Compressed Shaping: Concept and FPGA Demonstration

N.B. When citing this work, cite the original published paper.

(article starts on next page)

Compressed Shaping:
Concept and FPGA Demonstration

Tsuyoshi Yoshida, *Member, IEEE,* Koji Igarashi, *Member, IEEE,*
Magnus Karlsson, *Fellow, OSA; Senior Member, IEEE,* and Erik Agrell, *Fellow, IEEE*

Abstract—**Probabilistic shaping (PS) has been widely studied and applied to optical fiber communications. The encoder of PS expends the number of bit slots and controls the probability distribution of channel input symbols. Not only studies focused on PS but also most works on optical fiber communications have assumed source uniformity (i.e. equal probability of marks and spaces) so far. On the other hand, the source information is in general nonuniform, unless bit-scrambling or other source coding techniques to balance the bit probability is performed. Interestingly, one can exploit the source nonuniformity to reduce the entropy of the channel input symbols with the PS encoder, which leads to smaller required signal-to-noise ratio at a given input logic rate. This benefit is equivalent to a combination of data compression and PS, and thus we call this technique *compressed shaping*. In this work, we explain its theoretical background in detail, and verify the concept by both numerical simulation and a field programmable gate array (FPGA) implementation of such a system. In particular, we find that compressed shaping can reduce power consumption in forward error correction decoding by up to 90% in nonuniform source cases. The additional hardware resources required for compressed shaping are not significant compared with forward error correction coding, and an error insertion test is successfully demonstrated with the FPGA.**

Index Terms—**Coding, data compression, distribution matching, entropy, implementation, modulation, optical fiber communication, probabilistic shaping, source coding.**

I. INTRODUCTION

Traffic demands are growing with deployments of mobile communication systems for the 5th generation and beyond, which is further accelerated by the pandemic [1]. Optical fiber communications take a key role in the communication infrastructure because of its high capacity. In the past, the modulation formats used in optical fiber communications were binary, e.g., on–off keying, binary, or quaternary phase-shift keying without forward error correction (FEC) or with hard-decision FEC [2]. However, the latest 400 Gb/s standards [3], [4] utilize 16-ary quadrature amplitude modulation (QAM) with soft-decision (SD) FEC under bit-interleaved

coded modulation (BICM) [5]–[8]. Furthermore, constellation shaping [9], or more specifically, probabilistic shaping (PS) [10], [11], has attracted wide research interest due to its capacity-approaching performance [12]–[15]. Especially reverse concatenation, where the shaping encoding (also known as *distribution matching*, DM) [16]–[21], is done outside the FEC encoding [22]–[24], made PS deployable in terms of implementation capability.

An optimal encoder, which minimizes the rate loss in the conversion process, can be theoretically achieved in two steps if the block length is large enough; by applying source coding (often called "data compression") first and channel coding (i.e., constellation shaping and FEC coding) next. Information-theoretic coding and modulation techniques have realized significant performance improvements in recent years, thus almost closing the gap to the Shannon channel capacity [25]. In contrast, coding for dynamically variable source information [1] has rarely been investigated for fiber-optic communication systems, which aggregates massive user traffic in frames. In the standard [26], simple bit scrambling (flipping bits by the exclusive OR operation with a pseudorandom bit sequence (PRBS)) has been implemented to balance the mark (logic '1') and space (logic '0') counts instead of applying any source coding. Often we tend to assume the source bits as just uniformly distributed and independent, although the true source entropy before bit scrambling is variable and dependent on the user traffic, e.g., due to the existence of idle frames in the media access control (MAC) protocol [26].

Data compression and shaping are almost inverse operations, i.e., the former converts a nonuniform information sequence into a shorter uniform one, while the latter does the opposite. Simultaneous realization of data compression and shaping is not only an interesting research topic but also a key technique for more efficient communications in practice. Thus in this paper, we propose and investigate *compressed shaping*, which combines the benefits from data compression and shaping. Similar ideas have been studied in the context of joint source–channel coding in communication theory [27], [28], but an application to fiber-optic communication is presented here, for the first time to the best of our knowledge.

Compressed shaping is enabled by a shaping encoding that is sensitive to the source entropy. As data compression allocates short bit patterns to frequently occurring source words, compressed shaping allocates amplitudes with small energy to such frequent source words. This compression feature is similar to burst signalling in time-domain multiple access [29], i.e., optical power variation depending on the traffic. Com-

pressed shaping is a fixed-length to fixed-length conversion, and the average energy of channel input symbols is reduced for source information sequences having a small entropy. We do not need an operational mode change such as updating the source statistics based on prior knowledge, although this would be a kind of data compression. Our previously proposed look-up table (LUT)-based hierarchical DM [19] and following works [30], [31] can be applied for this purpose by reordering the LUT entries without significantly increased complexity. The proposed technique can reduce the rate losses associated with source and channel coding compared with state-of-the-art DM schemes such as constant-composition DM (CCDM) [16], or reduce the power consumption in the FEC at a given information rate and signal-to-noise ratio (SNR) by relaxing the FEC performance. We also report a field programmable gate array (FPGA) implementation, and the hardware demonstration results for compressed shaping 16- and 64-QAM at system throughputs of 153 and 113 Gb/s, respectively.

This is an evolutional work of [32]–[34], which are here extended by providing a more detailed theoretical background of compressed shaping. The system throughput is increased by separating the clock domain into data and controlling because the control circuitry was the bottleneck in the logical circuitry when making the clock frequency faster. The accuracy of the power consumption estimation is improved by introducing a dynamic simulation. Even if we consider shaping encoding only, there have been very few other reports on FPGA implementations [35], [36]. Furthermore, there are neither any reports on static/dynamic power consumption estimates nor any demonstrations by FPGA implementations including both shaping encoding and decoding, except [33].

The rest of the paper is organized as follows. The principle of compressed shaping is explained in Sec. II, and its numerical simulations are shown in Sec. III. An FPGA implementation example of compressed shaping is presented in Sec. IV, and the hardware-based demonstration results are summarized in Sec. V. Finally, Sec. VI concludes the paper.

II. COMPRESSED SHAPING—BASIC PRINCIPLES

While conventional PS systems employ full bit-scrambling and assume source uniformity, the proposed compressed shaping scrambles the sign bit only and applies source-sensitive amplitude shaping to realize better performance in the case of small source entropy. In this section, we firstly review the historical source uniformity assumption in conventional systems and discuss rate loss under either uniform or nonuniform source conditions. Then we compare compressed shaping with bit-sequence data compression and PS. Finally, we show the system model and characterize entropy bounds in compressed shaping systems.

A. Rate loss in conventional systems

Optical fiber communication systems gather many MAC frames from client traffic. When investigating coding and modulation techniques, source uniformity is usually assumed. It is because a 50% mark ratio is produced by line coding or bit scrambling, even if the true source information is nonuniform. This may be a reasonable model under the idealized assumption of a stable, uniform traffic flow. However, due to dynamically variable traffic volumes [1], there will be many idle frames. These are encoded by 64B/66B line coding into the zero codeword except for the sync header and block type field [26]. Due to the existence of such idle frames, the source bits $S_i \in \{0, 1\}$ can have more than 50% '0's, which makes the binary source entropy $\mathbb{H}(S_i) < 1$, where $\mathbb{H}(\cdot)$ denotes entropy. For example, when an 8-byte idle signal is encoded by 64B/66B, a 66-bit block consists of the 2-bit sync header 01, the 8-bit block type field 00011110, and eight 7-bit control bits 0000000 for the 8-byte idle signal, which results in a source mark ratio of $5/66 = 7.6\%$.

