Backstrom, Tom; Fischer, Johannes

Fast Randomization for Distributed Low-Bitrate Coding of Speech and Audio

Published in:
IEEE/ACM Transactions on Audio Speech and Language Processing

DOI:
10.1109/TASLP.2017.2757601

Published: 01/01/2018

Document Version
Peer reviewed version

Please cite the original version:
Backstrom, T., & Fischer, J. (2018). Fast Randomization for Distributed Low-Bitrate Coding of Speech and Audio. *IEEE/ACM Transactions on Audio Speech and Language Processing*, *26*(1), 19-30. https://doi.org/10.1109/TASLP.2017.2757601

Fast Randomization for Distributed Low-Bitrate Coding of Speech and Audio

Tom Bäckström, *Senior Member, IEEE,* and Johannes Fischer

***Abstract*—Efficient coding of speech and audio in a distributed system requires that quantization errors across nodes are uncorrelated. Yet with conventional methods at low bitrates, quantization levels become increasingly sparse, which does not correspond to the distribution of the input signal and importantly, also reduces coding efficiency in a distributed system. We have recently proposed a distributed speech and audio codec design which applies quantization in a randomized domain such that quantization errors are randomly rotated in the output domain. Similar to dithering, this ensures that quantization errors across nodes are uncorrelated and coding efficiency is retained. In this paper we improve this approach by proposing faster randomization methods, with a computational complexity of $\mathcal{O}(N \log N)$. Presented experiments demonstrate that the proposed randomizations yield uncorrelated signals, that perceptual quality is competitive and that the complexity of the proposed methods is feasible for practical applications.**

***Index Terms*—orthonormal matrix, superfast algorithm, randomization, distributed coding, speech coding, audio coding**

I. Introduction

DIGITAL compression of speech signals for transmission and storage applications, known as speech coding, is a classic topic within speech processing and modern speech coding standards achieve high efficiency in their respective application scenarios [1]–[5]. Though these standards are high-fidelity products, they are constrained to configurations with a single encoder. Designs which would allow using the microphones of multiple independent devices could improve signal quality, and moreover, it would allow a more natural interaction with the user-interface as the speaker would no more be constrained to a single device. If the codec can flexibly use all available hardware, then the user does not need to know which devices are recording, releasing mental capacity from attention to devices to the communication at hand.

Such an ideal user interface is possible only if devices cooperate in the speech coding task. The aim is that, through cooperation, the acoustic signal should be flexibly captured and transmitted to one or several decoders or fusion centers. Clearly we thus require a *distributed* speech and audio codec.

A distributed system however requires substantial modifications to existing codec designs; most notably, 1) the increase in algorithmic complexity due to added nodes becomes an issue and 2) we need a method to ensure that each transmitted bit conveys unique information. Specifically, conventional codec designs are based on an intelligent encoder and a simple decoder, whereby a majority of the computational complexity resides at the encoder. In a distributed system, the overall computational complexity increases linearly with both the encoder complexity as well as the number of nodes, whereby it is important to keep encoder complexity low to be able to use a large number of nodes. If we can move the main intelligence of the codec from the encoder to the decoder, then the overall complexity of the system would thus be much lower.

A majority of speech coding standards are based on the code-excited linear prediction (CELP) paradigm [1]. It is based on an analysis-by-synthesis loop, where the perceptual quality of a large number of different quantizations are evaluated to optimize output quality. While this approach provides the best quality for bitrate trade-off, its usefulness in distributed coding is limited by its computational complexity, rigid design and error propagation issues. Frequency domain methods, on the other hand, have not yet reached quite the same efficiency as CELP, but it is clear that coding in the frequency domain is computationally much simpler. Moreover, since most noise attenuation and spatial filtering methods are defined in the frequency domain [6], it will be straightforward to implement such methods if we use frequency-domain coding, and we follow the approach proposed in the current paper.

Another issue is the amount of interaction between encoder nodes. Clearly communication between nodes requires some administration, whereby it would be beneficial to minimize or even avoid interaction between nodes if possible. If nodes only transmit data and we avoid interaction between nodes, then the overall system structure is simpler and we avoid the computational complexity required for said interaction. The question is thus whether interaction between nodes is required for coding efficiency. At high bit-rates (say 100 kbits/s), very small differences in the signal, such as variations in delay, background noise or sensor noise, would be sufficient to make quantization noise between nodes uncorrelated [7], whereby each node will provide unique information. However, experience with lower bit-rates (such as 10 kbits/s) has shown that low-energy areas of the signal are often quantized to zero, whereby quantization errors are perfectly correlated with the input signal [1]. Multiple nodes transmitting zeros would then convey no new information about the signal, whereby there is little advantage of using multiple devices.

Our objective is to develop a distributed codec for speech and audio, where coding efficiency is optimized, but which can also be applied on any device, including simple mobile or even wearable devices with limited CPU and battery resources. Recently, we have proposed an overall design for

T. Bäckström is with the Department of Signal Processing and Acoustics, Aalto University, Espoo, Finland. Email: tom.backstrom@aalto.fi

J. Fischer is with the International Audio Laboratories Erlangen, a joint institute of Fraunhofer IIS and Friedrich-Alexander University (FAU), Germany.

Manuscript received Month XX, 201Y; revised Month XX, 201Z.

such a method [8], [9]. The approach is based on randomizing the signal before quantization, such that quantization error expectations between devices are uncorrelated. We assume that the randomizer uses a random-number generator whose seed is communicated from the encoder to the decoder either offline or sufficiently seldom that it has a negligible effect on the bitrate. Overall, the randomizer in this context is similar to dithering and was inspired by the 1 bit quantization used in compressive sensing [10], [11].

Randomization has several distinct benefits in the proposed codec: 1) In *low-bitrate coding* (below 10 kbits/s), we have only a limited number of quantization levels which can be encoded with the available bits. With decreasing bitrate, the quantized signal distribution thus becomes increasingly sparse, granular and biased. By applying a randomization and its inverse before and after quantization, respectively, we can hide the undesirably sparse structure. Similarly as dithering, we can thus retain the signal distribution, without any penalty on the signal to noise ratio. 2) In *perceptual audio coding* a too low number of quantization levels for speech and audio signals leads to artifacts known as musical noise, where components which sporadically appear and disappear become coherent sound objects in their own right. A standard approach for avoiding musical noise in audio codecs is noise filling, a method similar to dithering, where noise is added to spectral areas quantized to zero [12]. In our approach, by quantization in randomized domain, errors become incoherent and we can avoid the reduction in SNR caused by noise filling. 3) Randomization of the signal can also work as a component of *encryption* [13]. It provides diffusion in a similar way as the Hill cipher [14], that is, it distributes the contribution of the input vector evenly onto the bitstream. 4) In *distributed coding*, we can apply two alternative approaches. If nodes encode separate subspaces (or cosets in the vocabulary of distributed coding), then increasing the bitrate by 1 bit/sample yields a 6 dB improvement in quality. The downside in an ad-hoc network is that then the nodes have to negotiate which subspaces/cosets to transmit, which requires extra bandwidth, administration and increases the risk of eavesdropping. Moreover, spatio-temporal filtering such as beamforming is impossible if cosets do not overlap. On the other hand, if nodes are independent, then we can get a 3 dB improvement from a doubling of the number of nodes, as long as the quantization errors are uncorrelated [6]. Randomized quantization yields quantization errors which are uncorrelated, whereby in difference to conventional quantization, we achieve the 3 dB improvement when doubling the number of microphones. 5) When *transmission errors* corrupt some transmitted bits, conventional entropy coders (e.g. arithmetic coding) will loose all data after the first corrupted bit. With the proposed entropy coding which is enabled by the randomizing scheme, there is no serial dependency of bits, whereby we can reconstruct the signal also when some bits are corrupted. The transmission errors will then be visible as noise in the reconstructed signal, which can be attenuated by conventional noise attenuation methods such as Wiener filtering [15], [16]. The details of these benefits are discussed in the following sections.

