Generic Compression of Off-The-Air Radio Frequency Signals with Grouped-Bin FFT Quantisation

Damien Muir
Department of Electronic and Electrical Engineering
University of Strathclyde
Glasgow, Scotland
damien.muir@strath.ac.uk

Louise H. Crockett
Department of Electronic and Electrical Engineering
University of Strathclyde
Glasgow, Scotland
louise.crockett@strath.ac.uk

Robert W. Stewart
Department of Electronic and Electrical Engineering
University of Strathclyde
Glasgow, Scotland
r.stewart@strath.ac.uk

***Abstract*—This paper studies the capabilities of a proposed lossy, grouped-bin FFT quantisation compression method for targeting Off-The-Air (OTA) Radio Frequency (RF) signals. The bins within a 512-point Fast Fourier Transform (FFT) are split into groups of adjacent bins, and these groups are each quantised separately. Additional compression can be achieved by setting groups which are not deemed to contain significant information to zero, based on a pre-defined minimum magnitude threshold. In this paper, we propose two alternative methods for quantising the remaining groups. The first of these, Grouped-bin FFT Threshold Quantisation (GFTQ), involves allocating quantisation wordlengths based on several pre-defined magnitude thresholds. The second, Grouped-bin FFT Error Quantisation (GFEQ), involves incrementing the quantisation wordlength for each group until the calculated quantisation error falls below a minimum error threshold. Both algorithms were tested for a variety of signal types, including Digital Private Mobile Radio 446 MHz (dPMR446), which was considered as a case study. While GFTQ allowed for higher Compression Ratios (CR), the compression process resulted in added quantisation noise. The GFEQ algorithm achieved lower CRs, but also lower noise levels across all test signals.**

***Index Terms*—radio, compression, generic, RF**

I. Introduction

With the current roll-out of next-generation 5G architectures, Radio Frequency (RF) compression has become an area of interest – particularly as a means to increase throughput and the efficiency of fibre bandwidth [1]. RF Compression techniques have been devised for both the fronthaul and backhaul of 5G networks over the past few years [2], [3].

Many pre-existing compression algorithms have been successfully re-purposed for the compression of RF signals. Interestingly, some of these algorithms invoke parallels with the audio compression journey. These include companding solutions such as mu-law [4], or speech compression approaches like Adaptive Differential Pulse Code Modulation (ADPCM) and Linear Predictive Coding (LPC) [5], [6], [7]. Compression methods from other fields – such as the Lloyd algorithm, or Internet Protocol payload compression using a Two-Way Active Measurement Protocol (TWAMP) server – have likewise been successfully adapted to RF compression [8], [9].

The aforementioned compression approaches have most commonly been tested for Radio-over-Fibre (RoF), assume a clean signal with negligible noise, and occasionally assume some knowledge of the signal to be compressed. However, there is also scope for the compression of noisy, Off-The-Air (OTA) signals with unknown characteristics. This paper serves to propose generic, lossy RF compression methods which may find application over multiple use-cases.

Candidate applications for these techniques are not solely limited to the field of mobile architectures. RF compression has also been proved useful in the field of spectrum sensing as a reliable method of signal detection [10]. There is likewise potential for RF compression to be used for public safety or environmental monitoring applications. A generic OTA compression algorithm could allow for remote locations of importance to receive transmitted RF signals in the area – such as Private- or Professional-Mobile-Radio (PMR) signals – and potentially store the compressed data for later analysis. Such an algorithm could increase RoF throughput for such applications by taking advantage of spectral redundancy.

To this end, the compression of Digital Private Mobile Radio signals at 446 MHz (dPMR446) will be explored as a case study in this paper. dPMR446 is a licence-free, Frequency Shift Keying (FSK) modulated protocol that complies with the European Telecommunications Standards Institute (ETSI) TS 102 490 standard, and is capable of voice, data, and voice+data modes of operation [11]. There are sixteen 12.5 kHz dPMR446 channels and/or thirty-two 6.25 kHz dPMR446 channels equally spaced over the 446-446.2 MHz band in Europe [12], and the interface itself is very similar to the Next Generation Digital Narrowband (NXDN) open standard [13].