In conventional systems, the (serial) source bits S_i are parallelized to form a source bit sequence $[S_1 \ldots S_{k_\mathrm{bs}}]$, after which the bit scrambling converts $[S_1 \ldots S_{k_\mathrm{bs}}]$ into a scrambled bit sequence $[U_1 \ldots U_{k_\mathrm{bs}}]$. The sequence length k_bs is chosen to match the applied bit-scrambling protocol. An arbitrary (randomly selected) bit in this scrambled sequence has the distribution $P_U = (1/k_\mathrm{bs}) \sum_{i=1}^{k_\mathrm{bs}} P_{U_i}$, which is assumed to be uniform and $\mathbb{H}(U) = 1$, even when the corresponding (serial) unscrambled bit distribution $P_S = (1/k_\mathrm{bs}) \sum_{i=1}^{k_\mathrm{bs}} P_{S_i}$ yields an entropy $\mathbb{H}(S) < 1$. The bit scrambling is essential in binary modulation for maintaining direct current levels in electronic devices and for recovering the clock signal at the receiver. Non-PS QAM systems also utilize bit scrambling to maintain the 50% mark ratio. In general, neither S_i nor U_i ($i = 1, 2, \ldots, k_\mathrm{bs}$) is ensured to be identical and independent distributed (i.i.d.), so $\mathbb{H}([S_1 \ldots S_{k_\mathrm{bs}}]) \le k_\mathrm{bs}\mathbb{H}(S)$ and $\mathbb{H}([U_1 \ldots U_{k_\mathrm{bs}}]) \le k_\mathrm{bs}\mathbb{H}(U)$ by the concavity of entropy. However, $[U_1 \ldots U_{k_\mathrm{bs}}]$ is assumed to be i.i.d. and uniform in most research works on channel coding.

In PS, the goal is to reduce symbol entropy at a given information rate in order to reduce the required SNR for quasi-error-free operation over channels approximated by the Gaussian channel. The scrambled bit sequences $[U_1 \ldots U_{k_\mathrm{bs}}]$ are rearranged into sequences of length k, and each such length-k bit sequence is mapped into a sequence of amplitudes $[A_1 \ldots A_n]$. The sequence lengths k and n are selected to match the PS scheme, regardless of k_bs. Since U_i is in general not i.i.d., neither is A_i, and thus $\mathbb{H}([A_1 \ldots A_n]) \le n\mathbb{H}(A)$, where $P_A = (1/n) \sum_{i=1}^{n} P_{A_i}$. On the other hand, A_i is assumed to be i.i.d. in a mismatched (memoryless) receiver.

The PS performance is typically quantified with a rate loss

$$R_\mathrm{loss}(U, A) = \mathbb{H}(A) - \frac{k}{n}, \tag{1}$$

where $R_\mathrm{loss}(\cdot, \cdot)$ denotes a rate loss in a sequence conversion, and $R_\mathrm{loss}(U, A) \ge 0$ for i.i.d. U_i. The CCDM [16], which has been the state-of-the-art PS coding, has an almost negligible rate loss when n is sufficiently large, e.g., 1000–10000. The CCDM generates a fixed probability mass function (PMF) P_A of output amplitudes A regardless of the DM encoder input bit sequence and its statistics. For a fixed PMF P_A with a nonconstant-composition DM, U_i is required to be uniformly distributed in general, which can be achieved by bit scrambling.

B. Rate loss in source to amplitude conversion

The CCDM shows negligible rate loss $R_{\mathrm{loss}}(U, A)$ in (1) with a sufficiently large block length in PS for a uniform source, but not for nonuniform sources, in which case the channel input symbol entropy or average symbol energy can be further reduced by exploiting nonconstant-composition DM.

During a conversion process from source bit sequence to amplitude symbol sequence, bit scrambling acts against the symbol entropy reduction because it maximizes binary entropy ($\mathbb{H}(U) = 1$) even if $\mathbb{H}(S) < 1$. Instead, we propose to exploit this source nonuniformity. In the case of $\mathbb{H}(S) < 1$, the rate loss in the conversion from a nonuniform source bit sequence $[S_1 \ldots S_k]$ into an amplitude sequence $[A_1 \ldots A_n]$ can in general be expressed as

$$R_{\mathrm{loss}}([S_1 \ldots S_k], A) = \mathbb{H}(A) - \frac{\mathbb{H}([S_1 \ldots S_k])}{n}, \quad (2)$$

which is bounded as

$$R_{\mathrm{loss}}([S_1 \ldots S_k], A) \geq R_{\mathrm{loss}}(S, A) \quad (3)$$

$$R_{\mathrm{loss}}(S, A) = \mathbb{H}(A) - \mathbb{H}(S)\frac{k}{n}. \quad (4)$$

Obviously, CCDM is not optimum (with or without bit scrambling) because of the constant P_A and $\mathbb{H}(A)$ regardless of the source distribution. The general rate loss bound $R_{\mathrm{loss}}(S, A)$ in (4) becomes significantly larger than $R_{\mathrm{loss}}(U, A)$ in (1) under a small $\mathbb{H}(S)$. A nonconstant-composition DM could be source sensitive, i.e., realizing a smaller output entropy $\mathbb{H}(A)$ for a smaller input entropy $\mathbb{H}(S)$, resulting in a smaller $R_{\mathrm{loss}}(S, A)$ compared with CCDM. Even in such cases, at least the sign bits should be bit-scrambled for the direct current level management and clock recovery. To enhance the performance under nonuniform source information, data compression is another option, at the expense of, possibly large, digital signal processing circuit resources, since it has to adapt to the time variation of $\mathbb{H}(S)$.

C. Proposed bit sequence conversion

This section explains the principle of the proposed compressed shaping compared to well-known data compression and PS. Tab. I shows small examples of bit sequence conversions using (a) data compression, (b) PS, and (c) compressed shaping.

Tab. I(a) assumes a nonuniform source input bit sequence $S = [S_1 S_2]$ with a fixed length $k = 2$. The bit sequence conversion is given by Huffman coding [37], which allocates a short output string to a high probability input word. The output bit sequence C have variable lengths n_C from 1 to 3 in this example. Then the fixed input length $k = 2$ is shortened to $n_{\mathrm{avg}} = \mathbb{E}[n_C] = 1.7$ on average, where $\mathbb{E}$ denotes expectation. Such conversion is useful to reduce the required storage size after conversion. Huffman coding is an invertible data compression; however, it is usually not suitable for high throughput data communications due to issues of latency and required storage size in the variable-length conversion process.

Tab. I(b) assumes a uniform source input bit sequence $U = [U_1 U_2]$ with a fixed length $k = 2$, which is converted into an amplitude sequence $A = [A_1 A_2 A_3]$ with a fixed length

$n = 3$ and an amplitude element $A \in \{1, 3\}$. Candidates of output amplitudes are sorted in ascending order of average energy per codeword $E_A = \|A\|_2^2/n$. The output amplitude sequence $A = 111$ has the smallest E_A of 1, and $A = 113$, 131, and 311 have the second smallest E_A of 11/3. Amplitude sequences $A = 133$, 313, 331, and 333 are not chosen as output strings in this codebook because of their large E_A. Then the average output energy $E_{\mathrm{avg}} = \mathbb{E}[E_A] = 3$. When we use a codebook with uniform output amplitudes of 1 and 3 with a length of 2 (i.e., 11, 13, 31, and 33), $E_{11} = 1$, $E_{13} = E_{31} = 5$, $E_{33} = 9$, and $E_{\mathrm{avg}} = 5$. Compared with such a uniform output amplitude case, E_{avg} is reduced by $2.2\,\mathrm{dB}$ with this exemplified PS, leading to a small required SNR at a given information rate over the Gaussian channel. To shape the output amplitude probabilities (i.e., to reduce the output amplitude entropies), we thus need more output bit slots than input bit slots.

Tab. I(c) exemplifies the proposed compressed shaping, which assumes a nonuniform source input bit sequence $S = [S_1 S_2]$ as in the case of Tab. I(a). The source bit sequence S with a fixed length $k = 2$ is converted into an amplitude sequence $A = [A_1 A_2 A_3]$ with a fixed length $n = 3$. When generating the codebook, we need two sortings: (i) the input bit sequences S are sorted in descending order of probability P_S, and (ii) the output amplitude sequence A is sorted in ascending order of average energy E_A. By allocating output amplitude sequences with a small energy to input bit sequences with a high probability, the average output energy E_{avg} becomes $7/3$, which is $1.1\,\mathrm{dB}$ further less than the one in Tab. I(b). Both conventional PS and compressed shaping are fixed-length to fixed-length conversions in these examples, and are therefore suitable for high-throughput data communication which arranges data into fixed-length frames. We do not need any adaptation of the codebook to different source statistics.