In comparison, conventional single-device quantization and coding methods all suffer from some constraints. Entropy coders such as Huffman or arithmetic coders with uniform quantization do not scale to very low bitrates (less than 2 bits/sample), since the output signal distribution becomes unnaturally sparse [1], [17]. Lattice quantization does reduce quantization error, but does not solve the issue of granularity at low bitrates and moreover, it does not easily lend itself to arbitrary probability distributions [1], [18]. Vector coding is optimal in accuracy and does not suffer much from sparsity, but computational complexity is high and it is challenging to encompass variable bitrates [19]. Moreover, achieving robustness to transmission errors is difficult with all of the above methods.

Distributed source coding methods, on the other hand, do provide methods for optimal joint encoding [20], [21]. These methods make use of the correlation between the microphone signals by binning in order to reduce the rates using the results from distributed source coding. The most common way of implementing this is by using error-correcting codes, but in a practical setup due to complexity and other considerations, such implementations will be highly suboptimal, leading to higher complexity without significant gains. For these reasons, in the current work, we do not focus on binning. Specifically, we are not aware of distributed source coding methods for ad-hoc networks, which would include signal-adaptive perceptual modeling, which would solve the above mentioned problems with sparsity and which would simultaneously apply signal-adaptive source models. All of these properties are required features of a speech and audio codec to retain a competitive performance for the single-channel case.

The fields of speech and audio coding [1], [22], and distributed source coding e.g. [20], [21], are well-understood topics. Work on wireless acoustic sensor networks has however not yet made it to consumer products of the type discussed here [23]–[25]. Some works have addressed higher bitrates [26], or with the assumption that nodes are fully connected and co-operating [7], [27], though both approaches lack a perceptual model. Some early work do apply a perceptual model [28], [29], but do not include other elements of main-stream speech and audio codecs, such as source modeling. A somewhat similar problem is design of hands-free devices and the associated beamforming tasks [6], which can be applied in a distributed system [30], though methods usually require accurate synchronization of nodes to give competitive performance [31]. Similar methods have also been tried for hearing-aids, though distributed source coding gave there only a negligible improvement [32]. More generally, distributed processing of speech and audio can be also applied for classification and recognition tasks [33]–[35], though those remain well outside the scope of the current work.

In comparison, the proposed distributed scheme is now a complete system, with the exception of speech source modeling, which has not yet been incorporated into the system, though we have studied it intensely, e.g. [36], [37]. Moreover, since source modeling is a clearly distinct and large topic, we have left it to a future publication. Specifically, while we do apply a rudimentary entropy codec, in the current experiments we do not include a model of the spectral magnitude envelope,

of harmonic structure nor spatio-temporal correlations. While many of the prior works have admirably fine theoretical analyses, a central novelty of the current work is that the design has no barriers against creating a practical system whose single-channel quality is near state-of-the-art while simultaneously providing a benefit when increasing the number of nodes. That is, by including the above mentioned source models, the single-channel performance should be similar to the performance of the TCX mode of the EVS standard [2]. It should be emphasized that we have not included all details of best practices in lossy distributed source coding, since we have opted to take incremental steps in order to retain the single-channel quality near the state-of-the-art.

The current contribution addresses the complexity bottleneck of the system, namely, the randomization and its inverse. The algorithmic complexity of generic quantization together with randomization is $\mathcal{O}(N^2)$. Moreover, at the decoder, our original approach required the inversion of an $N \times N$ matrix, which gives a complexity of $\mathcal{O}(N^3)$ using Gaussian elimination [38]. It is our objective here to present methods which improve algorithmic complexity, and retain or improve the randomization properties and coding efficiency as much as feasible.

In addition to distributed coding, randomization and decorrelation are used also in many other fields of speech, audio and signal processing in general. For example, in upmixing of audio signals from a low to a higher number of channels, we need methods for generating uncorrelated source signals [39]. Randomization methods proposed in this paper may find application in any such applications which require low-complexity methods for generation of uncorrelated signals. Notably, moreover, randomization can be used in single-channel codecs to diffuse unnaturally sparse quantization levels which appear at low bitrates.

II. Randomized Entropy Coding

The main objective of coding is to quantize and encode an input signal $x \in \mathbb{R}^{N\times 1}$, with a given number of bits B, such that it can be decoded with highest accuracy possible. The objective of randomization, on the other hand, is to make sure that the resynthesised signal retains the continuous distribution of the original signal and that the quantization error is uncorrelated Gaussian noise. In other words, whereas quantization by construction yields a signal with a discrete distribution, our objective is to obtain a signal which follows a similar distribution as the original signal. Moreover, the aim is that if the signal is quantized and coded at multiple nodes, then the quantization errors of the outputs would be uncorrelated. Clearly we then need to introduce randomness in the signal without reducing accuracy of the reconstruction.

To achieve such randomness, we discuss three aspects of linear randomizing transforms (see Fig. 1); First, we show that orthonormal projections are optimal for our error criterion. Secondly, we discuss random permutations for diffusing information across the input vector. Finally, we demonstrate that low-order random rotations can be used in block-matrices to diffuse quantization levels.

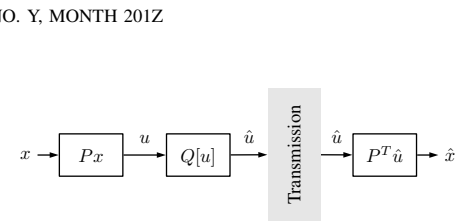

Fig. 1. Flow-diagram of the randomization process where P is a random (orthonormal) matrix and $Q[\cdot]$ is a quantizer.

As objective design criteria for the randomization, we use the following methods: 1) The accuracy of reconstruction is measured by the minimum mean square error $\min E[\|e\|^2]$, where $e = x - \hat{x}$ is the quantization error and $\hat{x}$ is the quantized signal. 2) To measure the correlation between randomized vectors, we measure the normalized covariance between the original x and its randomized counterpart Px. If the randomization is effective, then the normalized covariance should behave like the normalized covariance $\frac{x^T y}{\|x\|\,\|y\|}$ between two uncorrelated signals x and y. Specifically, the mean of the normalized covariance should be zero and its variance $\frac{1}{N}$ for Gaussian signals (see Appendix for details). 3) The accuracy with which the distribution of the output signal follows the distribution of the input signal can be measured with the Kullback-Leibler (KL) divergence. However, since analytic analysis of divergences is difficult, we apply the KL-divergence only experimentally in Sec. IV. 4) Finally, algorithmic complexity is characterized with the Big-O notation.

A. Orthonormal Randomization

To introduce randomness in the signal without compromising accuracy, we propose to multiply the signal with a random orthonormal matrix P before quantization. At the decoder we then need to multiply with the inverse P^T (see Fig. 1). It is important that the transform is orthonormal, since it preserves signal energy, such that the transform provides perfect reconstruction and a minimal white noise gain. Specifically, let the output signal be $\hat{x} = P^T Q[Px]$ where $Q[\cdot]$ signifies quantization. If the quantization is perfect, $u = Q[u]$, then we have $\hat{x} = P^T Q[Px] = P^T Px = x$. In other words, the randomizing transform P does not corrupt the signal in any way; all information is retained and the signal can be perfectly reconstructed.

Moreover, if the quantization error is $v = u - \hat{u}$, then the output error energy will be

$$\begin{aligned} \|e\|^2 := \|x - \hat{x}\|^2 &= \left\|x - P^T Q[Px]\right\|^2 \\ &= \|Px - Q[Px]\|^2 = \|v\|^2. \end{aligned} \tag{1}$$

In other words, since for an orthonormal P we have $\|e\|^2 = \|Pe\|^2$, it follows that quantization error energy in the transform domain is exactly equal to the error in the output domain.

If we would relax the constraint from orthonormal matrices, and consider matrices P whose samples are uncorrelated and have unit variance, then we would have $\mathrm{E}[P^T P] = I$ and the matrices would be orthonormal with respect to the expectation. However, as is known from random matrix theory [40], the eigenvalue distribution of such matrices is significantly off unity. It follows that P^T would not be an accurate inverse

of the transform but we would have to use the actual inverse P^{-1} or a pseudo-inverse. Consequently, the inverse transform P^{-1} would emphasize the quantization errors corresponding to small eigenvalues, whereby the inverse transform would, on average, increase the error energy. Orthonormal random matrices are thus preferable to random matrices since they provide perfect reconstruction and unit white noise gain. Orthonormal matrices have also computational benefits with respect to noise attenuation at the decoder side (see Section III).