In addition to dPMR446, a range of analogue and digital speech signals were modulated using a set of different modulation schemes to provide context and a more diverse testing regime. These modulation schemes included: Amplitude Modulation (AM), Frequency Modulation (FM), Quadrature

Amplitude Modulation (QAM), and Orthogonal Frequency Division Multiplexing (OFDM). All results were obtained using floating point MATLAB and Simulink simulations for a range of noise values. All test signals were likewise generated using MATLAB and were compressed at baseband.

The remainder of this paper will be organised as follows: two frequency-domain compression algorithms will be proposed, with a brief explanation of the functionality of each. The performance of each will then be explored in terms of achieved CRs and error rates, based on results achieved in a MATLAB test environment.

II. Proposed Compression Algorithms

The proposed generic compression algorithms in this paper are the Grouped-bin Fast Fourier Transform (FFT) Threshold Quantisation (GFTQ) and Grouped-bin FFT Error Quantisation (GFEQ) algorithms. Both involve frequency-domain compression via the quantisation of rectangular-windowed, non-overlapped FFTs, but have different methods of assigning quantisation wordlengths.

A. Grouped-bin FFT Threshold Quantisation (GFTQ)

The GFTQ algorithm is a frequency-domain compression approach that relies on grouping equal numbers of adjacent FFT bins and compressing these groups independently. For example, given an 512-point FFT, 8 groups of 64 bins could be formed, each of which would be compressed separately. The compressed groups are then recombined and transformed using an inverse FFT to reconstruct the signal.

Note that the number of groups can be varied: e.g., a 512-point FFT could also have 4 groups of 128 bins or 32 groups of 16 bins. There is a positive correlation between higher numbers of groups and higher compression ratios, but an increase in groups will necessitate a larger overhead between the compressor and reconstructor. This overhead must communicate the quantisation wordlength of each group to the reconstructor for each compressed FFT window. Thus, a balance between number of groups and size of overhead must be struck to achieve the highest possible compression ratio (CR). The ideal number of groups for a 512-point FFT would then appear to be 32, as beyond this the overhead would become large enough to actually decrease the compression ratio. This correlation between pre-overhead CR and number of groups is shown by the results tabulated in Table I.

Prior to the FFT stage, a buffer of equal width to the FFT is used to block-scale the received signal. The 512 samples

TABLE I
Performance metrics for number of FFT groups

Parameter	Num Groups	AM	QAM64	OFDM
Minimum MSE	4	0.007	0.01	0.005
	8	0.003	0.001	0.01
	16	0.005	0.003	0.01
	32	0.007	0.003	0.01
Pre-Overhead Compression Ratio	4	3.5:1	7:1	6:1
	8	5.7:1	14:1	9:1
	16	9:1	27:1	14:1
	32	13:1	55:1	27:1

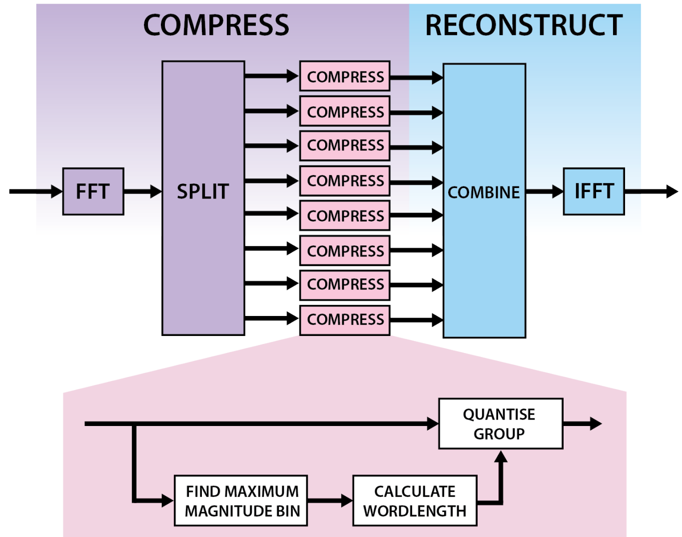

Fig. 1. Top Level Diagram of GFTQ algorithm

in this buffer are divided by the highest-magnitude sample within the buffer. This normalises each sample and ensures the quantisation range is consistent throughout. The block-scaling is reversed following the IFFT via a multiplication with the maximum-magnitude bin, which has been communicated to the reconstructor via the overhead.