D. Proposed system model

Fig. 1 shows the system model for the proposed compressed shaping. Here we exemplify using hierarchical DM [19] for the shaping encoder/decoder, but generally any nonconstant-composition DM can be used. The source information bits are parallelized into a source bit sequence $S = [S_1 \ldots S_{k_{\mathrm{bs}}+k}]$, which is separated into a source sign bit sequence $S_{\mathrm{s}} = [S_{\mathrm{s},1} \ldots S_{\mathrm{s},k_{\mathrm{bs}}}]$ and a source amplitude bit sequence $S_{\mathrm{a}} = [S_{\mathrm{a},1} \ldots S_{\mathrm{a},k}]$. The source sign bit sequence S_{s} is bit-scrambled into $U_{\mathrm{s}} = [U_{\mathrm{s},1} \ldots U_{\mathrm{s},k_{\mathrm{bs}}}]$ by taking the exclusive OR with a PRBS, to balance the numbers of '0's and '1's. The source amplitude bit sequence S_{a} is processed by a bit-flipping function, which flips all input bits if there are more '1's than '0's and adds a parity bit '1', otherwise just adds a parity bit '0', because a large mark ratio is not desirable in the compressed shaping scheme.[1] Then the bit-flipping encoding output bit sequence $F = [F_1 \ldots F_{k+1}]$ contains at least as many '0's as '1's. The sequence F is then processed in a hierarchical DM, which consists of hierarchically connected

---

[1]There may be situations with a predominance of '0's (due to many idle frames) or '1's (due to an alarm indication signal) [26]. In both cases, the source entropy is small.

TABLE I
BIT SEQUENCE CONVERSIONS.

(a) Data compression

Input		Output		
Bits S	Probability P_S	Bits C	Probability P_C	Length n_C
00	0.50	1	0.50	1
01	0.30	01	0.30	2
10	0.15	001	0.15	3
11	0.05	000	0.05	3
Average				1.7

(b) PS

Input		Output		
Bits U	Probability P_U	Amplitudes A	Probability P_A	Avg. energy E_A
00	0.25	111	0.25	1
01	0.25	113	0.25	3.67
10	0.25	131	0.25	3.67
11	0.25	311	0.25	3.67
Average				3

(c) Compressed shaping

Input		Output		
Bits S	Probability P_S	Amplitudes A	Probability P_A	Avg. energy E_A
00	0.50	111	0.50	1
01	0.30	113	0.30	3.67
10	0.15	131	0.15	3.67
11	0.05	311	0.05	3.67
Average				2.33

small look-up tables (LUTs) as shown in Fig. 2 [19], [39]. There are L layers and T_ℓ LUTs in each layer ℓ. Each LUT in a layer ℓ receives s_ℓ bits from the input interface of the DM and r_ℓ bits from layer $\ell + 1$, and it transmits $r_{\ell-1}$ bits to each of $t_{\ell-1}$ LUTs in layer $\ell - 1$, in total $u_\ell = t_{\ell-1}r_{\ell-1}$ transmitted bits. To determine the one-to-one correspondence of input and output words in each small LUT, we sort the input words in descending order of the number of '0's and the output amplitudes in ascending order of average energy, as in the small example shown in Tab. I(c). The output amplitude sequence from the hierarchical DM is $A = [A_{c,1} \dots A_{c,n}]$, where each element is a two-dimensional vector $A_{c,i} \in \{1, 3, \dots, 2^{m_a/2-1}\}^2$, m_a is the number of bit tributaries for a two-dimensional amplitude, and $P_{A_c} = (1/n)\sum_{i=1}^{n} P_{A_{c,i}}$. The amplitude sequence A is represented by a bit sequence $B_a = [B_{a,1} \dots B_{a,nm_a}]$. From B_a and U_s, an FEC parity bit sequence B_{fp} is generated by a systematic FEC encoder with an FEC code rate R_c. Then U_s and B_{fp} are concatenated into a sign bit sequence $B_s = [B_{s,1} \dots B_{s,nm_s}]$, where m_s denotes the number of sign-bit tributaries. The number of elements in S_s or U_s is $k_{bs} = n(m_s R_c - m_a(1 - R_c))$ and that in B_{fp} is $(1 - R_c)n(m_s + m_a)$. Finally channel input QAM symbols $X = [X_{c,1} \dots X_{c,n}]$ are generated from B_s and B_a, where $P_{X_c} = (1/n)\sum_{i=1}^{n} P_{X_{c,i}}$.

The channel output QAM symbols $Y = [Y_{c,1} \dots Y_{c,n}]$ are demapped into an L-value sequence $L = [L_1 \dots L_{(m_s+m_a)n}]$ by memoryless bit-metric decoding [12], where $Y_{c,i} \in \mathbb{R}^2$ and $\mathbb{R}$ denotes the real number set. The receiver-side processing reverses the one on the transmitter side. QAM symbol demapping and FEC decoding are performed to recover the scrambled source sign bit sequence $\hat{U}_s$ and the shaped ampli-

tude bit sequence $\hat{B}_a$. The recovered scrambled source sign bit sequence $\hat{U}_s$ is bit-descrambled into the source sign bit sequence $\hat{S}_s$. The shaping decoder, which is a hierarchical DM decoder here, converts $\hat{B}_a$ into the flipped bit sequence $\hat{F}$. The bit-flipping is terminated based on the parity bit, i.e., if the parity bit is '1', all input bits are flipped and the parity bit is removed, otherwise the parity bit is just removed, to obtain the source amplitude bit sequence $\hat{S}_a$. Finally, the source sign bit sequence $\hat{S}_s$ and the source amplitude bit sequence $\hat{S}_a$ are concatenated into the source information bit sequence $\hat{S}$.

We here summarize the entropy and rate loss in this compressed shaping system. The entropy of the channel input symbol is given by

$$\mathbb{H}(X_c) = m_s + \mathbb{H}(A_c). \tag{5}$$

If $S_1, \dots, S_{k_{bs}+k}$ are i.i.d. with a distribution $P_S = P_{S_a} = \frac{1}{k_{bs}+k}\sum_{i=1}^{k_{bs}+k} P_{S_i}$, then the rate loss in the sequence conversion from S_a to A_c is, similarly to (4),

$$R_{loss}(S_a, A_c) = \mathbb{H}(A_c) - \mathbb{H}(S)\frac{k}{n}. \tag{6}$$

In order to characterize the performance in Sec. III, we here define the minimum entropy of a channel input symbol as

$$\mathbb{H}_{LB} = m_s + \mathbb{H}(S)\frac{k}{n}. \tag{7}$$

Note that the essential function of compressed shaping is the shaping encoder/decoder based on a nonconstant-composition DM. A small example of this is shown in Tab. I(c). Once the shaping encoder/decoder functions are determined, they will not be adapted even if the source statistics would vary dynamically. This feature helps to ensure a low-complexity compressed shaping. The other additional functions in Fig. 1 are optional. For an implementation with the standardized framing using 64B/66B encoding and the alarm indication signal, the bit-flip encoding/decoding is required. To support bipolar signaling such as QAM for coherent systems, the bit scrambling/descrambling of the sign bit is required to maintain the DC level at zero. With some adjustments, we can apply compressed shaping to 400ZR [3] and openROADM [4]. These employ 256B/257B transcoding after the 64B/66B encoding. In the transcoding, most bits are transparently output, while several bits are shortened or changed. Thus the source nonuniformity from the 66-bit blocks are retained. Before the symbol mapping, the sign and amplitude bits must be properly distributed, and the bit scrambling should be modified to avoid breaking the source nonuniformity. These requirements are reasonably minor in a practical implementation.