B. Random Permutations

Permutations are computationally fast operations which correspond to orthonormal matrices, whereby their use in randomization is interesting. As a matter of definition, we will say that a permutation perfectly diffuses input information over the output vector if the output location of all input samples are uniformly distributed over the whole vector. Specifically, if an input sample ξ_h has an input location $h \in [0,\, N-1]$, then the probability that it will appear at location k after permutation is $p(k) = \frac{1}{N}$, whereby the input and output locations are not correlated.

The covariance c_P of the original signal x and the permuted signal $y = Px$ is

$$c_P = x^T y = x^T P x = \sum_{k=1}^{N} \xi_k \xi_{q(k)}, \tag{2}$$

where $q(k)$ is the permutation function. If we define the set of fixed points of the permutation as $S = \{k \,|\, k = q(k)\}$, that is, this is the set of samples which do not move due to the permutation, whereby

$$c_P = \sum_{k \in S} |\xi_k|^2 + \sum_{k \notin S} \xi_k \xi_{q(k)}. \tag{3}$$

While the expectation of the latter sum is zero $E[\xi_k \xi_h] = 0$, for $k \neq h$, the former sum has a non-negative expectation. It follows that the expectation of the covariance has a non-negative bias $E[c_p] \geq 0$, which is in contradiction with our objective. The set S is, however, small when N is large, whereby the deviation is not large. Nevertheless, for the sake of completeness, we can easily remedy the problem.

Let $\Lambda_\pm$ be a diagonal matrix whose diagonal elements are randomly chosen as ± 1. Clearly this matrix is also orthonormal. We can thus apply a randomization $y = \Lambda_\pm P x$, whereby the correlation is

$$c_P = x^T y = x^T \Lambda_\pm P x = \sum_{k=1}^{N} \pm \xi_k \xi_{q(k)}. \tag{4}$$

If both signs have equal probability, then clearly $E[\pm \xi_k \xi_{q(k)}] = 0$ and $E[c_P] = 0$ as required. The combination of random signs and permutations thus decorrelates the signal in the sense that the covariance has zero mean and we simultaneously achieve perfect diffusion. Moreover, multiplication by $\Lambda_\pm$ can be readily generalized to orthonormal block-matrices, which are discussed below.

Random permutations can be easily generated at algorithmic complexity $\mathcal{O}(N \log N)$ [41]. A heuristic approach is for example to apply a sort algorithm, such as merge sort, on a vector of random uncorrelated samples. The sort operation then corresponds to a permutation, which is uniformly distributed over the length of the vector.

C. Block-wise Random Rotations

Multiplication with matrices has in general an algorithmic complexity of $\mathcal{O}(N^2)$. To reduce complexity, consider $N \times N$ random orthonormal block-diagonal matrix rotations B of the form

$$B = \begin{bmatrix} Q_1 & & & 0 \\ & Q_2 & & \\ & & \ddots & \\ 0 & & & Q_K \end{bmatrix}, \tag{5}$$

where K is the number of blocks and Q_k are random orthonormal matrices of size $N_k \times N_k$ such that $\sum_{k=1}^{K} N_k = N$. The complexity of the transform is $\mathcal{O}(\sum_{k=1}^{K} N_k^2)$. Clearly the random sign operation $\Lambda_\pm$ is a block-matrix of this form with $N_k = 1$ and $K = N$.

Specifically, consider size 2×2 orthonormal matrices Q of the form

$$Q = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}. \tag{6}$$

The related covariance is

$$\begin{aligned} c_Q := x^T Q x &= \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}^T \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \\ &= (\xi_1^2 + \xi_2^2)\cos\alpha. \end{aligned} \tag{7}$$

If α is uniformly distributed on $\alpha \in [0,\, 2\pi]$, then clearly c_Q has zero mean, $E[c_Q] = \|x\|^2 E[\cos\alpha] = 0$ as desired. Moreover, if $y = Qx = \begin{bmatrix} \eta_1, \eta_2 \end{bmatrix}^T$, then the parameters η_k follow the arcsine-distribution on the range $|\eta_k|^2 \leq 2\|x\|^2$.

In other words, by applying 2×2 blocks of random rotations on the quantized signal $\hat{u}$, we can diffuse the quantization levels to make the distribution of the output less sparse. Note that since the outputs of the 2×2 transforms have symmetric distributions, sign-randomization by $\Lambda_\pm$ becomes redundant and can be omitted. The 2×2 matrices were chosen since they are simple to implement and any larger orthonormal rotation can be implemented as a combination of 2×2 rotations.

Diffusion of the quantization levels is however not yet complete; the arcsine-distribution is less spiky than the distribution of the quantized signal, but it is still far from the normal distribution. To obtain an output distribution which better resembles the normal distribution, we therefore apply a sequence of permutations P_k and block rotations B_k as

$$P = \prod_{k=1}^{M} B_k P_k, \tag{8}$$

where each block rotation B_k is of form Eq. 5 with 2×2 rotations Q_k of the form in Eq. 6. Each consecutive randomization $B_k P_k$ will then further diffuse the output distribution. We will experimentally determine a number of rotations M such that the output distribution is sufficiently diffused. However, the algorithmic complexity of applying the above rotation will in any case be $\mathcal{O}(MN)$, while generation of such permutations has complexity $\mathcal{O}(N \log N)$.

D. Conventional randomization methods

Generating random orthonormal matrices, with uniformly distributed rotations is not as easy as it would seem. With 2×2 matrices such as those in Eq. 6 we can get uniform distribution if α is uniformly distributed on $[0,\, 2\pi]$, however, with $N > 2$ such heuristic approaches are not as simple anymore. The reason is easy to understand in the 3×3 case, where uniform rotations along each axis would yield a higher concentration of points around the poles of the sphere.

In the general case, however, the problem is equivalent with choosing N random points on the unit N-sphere, such that the corresponding unit vectors are orthonormal. We can thus choose N random vectors of size $N \times 1$ from a distribution with spherical symmetry, and orthogonalize the set of vectors. A numerically stable and efficient algorithm which generates such orthonormal vectors is the QR-algorithm [38]. Specifically, we first generate an $N \times N$ matrix X with uncorrelated and normally distributed samples with zero mean and equal variance. Secondly, we apply the QR-algorithm to obtain an orthonormal matrix P_{QR}. The overall algorithmic complexity of this approach is $\mathcal{O}(N^2)$ [38], [41].

A simplification of the QR-algorithm is to apply Householder transformations with each of the columns of X [42]. While this approach is efficient for small matrices, in informal experiments we found that for large matrices, the simplification does not provide a uniform distribution. We also tried the subgroup algorithm presented in [43], [44]. Though this algorithm is faster than the Householder-based algorithm by a factor of 2, unfortunately however, it suffers from the same issues as the Householder-based algorithm. The QR-algorithm applied on random matrices thus remains our high-quality and -complexity method for reference.

E. Algorithmic complexity

In application of each of the proposed orthonormal matrices, we have two sources of algorithmic complexity; one which emerges from generation of the matrix and a second which is related to application of the matrix and its inverse. Furthermore, we should evaluate storage requirements for the generated matrices. In each case, we shall assume that we have access to a pseudo-random number generator, which produces a pseudo-random sequence of scalars ξ_k with independent and identically distributed values from the uniform distribution on $\xi_k \in [0, 1]$.

The simplest case is generation of random signs for the matrix $\Lambda_\pm$. For each of the N diagonal samples we need a random sign with equal probability, which can be obtained by thresholding ξ_k at 0.5. Given the sequence ξ_k the algorithmic complexity for generation and application is thus $\mathcal{O}(N)$.

For the block rotations, the number of 2×2 blocks is $N/2$, whereby we need $N/2$ random scalars to generate the matrix B. Application of B involves $N/2$ multiplications by a 2×2 matrix at complexity $\mathcal{O}(2N)$, as well as $N/2$ evaluations of $\cos\alpha$ and $\sin\alpha$ at $\mathcal{O}(N)$, though evaluation of trigonometric functions can have a high constant multiplier for the complexity.