A top-level block diagram of the GFTQ algorithm is shown in Fig. 1.

The maximum magnitude bin (B_{Max}) in each group (pre-block scaling) is compared to a set of pre-coded threshold values. If B_{Max} falls between two threshold values, it will be quantised with a uniform, linear quantiser using the quantisation wordlength, W_Q, corresponding to the lower of these values. For example, should B_{Max} fall between the 5-bit and 6-bit thresholds, then W_Q would be set to 5 bits. All bins within that group would then be quantised with 5 bits. Fig. 2 shows an illustration of an FFT window split into groups to be compressed via GFTQ.

For the experiment documented in this paper, the quantisation wordlength was allocated to each group using the

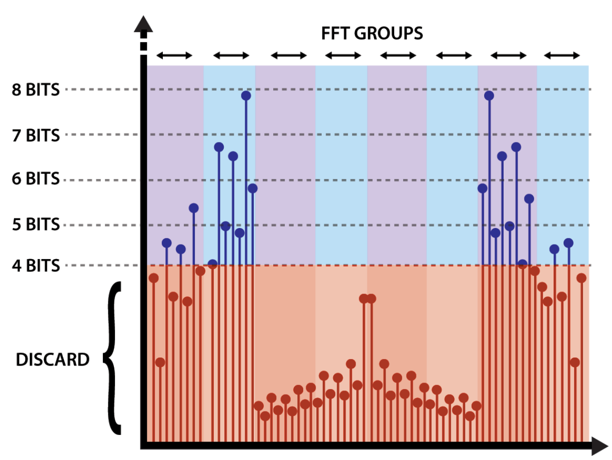

Fig. 2. FFT Groups with Magnitude Thresholds

amplitude thresholds shown in (1). Note that all groups where B_{Max} is less than or equal to 5 would be set equal to zero.

$$W_Q = \begin{cases} 8, & \text{if } B_{\text{Max}} > 80, \\ 7, & \text{if } 80 \geq B_{\text{Max}} > 60, \\ 6, & \text{if } 60 \geq B_{\text{Max}} > 40, \\ 5, & \text{if } 40 \geq B_{\text{Max}} > 20, \\ 4, & \text{if } 20 \geq B_{\text{Max}} > 5, \\ 0, & \text{if } B_{\text{Max}} \leq 5 \end{cases} \tag{1}$$

Equation (2) gives an estimate of how many bits would be required for an overhead (per FFT window) using the current grouped-bin approach, where N is the number of threshold levels, G is the number of groups, and N^G is the number of possible wordlength combinations. An additional 16 bits are added to represent the 16 bits required to reverse the block scaling process in the reconstructor.

$$Overhead = log_2(N^G) + 16 \tag{2}$$

It was assumed that the signal to be compressed was initially received in a 16-bit optimised format. Therefore, the compression ratio would be calculated as shown in (3), where CR is the compression ratio, N_S is the number of received samples, N_W is the total number of FFT windows to be compressed, and C_B is the sum total of quantisation wordlengths for all FFT bins over the compression period.

$$CR = \frac{16 \times N_{\text{S}}}{C_{\text{B}} + (N_{\text{W}} \times Overhead)} \tag{3}$$

B. Grouped-bin FFT Error Quantisation (GFEQ)

The GFEQ algorithm performs similarly to the GFTQ algorithm in terms of splitting the FFT window into separate groups to be quantised separately with a varying W_Q. However, while the GFTQ algorithm allocates W_Q based on magnitude thresholds, the GFEQ algorithm allocates W_Q based on the predicted total quantisation error *(E_Q)*.