III. SIMULATIONS

To verify the concept of compressed shaping, we performed numerical simulations based on the system model in Sec. II-D. In this section, nonuniform, independent source information bits S were generated for simplicity, by independent, uniformly distributed pseudorandom numbers from the Mersenne twister. For a given target source mark ratio $P_S(1)$, the uniformly distributed pseudorandom numbers ranging from 0 to 1 were binarized with a threshold level $1 - P_S(1)$, i.e.,
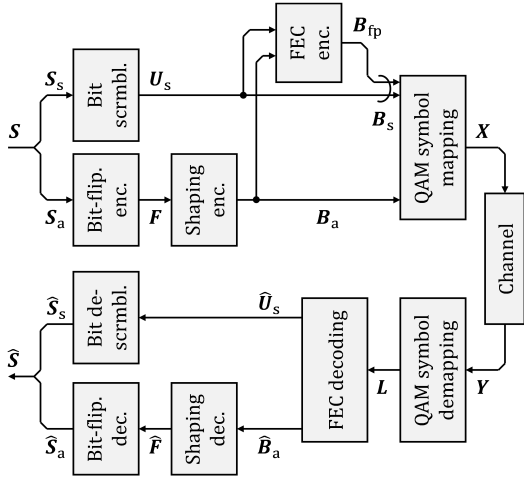
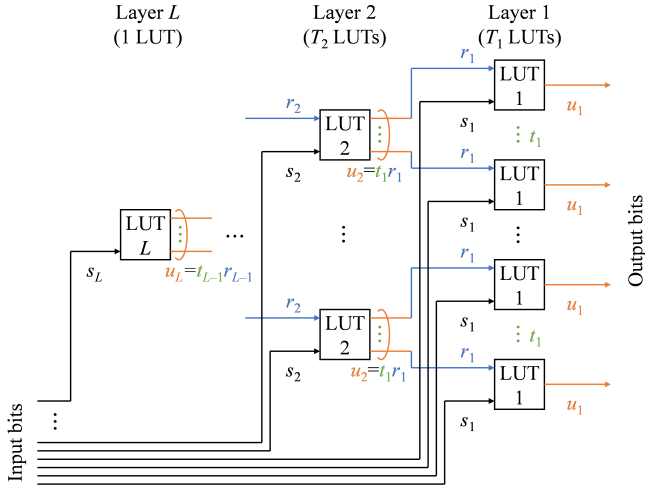Fig. 1. System model for the proposed compressed shaping.

Fig. 2. Schematic of hierarchical DM encoding [39, Fig. 2].

generating a logic '0' for a random number 0 to $1 - P_S(1)$ and a logic '1' otherwise. In this simulation, we set a static source mark ratio $P_S(1)$ in a simulation batch for a short period.

Fig. 3 shows the average two-dimensional symbol energy E_{2d} as a function of the lower bound entropy of channel input symbols $\mathbb{H}_{LB}$ in (7) for various PS-QAM formats and source mark ratios $P_S(1) = 0.5, 0.4, 0.3, 0.2, 0.1$, and 0.05, where the minimum Euclidean distance $d_{LB} = 2$. For compressed shaping with hierarchical DM, we employed 8-, 16-, 32-, 64-, and 128-QAM as base constellations. The PS overhead was around 7% in each case when assuming the use of a rate-5/6 FEC, and the PS codeword length (number of QAM symbols) was 256, 256, 192, 128, and 128 for 8-, 16-, 32-, 64-, and 128-QAM, respectively. The PS codeword length equals the block length of compressed shaping. The PS encoder input block consists of several 100 information bits, and the source mark ratio is averaged over each block. This number of several 100 bits is small enough to avoid over-averaging of dynamic variations of the source mark ratio due to idle MAC frames. Such granular base constellation and shallow shaping help to avoid excessive increases of peak-to-average power ratio and power consumption [38], [39] and
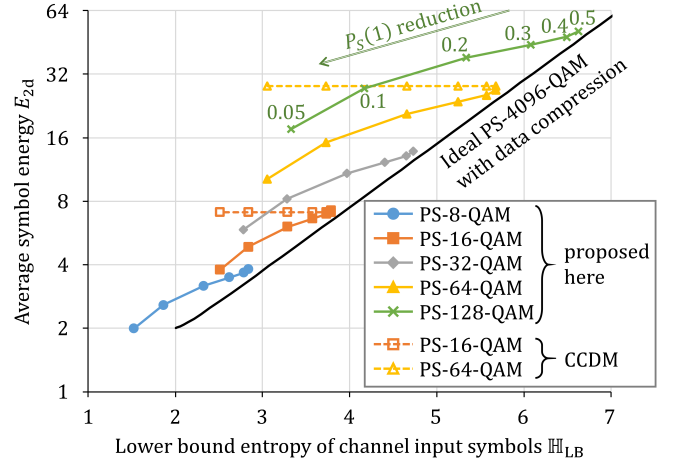
Fig. 3. Average two-dimensional symbol energy E_{2d} for various PS-QAM formats and source mark ratios $P_S(1)$ in the range 0.5–0.05 as a function of the lower bound entropy of channel input symbols $\mathbb{H}_{LB} = m_s + \mathbb{H}(S)k/n$.

penalties from nonideal FEC performance [40]. As the 8-QAM constellation, $\mathcal{C}_3$ in [41] was used to make the constellation symmetric around the imaginary axis,[2] so that the uniformly distributed FEC parity bits could be placed on the sign bits without changing $P_{|X_c|}$. The bit labelling for 128-QAM was based on [42, Fig. 3]. For comparison, CCDM-based PS-16-QAM and PS-64-QAM were also evaluated, using the same PS overhead and codeword length as with compressed shaping. We also evaluated the performance of PS-4096-QAM with an ideal Maxwell–Boltzmann input distribution and perfect data compression.

With CCDM, the energy E_{2d} in Fig. 3 is constant for various $P_S(1)$ cases, because its output PMF P_{A_c} does not depend on incoming bits. With compressed shaping, on the other hand, E_{2d} decreases with decreasing $P_S(1)$ (and $\mathbb{H}(S)$) for each base constellation, although there are significant performance gaps to the ideal case (black solid line in Fig. 3), especially for high-order QAM. This gap comes from the fixed compressed shaping regardless of the source statistics. While an adaptive compressed shaping would bring the performance closer to the ideal case, it would lead to a high complexity and is out of scope of this work. Fig. 4 exemplifies the PMF P_A for compressed shaping 64-QAM with source mark ratios $P_S(1) = 0.5$–0.05. Higher source nonuniformity (i.e., smaller $P_S(1)$) makes deeper PS. Under such source nonuniformity, we observed reduced $P_A(3)$, $P_A(5)$, and $P_A(7)$ and increased $P_A(1)$ compared with the uniform source case. The transmitted PMFs of one-dimensional amplitudes A and two-dimensional channel input symbols X_c at a given source mark ratio $P_S(1)$ are denoted as $P_{A(P_S(1))}$ and $P_{X_c(P_S(1))}$, respectively.

We then simulated the required SNR over the Gaussian channel with the DVB-S2 low-density parity check code [43] having a code rate of $5/6$, with a maximum number of decoding iterations of 50. Fig. 5 shows the simulated required SNR with the maximum number of FEC decoding iterations. The required SNR is defined at an asymmetric information

---

[2]The signal points are $(\pm 1, 0)$, $(\pm 1, \pm 2)$, and $(\pm 3, 0)$.
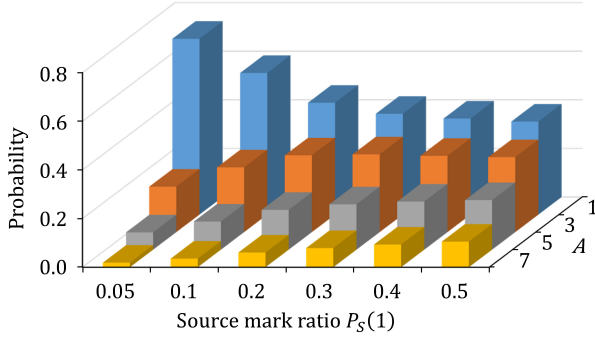
Fig. 4. An example of a one-dimensional amplitude PMF P_A by compressed shaping 64-QAM, for various source mark ratios $P_S(1)$. In situations with dynamically variable $P_S(1)$, the demapper assumes a constant $P_S(1) = 0.5$, as explained in the text.