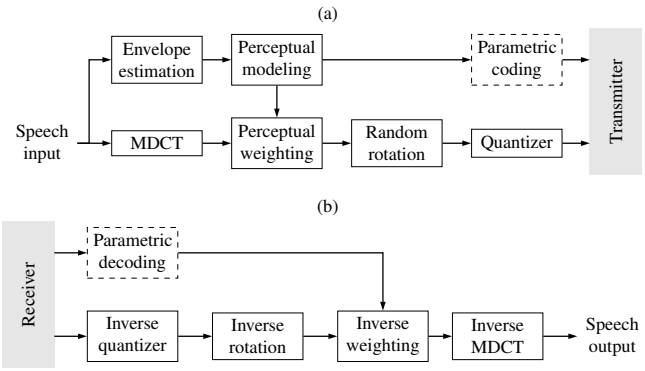

Fig. 2. Structure of the (a) encoder and (b) decoder of one node of the distributed speech and audio codec. Dashed boxes indicate modules which were not included in current experiments to facilitate isolated evaluation of randomization.

Application of permutations is straightforward; it is essentially a mapping of sample indices, whereby it does not involve computations other than moving operations, $\mathcal{O}(N)$. Here we have to, though, store both the permutation indices and we need to store the permuted vector, whereby the storage requirements are $\mathcal{O}(2N)$. Generation of permutations can be applied by sorting a vector of random scalars ξ_k with, for example, the merge sort algorithm [41]. It requires also a storage of $\mathcal{O}(2N)$, but not at the same time as the application of the permutation, whereby it does not add to the overall storage requirements. The algorithmic complexity for generating the permutation is $\mathcal{O}(N \log N)$ [41].

To generate a random orthonormal matrix, the QR-algorithm can be applied with arbitrary accuracy with an algorithmic complexity of $\mathcal{O}(N^2)$ and storage $\mathcal{O}(N^2)$ [38]. Application of the randomization and its inverse are then simply multiplications by a matrix and its transpose, both at complexity $\mathcal{O}(N^2)$. It however requires N^2 random scalars as input, whereby also the complexity of generating pseudo-random numbers becomes an issue. Moreover, the random values at the input should have rotational symmetry, whereby the uniformly distributed scalars ξ_k variables are not sufficient. We thus need to apply a transform such as the inverse cumulative normal distribution on ξ_k to obtain normally distributed variables, which comes at a considerable extra computational cost.

Each of the above algorithms assume that we have access to a sequence of pseudo-random numbers ξ_k. If we choose to generate the randomization on-line, then we need to consider the complexity of generating pseudo-random numbers. The algorithmic complexity of generating pseudo-random numbers is in general linear $\mathcal{O}(N)$ with the number N of scalars to be generated [45]. A commonly used generator is the Mersenne-twister, though there are also lower-complexity versions available [46], [47]. The trade-off is that if the random sequence is not generated on-line, then it needs to be stored. In any case, we assume that the seed of the random sequence is communicated either off-line, or sufficiently seldom such that it does not induce a significant penalty on the bit-rate.

In summary, algorithmic complexity of generating random matrices is $\mathcal{O}(MN \log N)$, while their application has

$\mathcal{O}(MN)$, where M is the number of iterations (typically $M = 4$) and N is the vector length. If the random coefficients are not generated on-line but stored, then we need storage of $\frac{3}{2}MN$ coefficients, while working memory must be always at least $2N$ coefficients.

A typical speech and audio codec, such as the TCX mode of the EVS, would use a step of 20 ms between windows [1], [2], whereby the spectra would be of length $N = 256$ at a sampling rate of 12.8 kHz, $N = 320$ at 16 kHz or $N = 882$ at 44.1 kHz. A typical frequency domain codec will have no components which require a complexity more than $\mathcal{O}(N \log N)$. Since the proposed randomization is also $\mathcal{O}(N \log N)$, in terms of algorithmic complexity, we are now in-line with conventional TCX codecs. The complexity bottleneck thus returns to the rate-loop of the entropy codec [1], [48].

III. Application in Distributed Coding of Speech And Audio

As a demonstration of the proposed randomizer, we applied the randomized quantizer in coding of the fine-spectral structure in a distributed speech and audio codec. The overall structure is similar to that of the TCX mode in 3GPP EVS [2] and the implemented codec structure is illustrated in Fig. 2. First, we apply the MDCT time-frequency transform and half-sine windowing on the input signal [22] to obtain spectral representations x_k of each time-frame at a node k. Here the window length was 20 ms with 10 ms overlap, and the sampling rate was 12.8 kHz. The sampling rate was chosen to match the core-rate of EVS in wide-band mode [2]. In EVS, the remaining bandwidth is coded with bandwidth-extension methods, to obtain a total sampling rate of 16 kHz.

We then analyze the signal envelope and perceptual model, using the LPC-based approach as in EVS [1], [2]. The signal is then perceptually weighted and multiplied with random rotation matrices to obtain randomized vectors. As a last step of the encoder, the signal is quantized and coded.

We used a fixed-bitrate entropy coder with 2 bits/sample as follows; the distribution was split into four quantization cells such that each cell had equal probability and each input sample was quantized to the nearest quantization cell. The quantization levels are thus fixed and the system does not require a rate-loop. This corresponds to entropy coding the spectral fine structure alone with a bitrate of 25.6 kbits/s. Perceptual envelopes and signal gain are usually transmitted with a rate in the range 2–3 kbits/s, whereby the overall bitrate is approximately 28 kbits/s. This does not strictly qualify as a low-bitrate codec, but on the other hand, we have here not implemented an explicit source model nor a rate-loop. Our experience with the EVS codec suggests that inclusion of a source model (such as a fundamental frequency model) and a proper rate-loop, would reduce bitrate to below 10 kbits/s without reduction in perceptual quality, whereby this experiment is representative for low-bitrate coding. Conversely, we chose the bitrate such that it roughly corresponds to the accuracy achieved in TCX in the EVS standard well below 10 kbit/s.

At the decoder, the operations are reversed, with the exception of the perceptual model estimation, which is assumed to be transmitted. Note that we want to focus on the performance of the randomizer, whereby in the current application, we did not include an explicit source model, rate-loop nor quantize the perceptual model. The quality of the output signal could be further enhanced with noise attenuation techniques such as Wiener filtering [6], [49]. Here, however, we chose to omit noise attenuation such that we can avoid tuning parameters and keep the comparison of different methods fair.

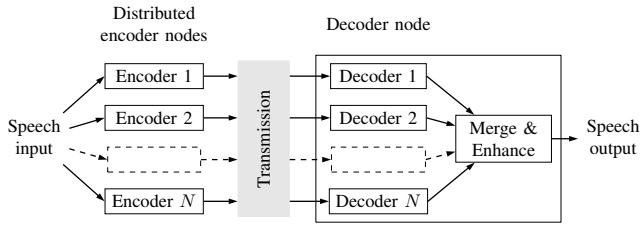

Fig. 3. Encoder/decoder structure of the distributed speech and audio codec with N encoder nodes and a single decoder node.

To demonstrate performance in a multi-node scenario, we implemented a distributed codec as illustrated in Fig. 3, where each individual encoder node follows the configuration of Fig. 2. As described above, the decoder can then contain the inverse randomizations and a "merge & enhance" -block can implement Wiener filtering independently from the randomization. As long as the quantization errors are orthonormal and quantization accuracy is uniform, we would not gain anything from joint processing of the randomization and enhancement, whereby independent blocks give optimal performance. We chose not implement Wiener filtering here, since it would present additional perceptual tuning factors, whereby the design of a fair comparison would be difficult. Instead we took here merely the mean of the two channels and study only the objective quality of the two-channel case.

IV. Experiments

To evaluate the proposed randomization methods, we performed objective experiments corresponding to the performance measures described in Section II as well as subjective perceptual experiments with the distributed speech and audio codec described in Section III.

A. Statistical properties

To quantify the influence that randomization has on the coding error magnitude, we created $N \times N$ matrices with $N = 100$, such that 1) the matrices P_o were orthonormal $P_o^T P_o = I$ and 2) the matrices P_r were orthonormal with respect to the expectation $E[P_r^T P_r] = I$. Specifically, we used a random number generator to produce uncorrelated, normally distributed samples with zero mean and variance $\frac{1}{N}$, which form the entries of P_r. It follows that $E[P_r^T P_r] = I$. By applying the QR-algorithm on P_r, we then obtain a random orthonormal matrix, which we define as P_o.