Like with GFTQ, the B_{Max} of a given FFT group is compared to a pre-set minimum magnitude threshold. Should B_{Max} fall below that threshold, then this group is set equal to zero. However, should B_{Max} exceed the magnitude threshold, then all bins within the group are quantised.

For GFEQ, the E_Q of a given group for an initial W_Q is calculated, and this predicted error is compared to a pre-set maximum acceptable error value *(E_{Max})*. Should E_{Max} be exceeded, then W_Q is incremented and a new E_Q is calculated. This process repeats until E_Q falls below E_{Max}, or until a maximum W_Q is reached. This process is described by the recursive function in (4) & (5), where W_N is the current quantisation wordlength and E_{Q+1} is the quantisation error for an incremented quantisation wordlength, W_{N+1}:

$$W_Q = f(E_Q, E_{Max}) \tag{4}$$

$$f(E_Q, E_{Max}) = \begin{cases} W_N & \text{if } E_Q \leq E_{Max} \\ f(E_{Q+1}, E_{Max}) & \text{if } E_Q > E_{Max} \end{cases} \tag{5}$$

TABLE II
TEST SIGNAL CHARACTERISTICS

Modulation Scheme	Parameters
FM	• Frequency Deviation – 7.5 kHz
OFDM	• Number of Sinusoids – 64 subbands
QAM64	• Baud Time – 0.001 seconds • RRC Rolloff – 0.25 • RRC Span – 6 symbols

This approach allows for the level of error in the result to be controlled, making for a cleaner reconstructed signal (though with a probable cost to the compression ratio). Due to the recursive nature of this algorithm, this will also be more computationally expensive on any eventual hardware implementation.

III. Experimental Simulations and Results

Various single-channel test signals – all of which included modulated speech – were generated using MATLAB, with characteristics as detailed in Table II. All signals had a sampling rate of 64 kHz and a carrier frequency of 16 kHz. Note that the QAM signal was pulse-shaped using a Root-Raised Cosine (RRC).

A dPMR446 signal was also generated in MATLAB. Rather than model a full dPMR446 transmitter/receiver, the control data bits (such as the framesync, colour code, and traffic channel fields) were all populated with random data, while maintaining the structure of a standard dPMR446 signal. This synthetic signal was then compressed and the payload bits were extracted from the eventual reconstructed signal.

To simulate the reception of OTA data, all signals were mixed with varying levels of noise within the test environment of a Simulink model. This allowed each signal type and each compression approach to be tested for a range of noise values. The noisy signals were block-scaled, compressed using the proposed algorithms, and immediately reconstructed. The Mean Squared Error (MSE) was then calculated for each noise value by comparing the reconstructed signal at that noise value to the original (noiseless) signal. MSE was preferred to other measurements of error such as Bit Error Rate (BER) and Error Vector Magnitude (EVM) as it can be calculated for all signal types (analogue or digital) which is appropriate for generic algorithms such as those proposed by this paper.

A. GFTQ Compression Simulations

For the GFTQ simulations, 32 groups of 16 bins were used. When compressing the target signals, the results varied with the modulation scheme. This was to be expected given the different characteristics of each signal type and the different spreads of information across the FFT windows. For example, in Fig. 3, the FFTs of an OFDM signal (top) and an oversampled QAM64 signal (bottom) was treated differently by the GFTQ algorithm. The OFDM signal has a greater spread of information across all frequencies, whereas the information in the oversampled QAM64 signal is confined to the extremes of the FFT window. GFTQ can take advantage of the spectral redundancy in the case of the QAM64 signal