[44], [45][3] of 0.87, which corresponds to a pre-FEC BER of about $3.4 \cdot 10^{-2}$. We observed no residual bit errors in more than $2.5 \cdot 10^6$ information bits after FEC decoding in the SNR regime larger than the required SNR for each case. We simulated both *mismatched* and *matched* decoding. The true transmitted two-dimensional symbol PMF is denoted by $P_{X_c(P_S(1))}$ and the transmitted symbol PMF assumed in the soft demapping by $Q_{X_c(P_S(1))}$. In the matched case, $Q_{X_c(P_S(1))}$ was set to $P_{X_c(P_S(1))}$ for all $P_S(1)$, while in the mismatched case, $Q_{X_c(P_S(1))}$ was set to a fixed PMF of $P_{X_c(0.5)}$ regardless of the true $P_S(1)$. This is because under dynamically variable source situations, the true transmitted symbol PMF $P_{X_c(P_S(1))}$ is hard to track in deployable systems. As shown in Fig. 5, the required SNR can be reduced by compressed shaping, in contrast to the fixed required SNR by CCDM and uniform QAM with bit scrambling due to the fixed P_{X_c} ($= P_{X_c(0.5)}$). The SNR penalty by the mismatch between $P_{X_c(P_S(1))}$ and $Q_{X_c(P_S(1))}$ is not significant except at very small $P_S(1)$.

The better performance of compressed shaping compared with conventional PS can be converted into lower power consumption. We quantified the power consumption in FEC decoding, because it dominates the power consumption among all coding functions. Fig. 6 shows the relative power consumption of the FEC decoding for PS-QAM with $P_S(1) = 0.5$ (circle), 0.4 (square), 0.3 (diamond), 0.2 (triangle), 0.1 (cross), or 0.05 (plus), so there are six curves for each QAM order with compressed shaping. The power consumption is assumed to be proportional to the average number of decoding iterations, which is almost proportional to the toggle rate in logical circuitry. The vertical axis in Fig. 6 is normalized by the maximum number of decoding iterations, i.e., 50. The soft demapping was assumed to be mismatched as in Fig. 5, i.e., $Q_{X_c(P_S(1))} = P_{X_c(0.5)}$. While CCDM consumes a fixed power even if $P_S(1)$ is reduced (this is the same for non-PS signaling with bit scrambling, but not shown in Fig. 6), compressed shaping significantly reduces the power to about 10% for highly nonuniform source probabilities, because of the smaller

[3]The asymmetric information is equivalent to normalized generalized mutual information [15], [46] and achievable FEC rate [14] under matched decoding.
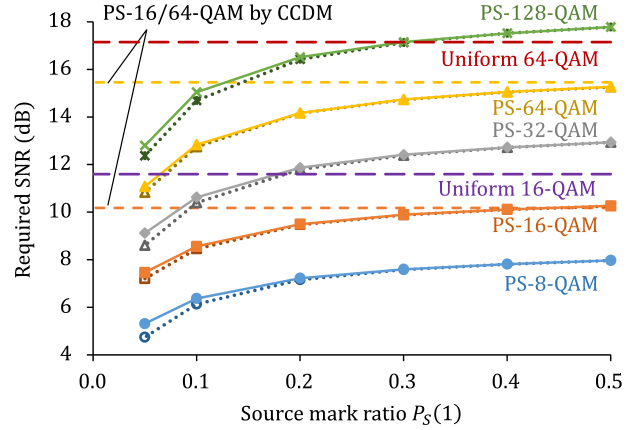
Fig. 5. Simulated required SNR for compressed shaping QAM under matched (dotted lines) and mismatched decoding (solid lines). In the mismatched case, we set the transmitted PMF assumed in the demapper to the one for a uniform source ($P_S(1) = 0.5$), for every source mark ratio $P_S(1)$. Dashed and long-dashed lines correspond to CCDM and uniform QAM, respectively.
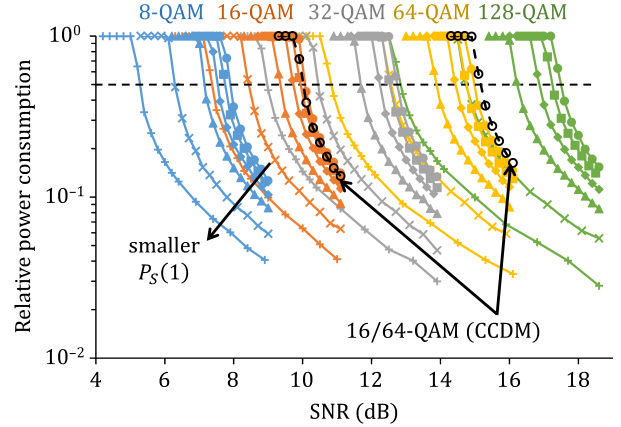
Fig. 6. Simulated relative power consumption of the FEC decoding, which is assumed to be proportional to average number of decoding iterations, as a function of SNR for PS-QAM. Solid and black dashed lines correspond to compressed shaping and CCDM, respectively.

required SNR, which leads to a smaller number of decoding iterations than with conventional bit-scrambled PS.

While in this work, the source mark ratio is assumed to be constant, dynamic variations thereof were simulated in [34, Sec. 5.4]. Source mark ratios were linearly varied, e.g., from 20% to 40% or from 10% to 50%, and compared with a constant 30% as a benchmark. For a larger range of variation, the performance improved slightly, even if the average source mark ratio over many blocks was the same. Note that additional complexity in compressed shaping is not significant when hierarchical DM is employed for shaping encoding/decoding, which will be shown in the next section.

IV. FPGA IMPLEMENTATION

We implemented compressed shaping in a single FPGA chip on an evaluation board Xilinx® Virtex® Ultrascale+™ VCU118 XCVU9P. Fig. 7 shows the functional block diagram of the implemented circuitry. The source generator outputs source information bits based on a given target mark ratio $P_S(1)$. The schematic of the source generator is illustrated in Fig. 8. Each source information bit is selected from one of

Fig. 7. Block diagram of FPGA implementation of compressed shaping.

Fig. 8. Schematic of source generation function. The mask signal selects the output source bit from logic '0', '1', or a bit given by the PRBS based on the target source mark ratio $P_S(1)$.

three possible candidates; 0) the logic bit '0', 1) a PRBS of length $2^{31} - 1$ bits, and 2) the logic bit '1'. The mask signal, used for the selection, is generated from the target source mark ratio $P_S(1)$, which is given by the user. If $P_S(1) \leq 0.5$, the mask signal takes on values 0 or 1 such that the average fraction of logic '1's in S is $P_S(1)$, and if $P_S(1) > 0.5$, the mask similarly is 1 or 2. Fig. 9 shows an exemplified mask signal when the target $P_S(1)$ is 0.4. For simplicity, we classified source bits into 20 groups (32 bits per group), provided a '0' mask window for four groups, and slided the window in every clock cycle.

The bit-flipping encoder counts the numbers of '0's and '1's. When the number of '1's is larger than that of '0's, it flips all bits at the clock cycle and adds a parity bit '1'. Otherwise it just adds a parity bit '0'. The shaping encoding/decoding is realized by hierarchical DM having a total codeword length of 404 bits for the shaped two-dimensional amplitudes. The number of shaped information bits per two-dimensional amplitude k/n for compressed shaping employing hierarchical DM is generally flexible, and is in this implementation fixed to $372/202$ and $372/101$ for 16-QAM and 64-QAM, respectively, with compressed shaping.[4] Before the receiver-side processing, we have an error insertion function, which inserts bit errors based on a given bit error rate (BER) before the shaping decoder. The shaping decoder is also implemented with a hierarchical DM decoder, and the bit-flipping decoder recovers the source bits. When the parity bit is '1', it flips all bits at the clock cycle and removes the parity bit. Otherwise, it just removes the parity bit.