We then generated $K = 1000$ vectors x of length N, whose samples are uncorrelated and follow the normal distribution. Each vector was randomized with the two matrices $u_o = P_o x$ and $u_r = P_r x$, quantized with the sign quantization

Method	Orth	Rand	None
SNR (dB)	4.52	2.25	4.52

TABLE I
SIGNAL TO NOISE RATIO (SNR) OF SIGN QUANTIZATION WITH ORTHONORMAL RANDOMIZATION (ORTH), RANDOMIZATION WITH A RANDOM MATRIX (RAND) AND WITHOUT RANDOMIZATION (NONE).

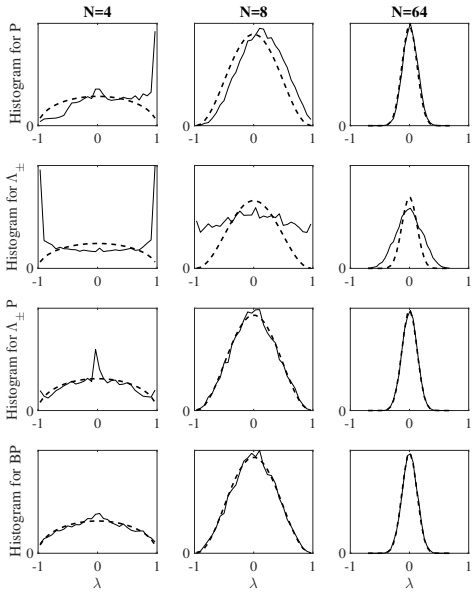

Fig. 4. Illustration of histograms of the normalized covariance $\lambda = \frac{x^T Ax}{\|x\|^2}$ for different $N \times N$ orthonormal matrices: the random permutation matrix P, the random sign matrix $\Lambda_\pm$, the combination $\Lambda_\pm P$ as well as the block-matrix (with random 2×2 rotations) in combination with permutation BP, evaluated over $K = 10\,000$ matrices. The dashed line indicates the theoretical distribution of normalized covariance between random Gaussian vectors, scaled to fit to each histogram.

$\hat{u}_o = \text{sign}(u_o)$ and $\hat{u}_r = \text{sign}(u_r)$ and the randomization was reverted by $\hat{x}_o = P_o^T \hat{u}_o$ and $\hat{x}_r = P_r^T \hat{u}_r$. As reference, we used quantization without randomization as $x_{ref} = \text{sign}(x)$. Finally, each of the vectors x_o, x_r and x_{ref} were individually scaled for minimum error, and the quantization error energy for each vector was calculated.

The results of this experiment are listed in Table I. Clearly randomization with the random matrix P_r yields a much lower SNR upon reconstruction, whereas randomization with the orthonormal matrix P_o has no influence on accuracy. The results thus exactly follow the theory in Section II and we should always use randomization by orthonormal matrices.

To determine the efficiency of randomization in terms of decorrelation, we then calculated normalized covariances $\lambda = \frac{x^T Ax}{\|x\|^2}$ for the different orthonormal matrices A. Fig. 4 illustrates the histogram of $K = 1000$ iterations of the normalized covariances for matrices of different sizes $N = \{4, 8, 256\}$ as well as for random permutations P, random signs $\Lambda_\pm$ and random permutations with random signs $P\Lambda_\pm$. As a reference, we used the theoretical distribution illustrated with a dashed line (see Appendix for details).

We observe that the distribution of the random permutation P follows the the theoretical distribution (dashed line) at higher N. However, it is biased to positive values especially at lower N. The random sign $\Lambda_\pm$ yields a covariances whose variance (width of histogram) is higher than the theoretical

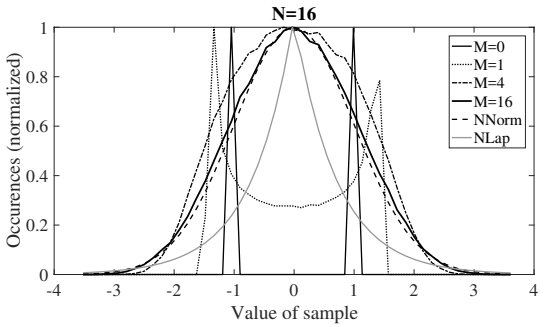

Fig. 5. Normalized histograms of the output samples after M consecutive randomizations, as well as the theoretical distributions of normalized Gaussian (*NNorm*, dashed line) and Laplacian (*NLap*, gray line), for a vector of length $N = 16$.

distribution. Clearly neither method is sufficient alone in decorrelating the signal. The combination of random signs and a permutation $\Lambda_\pm P$ performs much better in that the bias to positive values is removed and the variance is similar to the theoretical distribution. At $N = 4$, however, we observe that the input signal x cannot be treated as a random variable anymore, but since samples of x frequently get multiplied with itself (though with random sign), we obtain peaks in the histogram corresponding to ± 1 and 0. The situation is further improved by replacing the random signs with block-matrices B, where the 2×2 blocks are calculated with uniformly distributed angles α. The peaks at ± 1 and 0 for $N = 4$ have almost completely disappeared and the histogram nicely follows the theoretical distribution. Overall, we find that the decorrelation performance of randomization improves with increasing vector length, as well as when we use a combination of at least two orthonormal matrices.

The third objective performance measure is the ability of randomization to diffuse the quantization levels in the output signal. Specifically, the aim is that the distribution of the output is transformed from a sparse distribution to something which resembles the input distribution. We have already found that application of random permutations and block-matrix rotations is an effective combination, whereby our aim is to evaluate how many such pairs we have to apply to get proper diffusion. To that end we define

$$P_M = \prod_{k=1}^{M} B_k P_k, \tag{9}$$

where B_k and P_k are random block-matrix rotations and permutations, respectively. Fig. 5 illustrates the output histogram when applying sign-quantization and the inverse rotation P_M^T for different values of M. We have here chosen to use sign-quantization since it is the worst-case in terms of sparsity.

We observe in the figure that the original quantized signal has a sparse distribution, where all samples are ± 1, but each consecutive randomization makes it resemble more the normalized Gaussian distribution (dashed line). Note that the normalized Gaussian is here the reference distribution, since the normalized covariance has a limited range (see Appendix for details). At $M = 4$ iterations the histogram has already converged to a unimodal distribution and at $M = 16$ the

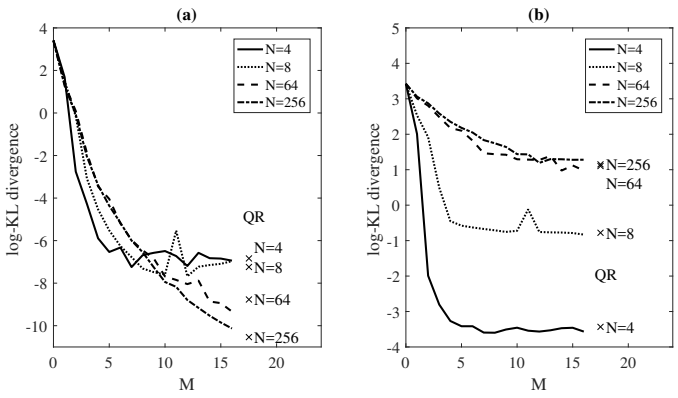

Fig. 6. Convergence of distribution with increasing number of rotations M to (a) the normalized Gaussian and (b) the normalized Laplacian, as measured by the Kullback-Leibler divergence, for different vector lengths N. As a reference, randomization with the QR-algorithm is depicted with crosses '×', representing a high-complexity and high-performance target level.

histogram is very close to the the normalized Gaussian distribution. The rate of convergence depends, however, on the vector length N.

As a final test of statistical properties, we therefore test the rate of convergence to the normalized Gaussian distribution with increasing number of iterations and different vector lengths N. As a measure of convergence, we use the Kullback-Leibler divergence between the normalized Gaussian distribution and the histogram of the output. As reference, we used randomization based on the QR-algorithm. The results are illustrated in Fig. 6(a).