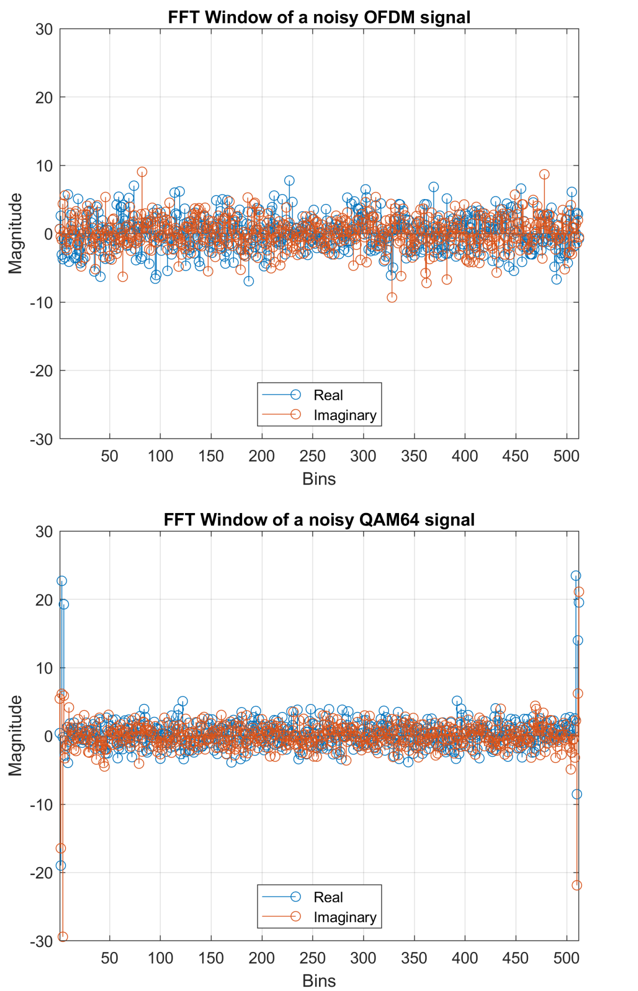

Fig. 3. OFDM (top) information density compared to oversampled QAM64 (bottom) in an FFT window

to achieve a higher compression ratio, which is not the case for the information-dense OFDM signal. This is reflected in the results tabulated in Table III. For comparison purposes, the CR achieved by existing ADPCM and LPC approaches were around 2.5:1 for QAM1024 and 5:1 for QAM64 respectfully, with clean input signals [5], [7].

The GFTQ algorithm was not able to compress the FM test signal in a manner that enabled it to be satisfactorily reconstructed. When attempting to compress the FM signal, the algorithm introduced significant quantisation noise – enough to overwhelm the information. It is hypothesised that the significant difference in magnitude between the carrier wave and information sidebands caused issues at the block scaling section of the algorithm, and that these errors propagated throughout the compression process.

TABLE III
GFTQ SIGNAL COMPRESSION RESULTS

Signal Type	MSE Floor	Min CR	Max CR
AM	0.007	1.2:1	10:1
OFDM	0.01	2.3:1	20:1
QAM64	0.003	4:1	33:1
dPMR446	0.15	2.3:1	5:1

However, the GFTQ algorithm was able to successfully compress each of the AM, OFDM, QAM64, and dPMR446 signals. The reconstructed audio of all signal types contained varying degrees of audible noise, but it was possible to interpret the speech in each case. The compression of the QAM64 signal outperformed all other signal types in terms of compression ratio, largely due to the aforementioned information sparsity.

Fig. 4 shows an MSE and compression ratio vs SNR curves for GFTQ-compressed dPMR446 simulations, where multiple points aligned on the same vertical axis correspond to a single noise value. The MSE of the compressed data is compared to the MSE of the uncompressed data to illustrate the quantisation error that has been introduced as part of the compression process – with the divergence between the two blue lines indicating this error. The compressed data levels off to an MSE "floor" (as referred to in Table III), which is taken as a measure of performance for all signal types. The compression of the dPMR446 signal resulted in the highest achieved MSE floor, indicating that the GFTQ algorithm may have some trouble compressing signals which contain structured data. It was possible to interpret the speech within the reconstructed audio signal, although there was significant, loud background noise. The MSE floor value of 0.15 is therefore considered to be too high to be desirable.

While designed to be a generic algorithm, GFTQ compression performed at varying levels depending on the signal type. As suspected, signals with high information sparsity such as oversampled QAM allowed for the most impressive CRs, while signals with more significant information across all frequencies caused the algorithm to be less effective.