There are several monitoring functions, i.e., the PMF P_{A_c} at the shaping encoder output, the assumed post-FEC BER, and the system output BER. Note that hierarchical DM mainly consists of LUTs, which are implemented with random access memory (RAM). Because it is sensitive to unwanted bit-flipping due to radiation-induced soft errors, we implemented soft error protection circuitry.

The clock domain was initially single and the clock frequency f_{clk} for the fitting (FPGA synthesis) was 90 MHz [33]. Later we found a bottleneck in making the clock frequency higher inside the soft error protection circuitry (consisting of flip-flops and a selector tree for refreshing RAM contents), so we separated the clock domain into one for data processing and one for control. We then achieved $f_{\mathrm{clk}} = 240$ MHz for

---

[4]A flexible choice of (n, k) in an FPGA implementation of hierarchical DM is left as potential future work. The LUT contents can be reconfigured in software or firmware without increasing the RAM size.

		Clock cycle index																			
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	19	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
	18	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	17	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	16	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	15	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	14	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Group index of source information bits	13	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	12	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
	11	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
	10	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	9	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1
	8	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
	7	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1
	6	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1
	4	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

Fig. 9. Example of mask signals for given clock cycle indices and group indices for a target source mark ratio $P_S(1) = 0.4$. In each clock cycle index or each group index, 20% of the generated bits are set directly to '0' and 80% are taken from a uniform PRBS, which implies $20 + 80/2 = 60\%$ '0's on average.

the data processing for higher throughput. Assuming a suitable FEC concatenation (not implemented here), the system throughputs for compressed shaping 16- and 64-QAM would be 57 and 42 Gb/s at $f_{\mathrm{clk}} = 90$ MHz, and 153 and 113 Gb/s at $f_{\mathrm{clk}} = 240$ MHz, respectively. The number of bits per PS codeword were the same in both cases, i.e., there were half as many PS-64-QAM symbols as PS-16-QAM symbols.

Tab. II shows the utilized hardware resources at a clock frequency of the data signals $f_{\mathrm{clk}} = 90$ or 240 MHz. Fig. 10 depicts the utilized area of the FPGA chip having three dies. The used resource elements were mainly located in the left and right dies, and the center die was mainly used for connection between the two dies. The *register* elements were mainly used by the soft error protection circuitry for storing the RAM contents. Out of the about 290,000 utilized *LUT as logic* elements, 70% were provided to external functions such as source generator and BER/PMF monitors, 10% were for bit-flipping encoding, and the rest was for other combinational logics. The data processing of the hierarchical DM used totally 3.5 Mb *block RAM* elements. No *ultra RAM* or *DSP slice* elements were used.

As a benchmark, 400 ZR FEC was implemented in 50 Xilinx® Virtex® Ultrascale™ FPGAs [47]. The clock frequency was 125 MHz and the system throughput was 200 Gb/s with external functions, i.e., source generation, mapping, demapping, interleaver, de-interleaver, and noise loading. Even considering that this reference includes many external

TABLE II
UTILIZATION OF KEY RESOURCES IN FPGA FOR COMPRESSED SHAPING
AT $f_{\mathrm{clk}} = 90$ OR $240\,\mathrm{MHz}$.

Category	Element	Available	Utilization	
			90 MHz	240 MHz
System logic	LUT as logic	1182k	24.17%	24.64%
cell	Register	2364k	22.49%	21.38%
Memory	Block RAM	75.9 Mb	4.65%	4.65%
	Ultra RAM	270.0 Mb	Not used	Not used
DSP slice		6840 slices	Not used	Not used
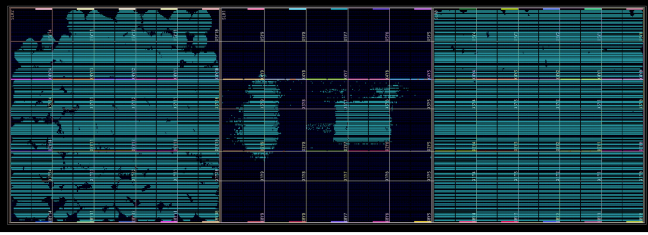
Fig. 10.   Utilized area (green) of a single FPGA chip for compressed shaping.

functions, compressed shaping utilizes a very small amount of hardware resources.

Tab. III shows the estimated dynamic power consumption for compressed shaping 64-QAM at $f_{\mathrm{clk}} = 240\,\mathrm{MHz}$. While we estimated a static power consumption based on a default toggle rate of 12.5% in our previous report [33], we now improved the estimation accuracy by taking realistic node switching activities into account based on register transfer level simulation waveform (a so-called switching activity interchange format) over 4000 clock cycles, where the source mark ratio $P_S(1)$ was set to 0.5.

The blocks of Tx data and Rx data in Tab. III are essential for data communications. Among them, the preprocessing (including bit-flipping encoding) and DM encoder core in the Tx data block, and the DM decoder core in the Rx data block, consumed most of the power. The DM encoder and decoder cores mainly consisted of block RAMs for LUTs. The bit-flipping encoding counted the logic '1's using adders, leading to a relatively large power consumption. Other pre- and postprocessing functions including the lane reorder consumed little power. Note that the power consumption of either the Tx or the Rx data block was smaller than that of external functions, e.g., source generation and BER monitoring. We had nonnegligible power consumption in the control blocks. They had soft error protection functions for the DM encoder and decoder cores by holding copies of the entire RAM contents in registers and refreshing the RAM intermittently. These powers can be expected to be less in an ASIC implementation by making the activation ratio small.

V. HARDWARE DEMONSTRATION

We demonstrated compressed shaping by employing the FPGA evaluation board at $f_{\mathrm{clk}} = 90\,\mathrm{MHz}$. First, the transmitter-side functions were verified. Histograms of the two-dimensional amplitude A_{c} for compressed shaping 16- and 64-QAM were measured over 10^{10} two-dimensional amplitude samples for various source mark ratios $P_S(1)$. The obtained histograms were interpreted as PMFs $P_{A_{\mathrm{c}}}$ and their

TABLE III
SIMULATED DYNAMIC POWER CONSUMPTION OF COMPRESSED SHAPING
64-QAM AT $f_{\mathrm{clk}} = 240\,\mathrm{MHz}$.

| Block	Function	Power (mW)
External	Source generation	589
functions	Error insertion	11
	BER monitor	638
	Clock generation	99
	Sub-total	1 337
Tx data	Pre-processing	232
	DM encoder core	456
	Post-processing	78
	Delay adjustment	46
	Sub-total	812
Rx data	Pre-processing	44
	DM decoder core	562
	Post-processing	34
	Delay adjustment	35
	Sub-total	675
Tx control		320
Rx control		295
Tx/Rx control		167
Total		3 613
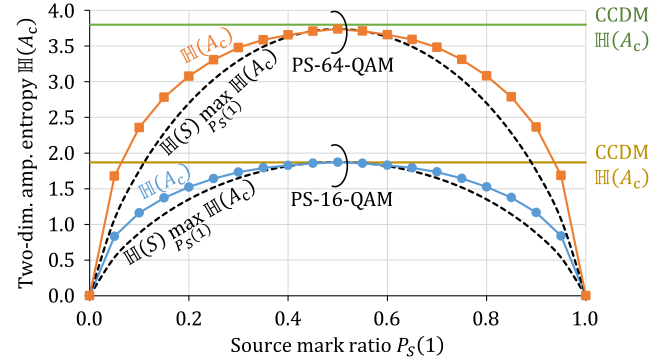
Fig. 11.   Measured two-dimensional amplitude entropy $\mathbb{H}(A_{\mathrm{c}})$ for compressed shaping as a function of the target source mark ratio $P_S(1)$ based on $> 10^{10}$ two-dimensional amplitude samples. The dashed lines show the binary entropy function scaled to the maximum entropies of the two-dimensional amplitudes for comparison.