We can see that convergence is faster for the shorter vectors, as is to be expected, since in a large space we need more rotations to reach all possible dimensions. The performance of the QR algorithm is illustrated with crosses '×' and we can see that after 16 iterations, for all vector lengths N, the proposed randomizers have more or less reached the diffusion of the QR algorithm. In fact, for $N = 4$ and $N = 8$, the performance saturates already after 5 and 7 iterations, respectively.

It is however clear that speech signals are not normally distributed, but we can often assume that spectral components follow the Laplace distribution [50], illustrated by the gray line in Fig. 5. Adding further rotations will not reduce the distance to a normalized Laplacian distribution, but it will saturate at some point, as is illustrated in Fig. 6(b). The divergence between the obtained histograms and the target distribution levels off after a few iterations. Moreover, when applying noise attenuation at the decoder, such as Wiener filtering, the distribution will be further modified. We therefore conclude that as few as $M = 4$ iterations should be sufficient to diffuse the quantization levels of a speech signal.

B. Application in a Speech and Audio Codec

To evaluate the performance of randomization in a practical application, we implemented the proposed speech and audio codec, whose generic structure was described in Sec. III as follows. As mentioned before, the LPC-based perceptual model was copied as-is from EVS [2]. The perceptually weighted frequency representation was then quantized with randomization using the QR-algorithm, the proposed low-complexity randomization (Eq. 9) with $M = 4$ iterations, as well as without randomization.

Method	Perceptual SNR (dB) Single node	Two nodes
None	4.08	4.08
QR	8.79	11.67
Proposed	6.42	8.02

TABLE II
PERCEPTUAL SNRS OF EACH EVALUATED METHOD.

For the randomized signals we used an assumption of Gaussian distribution and for the signal without randomization, we used the Laplacian distribution. Our previous experiments have shown that the Laplacian works best for speech signals [48], [50]. Above we have, however, shown that randomized signals are closer to Gaussian, whereby the choice of distributions is well-warranted. Informal experiments confirmed these choices as the best in terms of perceptual SNR. Here, the perceptual SNR refers to the signal to noise ratio between the perceptually weighted original and quantized signals [22].

As test-material, we randomly chose 6 samples (3 male and 3 female) from the TIMIT corpus [51]. For each coded sample, we calculated the window-wise perceptual SNR in decibel, and calculated the mean perceptual SNRs of respective methods, which are listed in Table II.

Even though all methods use entropy coding with the same bitrate, we get rather large differences in perceptual SNR. The QR method is over 4.8 dB better than no randomization, and the proposed low-complexity falls in between the two. Informal experiments show that an increase in the number of iterations used for creating the proposed randomization, will improve the SNR. The number of iterations is therefore directly proportional to complexity and SNR. It should be noted, however, that we have here not applied source modeling explicitly (such as that in [48]), which would most likely increase the performance of all methods, but especially the version without randomization. The obtained results should therefore be treated as provisional results until a source model has been implemented (length restrictions prevents us from including a discussion about source models in this paper).

To evaluate the perceptual influence of the randomization on the quantization noise, we conducted a MUSHRA listening test [52]. Thirteen subjects, aged between 22 and 53, were asked to evaluate the quality of the different approaches. Seven of the thirteen test persons referred to themselves as expert listeners. As test items we used the same six sentences of the TIMIT database as above.

For each item we used five conditions: no randomization, the proposed fast randomization approach and as an upper bound, randomization using the QR approach, as well as a 3.5 kHz low pass signal as a lower anchor, and the hidden reference, in accordance with the MUSHRA standard.

The results of the listening test are presented in Figure 7. The results show that there is a clear trend that randomization improves the perceived quality, as both the QR and the fast randomization approach are rated higher than the approach without randomization. Moreover, with the exception of item 3, the QR approach has higher scores than the proposed

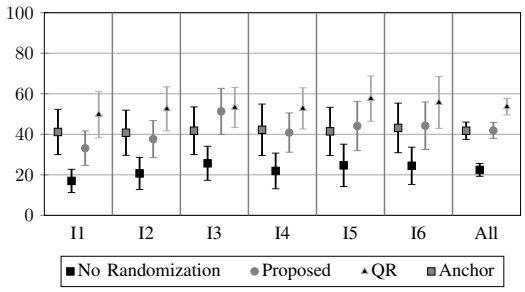

Fig. 7. The results of the MUSHRA test, given for the different items, and an average over all items. The reference was omitted as it was always rated to 100.

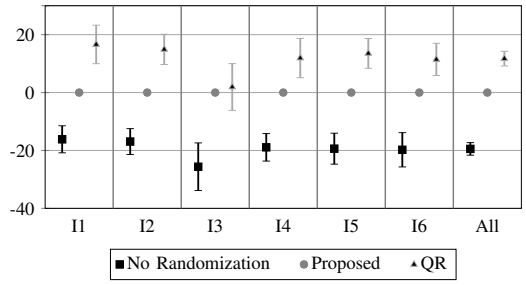

Fig. 8. The difference scores of the performed MUSHRA test, where the proposed approach was used as a reference point. The lower anchor and the hidden reference were omitted.

method. The coding quality of all methods is in the same range as the anchor, which is arguably low, even for a low bitrate codec. However, since the experiments did not include proper source modeling, the perceptual results overall should be treated as preliminary. In any case, for all items combined, the 95% confidence intervals do not overlap, and there is a clear difference between all three conditions under test, where the proposed approach performs on average about 20 MUSHRA points better than the conventional, and the QR approach can improve the quality by approximately 15 points.

To determine whether there is a statistically significant difference between the ratings of the proposed approach (Median = 42.5) and the lower anchor (Median = 40), hypotheses testing was applied. Since a Shapiro-Wilk test of the score differences ($W = 0.917$, $p < 0.01$) as well a visual inspection of Q-Q-plots indicated non-normally distributed data, a Wilcoxon signed rank test was performed which indicated no significant difference ($V = 1469$, $p = 0.87$) between the anchor and the proposed approach. However, it is unclear whether a comparison between the proposed method and lower anchor is relevant anyway, since the characteristics of the distortions in the two cases are very different, rendering a comparison difficult, and since we have not yet included a source model, whereby the absolute quality level was rather arbitrarily chosen. The anchor thus serves only as way to roughly characterize the absolute quality level used in the experiment.

The difference scores in Figure 8 support the findings of the above analysis. Taking the proposed approach as the reference point, the proposed approach performed always significantly better than the conventional. Moreover, with the exception of item 3, the QR approach performed always better than the proposed approach. It is unclear why QR does not have an advantage for item 3, but we suspect it is merely a statistical outlier. In any case, the low-complexity proposed method is always better than no randomization, which was our target. This argument also validates our choice of not using source modeling; by source modeling, we can improve quantization accuracy, but our experiments show that the perceptual quality of a codec can be improved by randomization even with a fixed quantization accuracy.

Finally, to determine how well quantization errors are decorrelated, we applied the randomization methods on two independent encoders (without difference in delay and without background noises) and took the mean of the outputs. In theory, taking the mean of two signals with uncorrelated noises should increase SNR by 3 dB. From Table II we see that randomization with the QR-algorithm almost reaches this target level, with an improvement of 2.88 dB. The proposed low-complexity randomizer achieves an improvement of 1.6 dB. It is thus again a compromise between complexity and quality, as the higher-complexity QR-method gives better SNR than the proposed low-complexity randomizer. In a real-life scenario we expect to see higher numbers, since any differences in acoustic delay and background/sensor noises would further contribute to decorrelate the quantization errors.

V. Conclusion

Quality of speech and audio coding can be improved in terms of both signal quality and ease of interaction with the user-interface by including, in the coding process, all connected hardware which feature a microphone. For this purpose, we have recently proposed a distributed speech and audio codec design based on randomization of the signal before quantization [8]. This paper addresses the complexity bottle-neck of the proposed codec, that is, the randomizer.

The proposed low-complexity randomizer is based on a sequence of random permutations and 2×2 block-rotations. Our experiments show that by successive randomizations we obtain high-accuracy decorrelation, such that the covariance of the original and the randomized signal behaves like uncorrelated signals, and such that the quantization levels of the output signal are diffused.