B. GFEQ Compression Simulations

As with the GFTQ simulations, the GFEQ experiments made use of 32 of 16 bins within a 512-point FFT. Due to the

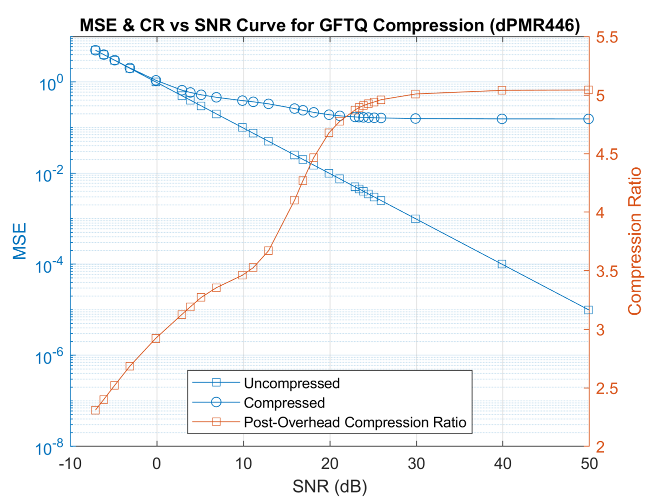

Fig. 4. MSE vs SNR and CR vs SNR Curves for GFTQ Compression of dPMR446 Voice Signal

TABLE IV
GFEQ SIGNAL COMPRESSION RESULTS

Signal Type	MSE Floor	Min CR	Max CR	Min Mag Threshold	Max Error Threshold
AM	0.001	1.3:1	5:1	5	0.3
FM	0.0006	1.7:1	2.4:1	1	0.05
OFDM	0.0001	2.8:1	3:1	2	0.5
QAM64	0.0005	3:1	33:1	2	0.5
dPMR446	0.01	2.2:1	4:1	10	0.2

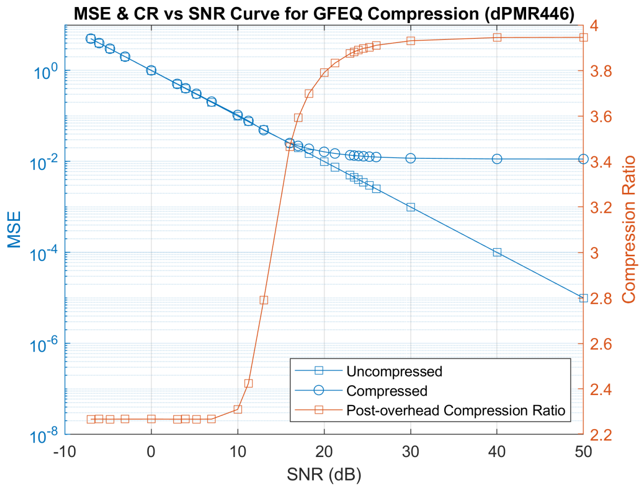

Fig. 5. MSE vs SNR and CR vs SNR Curves for GFEQ Compression of dPMR446 Voice Signal

similarities between the two algorithms, it was expected that the performances would be comparable, though with GFEQ achieving lower CRs in exchange for lower noise levels. GFEQ has an advantage over GFTQ in that the minimum-magnitude and maximum-error thresholds can be adjusted to better suit the characteristics of the signal type to be compressed. Notably, this flexibility allows for the compression of FM signals. What constitutes suitable threshold values will differ for each signal type, and as such the compression simulations took place with a non-uniform set of threshold values – which are listed along with the achieved MSEs and CRs in Table IV.

Fig. 5 shows the MSE and compression ratio vs SNR curves for a GFEQ-compressed dPMR446 signal. When compared to the GFTQ-compressed signal in Fig. 4, it is evident that the noise floor has dropped by an order of magnitude. While this comes at a cost to the compression ratio – which falls from 5:1 to 4:1 – the reconstructed signal is audibly of much better quality. The residual audible noise is significantly lower than that of the GFTQ-compressed signal.