entropies were computed. Fig. 11 shows $\mathbb{H}(A_{\mathrm{c}})$ as a function of $P_S(1)$. The entropy $\mathbb{H}(A_{\mathrm{c}})$ is maximum for $P_S(1) = 0.5$. The two-dimensional PS rate losses $R_{\mathrm{loss}}(U, A_{\mathrm{c}})$ are 0.034 and $0.064\,\mathrm{bpcu}$ for the exemplified PS-16-QAM and PS-64-QAM schemes, respectively. When $P_S(1)$ deviates from 0.5, i.e., $\mathbb{H}(S)$ decreases, $\mathbb{H}(A_{\mathrm{c}})$ also decreases monotonically. Because of the bit-flipping encoding, $\mathbb{H}(A_{\mathrm{c}})$ is symmetric around $P_S(1) = 0.5$. In contrast, the CCDM-based schemes show a constant $\mathbb{H}(A_{\mathrm{c}})$, which is independent of the source distribution. Scaled binary entropy functions $\mathbb{H}(S) \max_{P_S(1)} \mathbb{H}(A_{\mathrm{c}})$ are also depicted in Fig. 11 with black dotted lines for the two cases. The gap between $\mathbb{H}(A_{\mathrm{c}})$ and $\mathbb{H}(S) \max_{P_S(1)} \mathbb{H}(A_{\mathrm{c}})$ is a data compression rate loss. The data compression rate loss is ideally zero but is larger here for smaller $\mathbb{H}(S)$, due to the nonideal simple processing of compressed shaping. Here the bit sequence conversion loss in compressed shaping $R_{\mathrm{loss}}(S_{\mathrm{a}}, A_{\mathrm{c}})$ in (6) is given by the sum of rate losses in PS and data compression. Regardless of the rate loss increase in data compression, $\mathbb{H}(A_{\mathrm{c}})$ itself is reduced to help reduce the required SNR substantially.

Next, the receiver-side functions were verified. We turned
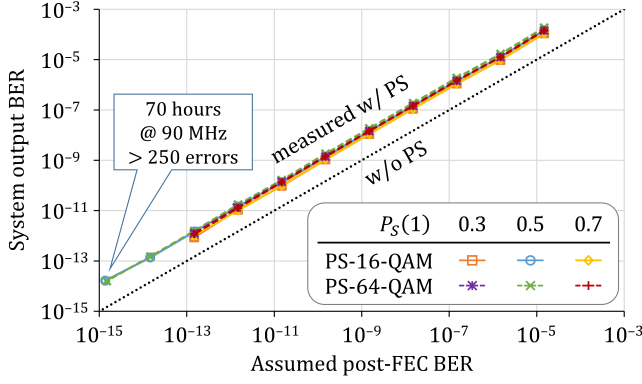
Fig. 12. Measured system output BER for compressed shaping as a function of assumed post-FEC BER in back-to-back error insertion tests.

the error insertion function on to provide sparse bit errors between the transmitter and the receiver. As reported in [19], sparse errors creates the worst system output BER at a given post-FEC BER. We swept the assumed post-FEC BER from 10^{-5} to 10^{-15} and examined $P_S(1)$ of 0.3, 0.5, and 0.7 for compressed shaping 16- and 64-QAM. After the receiver-side processing, the system output BER was measured. Fig. 12 shows the system output BER as a function of assumed post-FEC BER in the back-to-back error insertion test. The BER increase due to compressed shaping decoding was only around 10 times because hierarchical DM can partially decode correctly even if there are incoming bit errors. As predicted in [19], the BER increase factor is significantly smaller than for other DM techniques. For example, CCDM decoding having a PS codeword length of around 400 bits results in more than 100 times higher BER, though we could not implement CCDM in the FPGA due to its high complexity. We performed 70 hours of long-term measurement only in the case of $P_S(1) = 0.5$, due to time constraints. The number of observed bit errors after compressed shaping decoding were more than 250. The system output BER was $1.5 \cdot 10^{-14}$ at a post-FEC BER of $1.6 \cdot 10^{-15}$. This proves that the proposed compressed shaping does not cause an error floor or excessive error increase, so the required post-FEC BER remains around 10^{-16} to satisfy a required system output BER of 10^{-15}.

VI. SUMMARY

We proposed and demonstrated compressed shaping, i.e., the application of hierarchical DM to simultaneous source data compression and probabilistic shaping, which is an example of joint source–channel coding. Under a reduced source entropy, compressed shaping reduces channel input symbol entropy, symbol energy, and required SNR. Simulation results showed its smaller required SNR, as well as reduced power consumption compared with CCDM. We implemented compressed shaping, which are mainly hierarchical DM encoding/decoding, into a single FPGA and estimated the dynamic power consumption based on simulated waveforms. The system throughput reached 153 and 113 Gb/s for compressed shaping 16- and 64-QAM, respectively. Real-time evaluation results showed expected performance in both encoding and decoding. Compressed shaping works at a very small BER

of around 10^{-15} without any error floor, and its decoding increases the BER only around 10 times, which is small compared to other DM schemes.

The time-varying symbol energy in compressed shaping causes practical issues with respect to electrical amplitude, optical power, and SNR control in fiber-optic communication systems, although such a variation per channel can be statistically relaxed by multiplexing many channels. One example can be found in a wavelength-division-multiplexed system operated at a given total launch power. Without any control of the average symbol energy per channel, the average optical power in some channels can temporarily be reduced when the source entropy becomes small. This leads to a larger average optical power and thus higher nonlinearity impairments in other copropagating channels. Analyses of such issues and development of appropriate control methods are deferred to a future work. We can also potentially explore the adaption of compressed shaping to mass trends of source and channel variations.

REFERENCES

[1] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez, O. Hohlfeld, and G. Smaragdakis, "The lockdown effect: Implications of the COVID-19 pandemic on internet traffic," in *Proc. ACM Internet Measurement Conference (IMC)*, New York, NY, USA, Oct. 2020, pp. 1–18.

[2] K. Roberts, M. O'Sullivan, K.-T. Wu, H. Sun, A. Awadalla, D. J. Krause, and C. Laperle, "Performance of dual-polarization QPSK for optical transport systems," *J. Lightw. Technol.*, vol. 27, no. 16, pp. 3546–3559, Aug. 2009.

[3] Optical Internetworking Forum (OIF), 400ZR. [Online]. Available: www.oiforum.com/technical-work/hot-topics/400zr-2/

[4] Open ROADM MSA, [Online]. Available: www.openroadm.org/home.html

[5] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Trans. Commun.*, vol. 40, no. 3, pp. 873–884, May 1992.

[6] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

[7] A. Guillén i Fàbregas, A. Martinez, and G. Caire, "Bit-interleaved coded modulation," *Found. Trends Commun. Inf. Theory*, vol. 5, nos. 1/2, pp. 1–153, 2008.

[8] L. Szczecinski and A. Alvarado, *Bit-Interleaved Coded Modulation: Fundamentals, Analysis, and Design.* New York, NY, USA: Wiley, 2015.

[9] G. D. Forney, Jr. and L.-F. Wei, "Multidimensional constellations—Part I: introduction, figure of merit, and generalized cross constellation," *IEEE J. Selected Areas Commun.*, vol. 7, no. 6, pp. 877–892, Aug. 1989.

[10] A. R. Calderbank and L. H. Ozarow, "Nonequiprobable signaling on the Gaussian channel," *IEEE Trans. Inf. Theory*, vol. 36, no. 4, pp. 726–740, July 1990.

[11] F. R. Kschischang and S. Pasupathy, "Optimal nonuniform signaling for Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 913–929, May 1993.

[12] G. Böcherer, F. Steiner, and P. Schulte, "Bandwidth efficient and rate-matched low-density parity-check coded modulation," *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 4651–4665, Dec. 2015.

[13] F. Buchali, F. Steiner, G. Böcherer, L. Schmalen, P. Schulte, and W. Idler, "Rate adaptation and reach increase by probabilistically shaped 64-QAM: an experimental demonstration," *J. Lightw. Technol.*, vol. 34, no. 7, pp. 1599–1609, Apr. 2016.

[14] G. Böcherer, P. Schulte, and F. Steiner, "Probabilistic shaping and forward error correction for fiber-optic communication systems," *J. Lightw. Technol.*, vol. 37, no. 2, pp. 230–244, Jan. 2019.