The proposed randomization has multiple benefits for low-bitrate coding, distributed coding, perceptual performance, robustness and encryption. Our experiments confirm these benefits by showing that randomization improves perceptual SNR and subjective quality. Though inclusion of a randomizer shows here an SNR improvement of 2.4 dB, we expect this benefit to be reduced when a proper source model is included. However, we show that if quantization errors are randomized, taking the mean of signals improves SNR as much as 2.8 dB, whereby we can always improve quality by adding more microphones.

The algorithmic complexity of the proposed randomizer is $\mathcal{O}(N \log N)$, where the main complexity is due to generation of random permutations. If the permutations are generated off-line the overall complexity is $\mathcal{O}(MN)$, where M is the

number of iterations. Typically $M = 4$ is sufficient, whereby complexity is $\mathcal{O}(N)$. Storage requirements are in all cases $\mathcal{O}(N)$. We believe that the complexity of the encoder therefore becomes viable even on low-performance nodes, such as wearable devices. Generation of the randomizer requires a sequence of pseudo-random numbers. We assume that the seed of the pseudo-random number generator is either known at both the encoder and decoder or seldom communicated as side-info.

Overall, with this work, the distributed speech and audio codec takes a large step forward to become a full system. Only source modeling was here omitted from a full system, due to space constraints. The randomizer is, however, a necessary and important part of the overall design, whereby finding a low-complexity solution was crucial.

APPENDIX
THE DISTRIBUTION OF NORMALIZED GENERALIZED GAUSSIANS

When a signal which follows a Gaussian or Laplacian distribution is normalized by its norm, its range becomes limited. Consequently, normalization of a signal changes its distribution and the purpose of this appendix is to determine the form of such distributions. In interest of generality, we study the generalized normal distribution, which includes both the Gaussian and Laplacian distributions as special cases.

Suppose x is an $N \times 1$ vector whose entries x_k are uncorrelated and follow the generalized normal distribution with zero mean and equal variance σ^2:

$$f(x_k) = \frac{p}{2b\Gamma(1/p)} \exp\left(-\left|\frac{x_k}{b}\right|^p\right), \tag{10}$$

where the scaling factor is $b = \sigma^2\sqrt{\frac{\Gamma(1/p)}{\Gamma(3/p)}}$ and where $\Gamma(\cdot)$ is the Gamma function [53].

By normalizing x with its $\mathcal{L}_p$-norm $\|x\|_p$, we obtain a new random variable $y = \frac{x}{\|x\|_p}$, which is closely related to x but does not follow the generalized normal distribution. In particular, in difference to x, the entries y_k of y have a limited range

$$\sum_{k=1}^{N} |y_k|^p = 1, \qquad \text{whereby} \qquad y_k \in [-1, +1]. \tag{11}$$

To derive the marginal distributions of y_k, we begin by studying the entries x_k of x. Let $\gamma_k = 2\left|\frac{x_k}{b}\right|^p$, whereby we can find the distribution of γ_k by substitution of variables

$$f(\gamma_k) = 2f(x_k)\frac{dx_k}{d\gamma_k} = \frac{\gamma^{\frac{1}{p}-1}e^{-\frac{\gamma_k}{2}}}{b\Gamma(1/p)} \sim \chi^2\left(\frac{2}{p}\right). \tag{12}$$

In other words, γ_k follows the Chi-squared distribution with $\frac{2}{p}$ degrees of freedom. We can then define

$$\lambda_k := |y_k|^p = \frac{|x_k|^p}{\|x\|_p^p} = \frac{|x_k|^p}{|x_k|^p + \sum_{h\neq k}|x_h|^p}. \tag{13}$$

Since the x_k's follow the generalized normal distribution, then $|x_k|^p$ and $\sum_{h\neq k}|x_h|^p$ will follow the Chi-squared distribution with $\frac{2}{p}$ and $(N-1)\frac{2}{p}$ degrees of freedom, respectively. Ratios such as λ_k of Chi-squared distributed variables will follow the Beta-distribution with parameters $\alpha = \frac{1}{p}$ and $\beta = \frac{N-1}{p}$, or specifically [54, Sec. 4.2]

$$f(\lambda_k) = \frac{\Gamma\left(\frac{N}{p}\right)}{\Gamma\left(\frac{1}{p}\right)\Gamma\left(\frac{N-1}{p}\right)}\lambda_k^{\frac{1}{p}-1}(1-\lambda_k)^{\frac{N-1}{p}-1}. \tag{14}$$

Moreover, from Eq. 11 it follows that

$$\sum_{k=1}^{N}\lambda_k = 1, \qquad \text{and} \qquad 0 \leq \lambda_k \leq 1. \tag{15}$$

The joint distribution of the λ_k's therefore follows the Dirichlet distribution with $\alpha_k = \frac{1}{p}$ [55].

We can then substitute $\lambda_k = |y_k|^p$ to get the distribution of y_k as

$$f(y_k) = \frac{1}{2}f(\lambda_k)\frac{d\lambda_k}{dy_k} = \frac{\Gamma\left(\frac{N}{p}\right)(1-|y_k|^p)^{\frac{N-1}{p}-1}}{2\,\Gamma\left(1+\frac{1}{p}\right)\Gamma\left(\frac{N-1}{p}\right)}. \tag{16}$$

This is the marginal distribution of the normalized Gaussian y_k for any k. We can readily see that it is a scaled and translated version of a symmetric Beta distribution. Note, however, that the entries y_k are correlated with each other due to Eq. 15. The distribution is symmetric around zero, whereby the mean is zero $E[y_k] = 0$ and the variance is (omitting subscripts for brevity)

$$E\left[|y|^2\right] = \int_{-1}^{1} f(y)\,|y|^2 dy_k = \frac{\Gamma\left(\frac{N}{p}\right)\Gamma\left(\frac{3}{p}\right)}{\Gamma\left(\frac{1}{p}\right)\Gamma\left(\frac{N+2}{p}\right)}, \tag{17}$$

where we used the substitution $\lambda = |y|^p$ and recognized that the integrand is similar to the Beta-distribution, whereby solution is simple. In particular, we have the variances

$$E[|y|^2] = \begin{cases} \frac{1}{N} & \text{for } p = 2 \\ \frac{2}{(N+1)N} & \text{for } p = 1. \end{cases} \tag{18}$$

ACKNOWLEDGMENTS

We want to thank Alexander Adami for the fruitful discussions and Nils Werner for sharing his expertise in setting up a web server for the MUSHRA experiments [56].

This project was supported by the Academy of Finland research project No 28467.

REFERENCES

[1] T. Bäckström, *Speech Coding with Code-Excited Linear Prediction*. Springer, 2017.
[2] *TS 26.445, EVS Codec Detailed Algorithmic Description; 3GPP Technical Specification (Release 12)*, 3GPP, 2014.
[3] *TS 26.190, Adaptive Multi-Rate (AMR-WB) speech codec*, 3GPP, 2007.
[4] ISO/IEC 23003–3:2012, "MPEG-D (MPEG audio technologies), Part 3: Unified speech and audio coding," 2012.
[5] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, and M. Dietz, "ISO/IEC MPEG-2 advanced audio coding," *Journal of the Audio engineering society*, vol. 45, no. 10, pp. 789–814, 1997.
[6] J. Benesty, M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. Springer, 2008.
[7] A. Zahedi, J. Østergaard, S. H. Jensen, S. Bech, and P. Naylor, "Audio coding in wireless acoustic sensor networks," *Signal Processing*, vol. 107, pp. 141–152, 2015.

