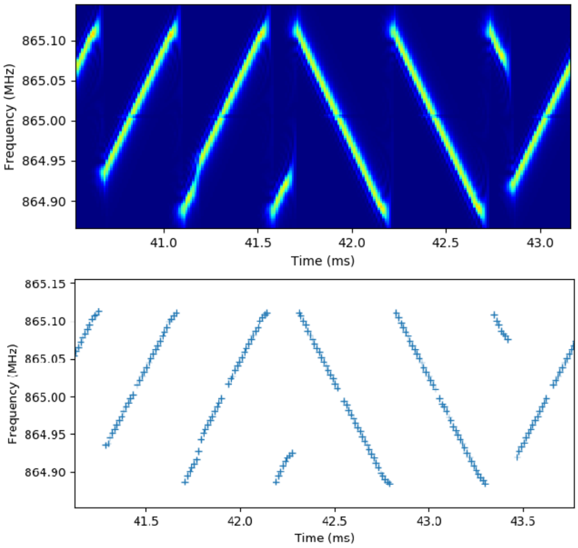


FIGURE 21. Segment of spectrogram containing a LoRa signal (top) and corresponding output data from the peak-detect processing stage (bottom). The frequency and time extents of the plotted data have been cropped for clarity.

LoRa parameters. The identified parameters of successfully classified signals are then written to disc.

The real-time processing chain described above consumes approximately 40% of the FPGA resources available on the ZU28DR. This is broken down as: 37% look-up-table (LUT) usage, 39% flipflop (FF) usage, 11% BRAM use and 9% digital signal processing slice (DSP) usage. Approximately half of the LUT and FF use is in the high speed division blocks required for calculating the normalised cumulative sum for the noise rejection algorithm. Division is known to be a resource-intensive operation to implement in FPGAs (compared to multiplication for example) and it is possible that our design could be streamlined by rearranging equation 8 such that only multiplication operations were required. However, this was not attempted during this study.

B. REAL-TIME CLASSIFICATION RESULTS

The real-time system was tested using a setup similar to that shown in Fig. 3, but without any trigger signal connection and without the attenuator installed. ARESTOR was configured to run the real-time LoRa classification processing chain described above. Once this had been running for a short period of time, the FiPy was programmed to transmit LoRa signals as fast as it could, stepping through all possible combinations of spreading factor, bandwidth and centre frequency (863-870 MHz with a step size of 0.5 MHz). The full parameter space sweep was repeated 10 times, resulting in a total of 1800 transmissions. The sequence of classified signals was then compared to the known sequence of transmitted signals. The results showed a 100% success rate, with all LoRa transmissions being correctly classified by the ARESTOR system.

VI. CONCLUSION

In this paper, we have presented a Hough-transform based algorithm for detecting LoRa transmissions and extracting their primary parameters. A database of test signals which covers the full parameter space of centre-frequency, bandwidth and spreading factors that LoRa can occupy was created using the ARESTOR platform as a receiver. This database was used to evaluate the performance of the algorithm, which showed excellent performance down to SNRs of -15 dB depending on bandwidth and spreading factor. The algorithm also showed success for congested test cases where three separate LoRa transmissions were overlapped in frequency and time.

The algorithm was adapted to run on the ARESTOR hardware platform. Real-time operation was achieved by mapping algorithm components to different hardware subsystems in order to make best use of their processing capabilities. Tests of the real-time system demonstrated a 100% classification success rate, albeit at high SNR and in an uncongested environment. Future work will look to test and improve the real-time system at lower SNRs and in more congested/contested scenarios.

This work has shown that the combination of direct-RF sampling and powerful signal processing capabilities present in RFSoC based hardware makes it a good candidate for performing spectrum-survey/signal-classification tasks at the edge. It has also demonstrated that, at least for LoRa waveforms, classical algorithms (i.e. not based on machine learning or artificial intelligence) can provide good classification performance and can readily be accelerated to achieve real-time performance.

ACKNOWLEDGMENT

The authors would also like to thank Dstl and The Alan Turing Institute for their technical input into this project.

REFERENCES

[1] Z. Alliance. (2015). *ZigBee Specification*. Accessed: Mar. 1, 2022. [Online]. Available: https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf

[2] A. I. Petrariu and A. Lavric, "SigFox wireless communication enhancement for Internet of Things: A study," in *Proc. 12th Int. Symp. Adv. Topics Electr. Eng. (ATEE)*, Mar. 2021, pp. 1–4.

[3] (2011). *Lorawan Specification*. Accessed: Mar. 1, 2022. [Online]. Available: https://lora-alliance.org/resource-hub/lorawanr-specification-v11

[4] R. Want, B. Schilit, and D. Laskowski, "Bluetooth LE finds its niche," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 12–16, Oct. 2013.