[15] J. Cho and P. J. Winzer, "Probabilistic constellation shaping for optical fiber communications," *J. Lightw. Technol.*, vol. 37, no. 6, pp. 1590–1607, Mar. 2019.

[16] P. Schulte and G. Böcherer, "Constant composition distribution matching," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 430–434, Jan. 2016.

[17] Y. C. Gültekin, F. M. J. Willems, W. J. van Houtum, S. Şerbetli, "Approximate enumerative sphere shaping," in *Proc. IEEE Int. Symp. Inf. Theory*, Vail, CO, USA, Jun. 2018, pp. 676–680.

[18] T. Fehenberger, D. S. Millar, T. Koike-Akino, K. Kojima, and K. Parsons, "Multiset-partition distribution matching," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 1885–1893, Mar. 2019.

[19] T. Yoshida, M. Karlsson, and E. Agrell, "Hierarchical distribution matching for probabilistically shaped coded modulation," *J. Lightw. Technol.*, vol. 37, no. 6, pp. 1579–1589, Mar. 2019.

[20] P. Schulte and F. Steiner, "Divergence-optimal fixed-to-fixed length distribution matching with shell mapping," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 620–623, Apr. 2019.

[21] J. Cho, "Prefix-free code distribution matching for probabilistic constellation shaping," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 670–682, Feb. 2020.

[22] W. G. Bliss, "Circuitry for performing error correction calculations on baseband encoded data to eliminate error propagation," *IBM Techn. Discl. Bul.*, vol. 23, pp. 4633–4634, 1981.

[23] J. L. Fan and J. M. Cioffi, "Constrained coding techniques for soft iterative decoders," in *Proc. Global Telecommunications Conference (GLOBECOM)*, Rio de Janeiro, Brazil, Dec. 1999, vol. 1(B), pp. 723–727.

[24] I. B. Djordjevic and B. V. Vasic, "Constrained coding techniques for the suppression of intrachannel nonlinear effects in high-speed optical transmission," *J. Lightw. Technol.*, vol. 24, no. 1, pp. 411–419, Jan. 2006.

[25] C. E. Shannon, "A matthematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, July/Oct. 1948.

[26] ITU-T, "Interfaces for the optical transport network," 2020. [Online]. Available: www.itu.int/rec/G.709/Y.1331.

[27] K. Sayood and J. C. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Trans. Commun.*, vol. 39, no. 6, pp. 838–846, June 1991.

[28] J. Kliewer and R. Thobaben, "Parallel concatenated joint source–channel coding," *IEE Electron. Lett.*, vol. 39, no. 23, pp. 1664–1665, November 2003.

[29] F. Vacondio, O. Bertran-Pardo, Y. Pointurier, J. Fickers, A. Ghazisaeidi, G. de Valicourt, J.-C. Antona, P. Chanclou, and S. Bigo, "Flexible TDMA access optical networks enabled by burst-mode software defined coherent transponders," in *Proc. Eur. Conf. Opt. Comm. (ECOC)*, London, UK, Sep. 2013, Paper We.1.F.2.

[30] S. Civelli and M. Secondini, "Hierarchical distribution matching: a versatile tool for probabilistic shaping," in *Proc. Opt. Fib. Commun. Conf. (OFC)*, San Diego, CA, USA, Mar. 2020, Paper Th1G.4.

[31] S. Civelli and M. Secondini, "Hierarchical distribution matching for probabilistic amplitude shaping," *Entropy*, vol. 22, no. 9, pp. 958–984, Aug. 2020.

[32] T. Yoshida, M. Karlsson, and E. Agrell, "Joint source-channel coding via compressed distribution matching in fiber-optic communications," in *Proc. Opt. Fib. Commun. Conf. (OFC)*, San Diego, CA, USA, Mar. 2019, Paper M4B.6.

[33] T. Yoshida, M. Binkai, S. Koshikawa, S. Chikamori, K. Matsuda, N. Suzuki, M. Karlsson, and E. Agrell, "FPGA implementation of distribution matching and dematching," in *Proc. Eur. Conf. Opt. Commun. (ECOC)*, Dublin, Ireland, Sep. 2019, Paper M.2.D.2.

[34] T. Yoshida and K. Igarashi, "Probabilistic constellation shaping and quasi data compression in fiber-optic communications," *The Institute of Electronics, Information and Communication Engineers (IEICE) Trans. Commun.*, vol. J103-B, no. 9, pp. 361–371, Sep. 2020 (in Japanese).

[35] Q. Yu, S. Corteselli, J. Cho, "FPGA implementation of prefix-free code distribution matching for probabilistic constellation shaping," in *Proc. Opt. Fib. Commun. Conf. (OFC)*, San Diego, CA, USA, Mar. 2020, Paper Th1G.7.

[36] Q. Yu, S. Corteselli, J. Cho, "FPGA implementation of rate-adaptable prefix-free code distribution matching for probabilistic constellation shaping," *J. Lightw. Technol.*, DOI: 10.1109/JLT.2020.3035039, Nov. 2020.

[37] D. A. Huffman, "A method for the construction of minimum-redundancy codes," in *Proc. the I.R.E.*, Sep. 1952, pp. 1098–1102.

[38] S. Zhang, Z. Qu, F. Yaman, E. Mateo, T. Inoue, K. Nakamura, Y. Inada, and I. B. Djordjevic , "Flex-rate transmission using hybrid probabilistic and geometric shaped 32QAM," in *Proc. Opt. Fib. Commun. Conf. (OFC)*, San Diego, CA, USA, March 2018, Paper M1G.3.

[39] T. Yoshida, E. Agrell, and M. Karlsson, "Hierarchical distribution matching with massively parallel interfaces for fiber-optic communications," in *Proc. International Zurich Seminar on Information and Communication (IZS)*, Zurich, Switzerland, Feb. 2020, pp. 16–20.

[40] J. Cho, S. L. I. Olsson, S. Chandrasekhar, P. Winzer, "Information rate of probabilistically shaped QAM with non-ideal forward error correction," in *Proc. Eur. Conf. on Opt. Comm. (ECOC)*, Roma, Italy, Sep. 2018, p. Th.1.H.5.

[41] L. Schmalen, A. Alvarado, R. Rios-Müller, "Performance prediction of nonbinary forward error correction in optical transmission experiments," *J. Lightw. Technol.*, vol. 35, no. 4, pp. 1015–1027, Feb. 2017.

[42] S. ten Brink and R. Mahadevappa, "Implementation aspects of high-speed wireless LAN systems," *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 2004, pp. 789–793.

[43] European Telecommunications Standards Institute, "Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications; Part 1 (DVB-S2)," ETSI Standard EN 302 307-1 V1.4.1, Nov. 2014. [Online]. Available: www.dvb.org/standards

[44] T. Yoshida, M. Karlsson, and E. Agrell, "Performance metrics for systems with soft-decision FEC and probabilistic shaping," *IEEE Photon. Technol. Lett.*, vol. 29, no. 23, pp. 2111–2114, Dec. 2017.

[45] T. Yoshida, A. Alvarado, M. Karlsson, and E. Agrell, "Post-FEC BER benchmarking for bit-Interleaved coded modulation with probabilistic shaping," *J. Lightw. Technol.*, vol. 38, no. 16, pp. 4292–4306, Aug. 2020.

[46] J. Cho, L. Schmalen, and P. J. Winzer, "Normalized generalized mutual information as a forward error correction threshold for probabilistically shaped QAM," in *Proc. Eur. Conf. Opt. Commun. (ECOC)*, Gothenburg, Sweden, Sep. 2017, Paper M.2.D.2.

[47] Y. Cai, W. Wang, W. Qian, J. Xing, K. Tao, J. Yin, S. Zhang, M. Lei, E. Sun, K. Yang, H. Chien, Q. Liao, and H. Chen, "FPGA investigation on error-floor performance of a concatenated staircase and Hamming code for 400G-ZR forward error correction," in *Proc. Opt. Fib. Commun. Conf. (OFC)*, San Diego, CA, USA, March 2018, Paper Th4C.2.