A comparison between the results achieved with GFTQ and GFEQ compression shows that GFEQ allows for lower error rates at the expense of a smaller compression ratio. However, the ability of GFEQ to more suitably compress spectrum-sparse signals such as FM – or signals with a defined data structure such as dPMR446 – means that GFEQ can be considered more suitable as a generic RF compression algorithm.

IV. CONCLUSION

Several simulations suggest that it is possible to achieve significant compression of OTA RF signals using grouped-bin FFT quantisation compression techniques. Compression in the frequency domain allows compression algorithms to take advantage of spectral redundancy when it appears, resulting in higher compression ratios. However, this is not quite so effective with signals which contain significant energy across all frequencies – such as OFDM. Despite this, the proposed GFTQ amd GFEQ compression algorithms have achieved significant CRs for a range of signal types, proving that the generic compression of noisy OTA RF signals is viable, and that there is scope for the compression of OTA RF signals in the frequency domain.

REFERENCES

[1] M. Xu, Z. Jia, J. Wang, L. A. Campos and G. Chang, "Statistical data compression and differential coding for digital radio-over-fiber-based mobile fronthaul," in *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, no. 1, pp. A60-A71, Jan. 2019.

[2] S. Nanba and A. Agata, "A new IQ data compression scheme for front-haul link in Centralized RAN," *2013 IEEE 24th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops)*, London, 2013, pp. 210-214.

[3] S. Park, O. Simeone, O. Sahin and S. Shamai, "Multivariate backhaul compression for the downlink of cloud radio access networks," *2014 IEEE International Symposium on Information Theory*, Honolulu, HI, 2014, pp. 2699-2703.

[4] D. Brubaker, W. Qian, M. Sussmann, Y. Takafuji, M. Akhter, T. Hiatt, "The Emerging Need for Fronthaul Compression", Altera White Paper, June. 2016.

[5] P. Zhu, Y. Yoshida and K. Kitayama, "Real-time FPGA Demonstration of Low-latency Adaptive Fronthaul Compression based on Adaptive Differential Pulse Code Modulation," *2018 Asia Communications and Photonics Conference (ACP)*, Hangzhou, 2018, pp. 1-3.

[6] P. Cummiskey, N. S. Jayant and J. L. Flanagan, "Adaptive quantization in differential PCM coding of speech," in *The Bell System Technical Journal*, vol. 52, no. 7, pp. 1105-1118, Sept. 1973.

[7] G. Chen, F. Yang, K. Niu and C. Dong, "A linear predictive coding based compression algorithm for fronthaul link in C-RAN," *2017 IEEE/CIC International Conference on Communications in China (ICCC)*, Qingdao, 2017, pp. 1-6.

[8] M. Xu, Z. Jia, J. Wang, L. A. Campos and G. Chang, "A Novel Data-Compression Technology for Digital Mobile Fronthaul with Lloyd Algorithm and Differential Coding," *2018 Optical Fiber Communications Conference and Exposition (OFC)*, San Diego, CA, 2018, pp. 1-3.

[9] Y. Trk and E. Zeydan, "A TWAMP Coordinated Data Compression System for 5G Backhaul," *2019 IEEE Conference on Network Softwarization (NetSoft)*, Paris, France, 2019, pp. 245-247.

[10] A. R. Rosete, K. R. Baker and Y. Ma, "Using LZMA Compression for Spectrum Sensing with SDR Samples," *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York City, NY, USA, 2018, pp. 282-287.

[11] European Telecommunications Standards Institute (ETSI), "ETSI TS 102 490, Electromagnetic compatibility and Radio spectrum Matters (ERM); Peer-to-Peer Digital Private Mobile Radio using FDMA with a channel spacing of 6,25 kHz with e.r.p. of up to 500 mW", V1.3.1, April, 2007.

[12] Ofcom, "Analogue and Digital PMR446 Information Sheet: Business Radio", February 2018

[13] Next Generation Digital LMR Technology, "NXDN™ White Paper: 4 Level FSK/FDMA 6.25 kHz Technology", Rev 0.4, 2013.