[5] N. Peters, C. Horne, and M. A. Ritchie, "ARESTOR: A multi-role RF sensor based on the Xilinx RFSoC," in *Proc. 18th Eur. Radar Conf. (EuRAD)*, Apr. 2022, pp. 102–105.

[6] A. Augustin, J. Yi, T. Clausen, and W. Townsley, "A study of LoRa: Long range & low power networks for the Internet of Things," *Sensors*, vol. 16, no. 9, p. 1466, Sep. 2016.

[7] P. Robyns, E. Marin, W. Lamotte, P. Quax, D. Singelée, and B. Preneel, "Physical-layer fingerprinting of LoRa devices using supervised and zero-shot learning," in *Proc. 10th ACM Conf. Secur. Privacy Wireless Mobile Netw. (WiSec)*, Boston, MA, USA, Jul. 2017, pp. 58–63.

[8] N. Chen, A. Hu, and H. Fu, "LoRa radio frequency fingerprint identification based on frequency offset characteristics and optimized LoRaWAN access technology," in *Proc. IEEE 5th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Mar. 2021, pp. 1801–1807.

[9] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for LoRa using deep learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2604–2616, Aug. 2021.

[10] G. Shen, J. Zhang, A. Marshall, and J. R. Cavallaro, "Towards scalable and channel-robust radio frequency fingerprint identification for LoRa," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 774–787, 2022.

[11] A. Elmaghbub and B. Hamdaoui, "LoRa device fingerprinting in the wild: Disclosing RF data-driven fingerprint sensitivity to deployment variability," *IEEE Access*, vol. 9, pp. 142893–142909, 2021.

[12] K. S. V. Prasad, K. B. Dsouza, and V. K. Bhargava, "A downscaled faster-RCNN framework for signal detection and time-frequency localization in wideband RF systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4847–4862, Jul. 2020.

[13] E. Lo and J. Kohl, "Internet of Things (IoT) discovery using deep neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 795–803.

[14] L. Angrisani, M. D'Arco, C. Dassi, and A. Liccardo, "LoRa signals classification through a CS-based method," in *Proc. IEEE 4th Int. Forum Res. Technol. Soc. Ind. (RTSI)*, Sep. 2018, pp. 1–5.

[15] A. Vagollari, V. Schram, W. Wicke, M. Hirschbeck, and W. Gerstacker, "Joint detection and classification of RF signals using deep learning," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Apr. 2021, pp. 1–7.

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

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

[18] M. Knight and B. Seeber, "Decoding LoRa: Realizing a modern LPWAN with SDR," in *Proc. GNU Radio Conf.*, vol. 1, Sep. 2016, pp. 1–5.

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

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

[21] B. Willetts, M. Ritchie, and H. Griffiths, "Optimal time-frequency distribution selection for LPI radar pulse classification," in *Proc. IEEE Int. Radar Conf. (RADAR)*, Apr. 2020, pp. 327–332.

[22] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[23] P. V. C. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3 069 654, Dec. 18, 1962.

[24] R. O. Duda and R. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[25] A. Pieniezny, "Intrapulse analysis of radar signal by the use of Hough transform," in *Proc. Int. Conf. Modern Problems Radio Eng., Telecommun. Comput. Sci. (TCSET)*, Feb. 2008, pp. 306–309.

[26] F. Li, Z. Yang, and C. Yang, "Radar signal sorting technology based on image processing and Hough transform," in *Proc. Int. Conf. Microw. Millim. Wave Technol. (ICMMT)*, May 2018, pp. 1–3.

[27] F. Xu, Q. Bao, Z. Chen, S. Pan, and C. Lin, "Parameter estimation of multi-component LFM signals based on STFT+Hough transform and fractional Fourier transform," in *Proc. 2nd IEEE Adv. Inf. Manage., Communicates, Electron. Autom. Control Conf. (IMCEC)*, May 2018, pp. 839–842.

[28] E. Yar, M. B. Kocamis, A. Orduyilmaz, M. Serin, and M. Efe, "A complete framework of radar pulse detection and modulation classification for cognitive EW," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2019, pp. 1–5.

[29] K. K. Guner, T. O. Gulum, and B. Erkmen, "FPGA-based Wigner–Hough transform system for detection and parameter extraction of LPI radar LFMCW signals," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–15, 2021.

[30] G. Zhu, C.-H. Liao, T. Sakdejayont, I.-W. Lai, Y. Narusue, and H. Morikawa, "Improving the capacity of a mesh LoRa network by spreading-factor-based network clustering," *IEEE Access*, vol. 7, pp. 21584–21596, 2019.