[8] T. Bäckström, F. Ghido, and J. Fischer, "Blind recovery of perceptual models in distributed speech and audio coding," in *Proc. Interspeech*, 2016.
[9] T. Bäckström and J. Fischer, "Coding of paremetric models with randomized quantization in a distributed speech and audio codec," in *12 ITG Fachtagung Sprachkommunikation*, 2016.
[10] P. T. Boufounos and R. G. Baraniuk, "1-bit compressive sensing," in *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*. IEEE, 2008, pp. 16–21.
[11] A. Magnani, A. Ghosh, and R. M. Gray, "Optimal one-bit quantization," in *Data Compression Conference*. IEEE, 2005, pp. 270–278.
[12] M. M. Truman, G. A. Davidson, M. C. Fellers, M. S. Vinton, M. A. Watson, and C. Q. Robinson, "Audio coding system using spectral hole filling," Nov. 4 2008, uS Patent 7,447,631.
[13] S. Vaudenay, "Decorrelation: a theory for block cipher security," *Journal of Cryptology*, vol. 16, no. 4, pp. 249–286, 2003.
[14] S. Saeednia, "How to make the Hill cipher secure," *Cryptologia*, vol. 24, no. 4, pp. 353–360, 2000.
[15] C.-C. Kuo and Wen-Thong, "Reduction of quantization error with adaptive Wiener filter in low bit rate coding," in *Signal Processing Conference, 2002 11th European*. IEEE, 2002, pp. 1–4.
[16] G. E. Øien and T. A. Ramstad, "On the role of Wiener filtering in quantization and DPCM," in *Proc. Norwegian Signal Processing Symposium and Workshop (NORSIG/IEEE)*, 2001.
[17] J. Rissanen and G. G. Langdon, "Arithmetic coding," *IBM Journal of research and development*, vol. 23, no. 2, pp. 149–162, 1979.
[18] J. D. Gibson and K. Sayood, "Lattice quantization," *Advances in electronics and electron physics*, vol. 72, pp. 259–330, 1988.
[19] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer, 1992.
[20] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Mag.*, vol. 21, no. 5, 2004.
[21] Z. Xiong, A. D. Liveris, and Y. Yang, "Distributed source coding," *Handbook on Array Processing and Sensor Networks*, pp. 609–643, 2009.
[22] M. Bosi and R. E. Goldberg, *Introduction to Digital Audio Coding and Standards*. Kluwer Academic Publishers, 2003.
[23] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," in *Communications and Vehicular Technology in the Benelux (SCVT), 2011 18th IEEE Symposium on*. IEEE, 2011, pp. 1–6.
[24] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Communications*, vol. 14, no. 6, 2007.
[25] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, 2005.
[26] F. de la Hucha Arce, M. Moonen, M. Verhelst, and A. Bertrand, "Adaptive quantization for multi-channel Wiener filter-based speech enhancement in wireless acoustic sensor networks," *Wireless Communications and Mobile Computing (Special issue on Wireless Acoustic Sensor Networks and Applications)*, 2017.
[27] A. Zahedi, J. Østergaard, S. H. Jensen, P. Naylor, and S. Bech, "Coding and enhancement in wireless acoustic sensor networks," in *Data Compression Conference (DCC), 2015*. IEEE, 2015, pp. 293–302.
[28] A. Majumdar, K. Ramchandran, and L. Kozintsev, "Distributed coding for wireless audio sensors," in *Applications of Signal Processing to Audio and Acoustics, IEEE Workshop on*, 2003, pp. 209–212.
[29] H. Dong, J. Lu, and Y. Sun, "Distributed audio coding in wireless sensor networks," in *Computational Intelligence and Security, 2006 International Conference on*, vol. 2. IEEE, 2006, pp. 1695–1699.
[30] G. Barriac, R. Mudumbai, and U. Madhow, "Distributed beamforming for information transfer in sensor networks," in *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*. IEEE, 2004, pp. 81–88.
[31] R. Lienhart, I. Kozintsev, S. Wehr, and M. Yeung, "On the importance of exact synchronization for distributed audio signal processing," in *Proc. ICASSP*, vol. 4. IEEE, 2003, pp. IV–840.
[32] O. Roy and M. Vetterli, "Rate-constrained collaborative noise reduction for wireless hearing aids," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 645–657, 2009.
[33] S. Bray and G. Tzanetakis, "Distributed audio feature extraction for music," in *ISMIR*, 2005, pp. 434–437.
[34] N. Rajput and A. A. Nanavati, "Distributed speech recognition," *Speech in Mobile and Pervasive Environments*, pp. 99–114, 2012.
[35] D. Pearce, "Distributed speech recognition standards," *Automatic Speech Recognition on Mobile Devices and over Communication Networks*, pp. 87–106, 2008.
[36] S. Korse, T. Jähnel, and T. Bäckström, "Entropy coding of spectral envelopes for speech and audio coding using distribution quantization," in *Proc. Interspeech*, 2016.
[37] S. Das, A. Craciun, T. Jähnel, and T. Bäckström, "Spectral envelope statistics for source modelling in speech enhancement," in *12 ITG Fachtagung Sprachkommunikation*, 2016.
[38] G. H. Golub and C. F. van Loan, *Matrix Computations*, 4th ed. John Hopkins University Press, 2004.
[39] V. Pulkki, "Spatial sound reproduction with directional audio coding," *J Audio Engineering Society*, vol. 55, no. 6, pp. 503–516, 2007.
[40] A. Edelman and N. R. Rao, "Random matrix theory," *Acta Numerica*, vol. 14, pp. 233–297, 2005.
[41] D. Knuth, *The Art of Computer Programming*. Addison-Wesley, 1998.
[42] D. E. Knuth, "Volume 2: Seminumerical algorithms," in *The Art of Computer Programming*, 3rd ed. Addison-Wesley, 2007.
[43] P. Diaconis and M. Shahshahani, "The subgroup algorithm for generating uniform random variables," *Probability in the Engineering and Informational Sciences*, vol. 1, no. 01, pp. 15–32, 1987.
[44] G. W. Stewart, "The efficient generation of random orthogonal matrices with an application to condition estimators," *SIAM Journal on Numerical Analysis*, vol. 17, no. 3, pp. 403–409, 1980.
[45] P. L'Ecuyer, "Pseudorandom number generators," *Encyclopedia of Quantitative Finance*, 2010.
[46] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, 1998.
[47] A. Jagannatam, "Mersenne twister–a pseudo random number generator and its variants," *George Mason University, Department of Electrical and Computer Engineering*, 2008.
[48] T. Bäckström and C. R. Helmrich, "Arithmetic coding of speech and audio spectra using TCX based on linear predictive spectral envelopes," in *Proc. ICASSP*, Apr. 2015, pp. 5127–5131.
[49] J. Fischer and T. Bäckström, "Wiener filtering in distributed speech and audio coding," *IEEE Signal Process. Lett.*, submitted to, 2017.
[50] T. Bäckström, "Estimation of the probability distribution of spectral fine structure in the speech source," in *Proc. Interspeech*, submitted, 2017.
[51] J. S. Garofolo, L. D. Consortium *et al.*, *TIMIT: acoustic-phonetic continuous speech corpus*. Linguistic Data Consortium, 1993.
[52] Recommendation BS.1534, *Method for the subjective assessment of intermediate quality levels of coding systems*, ITU-R, 2003.
[53] S. Nadarajah, "A generalized normal distribution," *Journal of Applied Statistics*, vol. 32, no. 7, pp. 685–694, 2005.
[54] C. Walck, *Handbook on statistical distributions for experimentalists*. University of Stockholm Internal Report SUF-PFY/96-01, 2007.
[55] A. Bela, A. Frigyik, and M. Gupta, "Introduction to the Dirichlet distribution and related processes," *Department of Electrical Engineering, University of Washington*, 2010.
[56] M. Schoeffler, F.-R. Stöter, B. Edler, and J. Herre, "Towards the next generation of web-based experiments: A case study assessing basic audio quality following the ITU-R recommendation BS. 1534 (MUSHRA)," in *1st Web audio conference*, Paris, France, 2015.

Tom Bäckström is a professor of practice at Aalto University, Department of Signal Processing and Acoustics, in Espoo, Finland, since 2016. Previously he was professor (2013-2016) and researcher (2008-2013) at the International Audio Laboratories Erlangen, which is a joint research institution of Fraunhofer IIS and the Friedrich-Alexander University (FAU) in Erlangen, Germany. He obtained his doctorate and master's degrees from Aalto University (formerly Helsinki University of Technology) in 2004 and 2001, respectively. His research is focused on speech signal processing, including the coding, analysis, enhancement and modeling of speech, as well as related mathematical methods and audio coding.

Johannes Fischer received his B.Sc. and M.Sc. in Electrical Engineering, Electronics, and Information Technology from the Friedrich-Alexander University (FAU) in Erlangen, Germany in 2010 and 2012, respectively. Currently he is a doctoral student at the International Audio Laboratories Erlangen, a joint research institution of Fraunhofer IIS and the FAU. His areas of interest include speech coding, enhancement and spatial filtering.