[31] Xilinx. *Zynq UltraScale+ RFSoC*. Accessed: Aug. 13, 2022. [Online]. Available: https://www.xilinx.com/products/silicon-devices/soc/rfsoc.html

[32] Xilinx. *ZCU111 Evaluation Kit*. Accessed: Aug. 13, 2022. [Online]. Available: https://www.xilinx.com/products/boards-and-kits/zcu111.html

[33] D. Dhulashia, N. Peters, C. Horne, P. Beasley, and M. Ritchie, "Multi-frequency radar micro-Doppler based classification of micro-drone payload weight," *Frontiers Signal Process.*, vol. 1, p. 14, Dec. 2021.

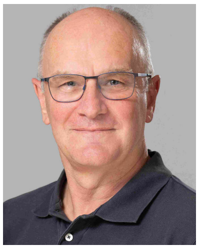
[34] M. Ritchie, N. Peters, and C. Horne, "Joint active passive sensing using a radio frequency system-on-a-chip based sensor," in *Proc. 23rd Int. Radar Symp. (IRS)*, Sep. 2022, pp. 130–135.

[35] M. Temiz, C. Horne, N. J. Peters, M. A. Ritchie, and C. Masouros, "An experimental study of radar-centric transmission for integrated sensing and communications," *IEEE Trans. Microw. Theory Techn.*, early access, Jan. 16, 2023, doi: 10.1109/TMTT.2023.3234309.

[36] Pycom. *Fipy*. Accessed: Aug. 12, 2022. [Online]. Available: https://pycom.io/product/fipy/

[37] F. J. Harris, C. Dick, and M. Rice, "Digital receivers and transmitters using polyphase filter banks for wireless communications," *IEEE Trans. Microw. Theory Techn.*, vol. 51, no. 4, pp. 1395–1412, Apr. 2003.

[38] J. P. Smith, "A high-throughput oversampled polyphase filter bank using Vivado HLS and PYNQ on a RFSoC," *IEEE Open J. Circuits Syst.*, vol. 2, pp. 241–252, 2021.



COLIN HORNE (Member, IEEE) received the B.Sc. (Eng.) degree in computer systems and electronics from King's College London, and the Ph.D. degree in electronic and electrical engineering from University College London (UCL), in 2020, on the topic of cognitive radar.

He spent several years with Plessey Research Roke Manor researching land-based and airborne bistatic radar. He subsequently worked on various radar and defense related projects in industry, Defence Research Agency and Diehl GmbH, Nurnburg, before becoming a partner in a small software company providing management and planning tools to manufacturing companies. He is currently a Research Fellow with the Radar Group, UCL, working on various projects applying the Xilinx RFSoC systems to reconfigurable multi-role RF sensing.

NIAL J. PETERS (Member, IEEE) received the B.A. and M.Sc. degrees in natural sciences (physics) from Cambridge University, in 2007, the joint M.Sc. degree in space and plasma physics from The University Centre in Svalbard (UNIS) and Oslo University (UiO), in 2009, and the Ph.D. degree in volcanology from Cambridge University, in 2010, focusing on instrumentation development and data processing methodologies for monitoring Mount Erebus volcano, Antarctica. After completing his Ph.D. in 2014, he undertook a postdoctoral research project split between Cambridge University and University College London (UCL) to design, build, and deploy an FMCW radar system for volcanic monitoring. He is currently a full-time Research Fellow with UCL, where he works on a range of radar-related projects, including target classification using micro-Doppler, spectrum-survey for the IoT devices, and multi-role RF systems based on Xilinx RFSoC hardware.

MATTHEW A. RITCHIE (Senior Member, IEEE) received the M.Sc. degree in physics from the University of Nottingham, in 2008, and the Eng.D. degree from University College London (UCL), London, U.K., in association with Thales U.K., in 2013.

He was a postdoctoral research associate focusing on machine learning applied to multistatic radar for micro-Doppler classification. In 2017, he was a Senior Radar Scientist with Defence Science and Technology Laboratories (Dstl). He is currently an Associate Professor with the Radar Sensing Group, UCL, and focused on areas, including multistatic radar, passive radar, micro-Doppler, and multi-role RF sensor hardware. He is the Head of the U.K. EMSIG Society. He was a recipient of the 2017 IET RSN Best Paper Award and the Bob Hill Award at the 2015 IEEE International Radar Conference. He also serves as the Chair for the IEEE Aerospace and System Society (AESS) in U.K. and Ireland. He is also a Subject Editor-in-Chief of the *IET Electronics Letters* journal.