A CSI Compression Scheme Using Context Trees

Author(s):
Miyamoto, Henrique K.; Yang, Sheng

Publication date:
2022-03-02

Permanent link:
https://doi.org/10.3929/ethz-b-000535273

International Zurich Seminar on Information and Communication (IZS), March 2 – 4, 2022

A CSI Compression Scheme Using Context Trees

Henrique K. Miyamoto
University of Campinas (Unicamp)
Institute of Mathematics (IMECC)
Campinas, SP, Brazil
Email: hmiyamoto@ime.unicamp.br

Sheng Yang
CentraleSupélec, Paris-Saclay University
Laboratory of Signals and Systems (L2S)
Gif-sur-Yvette, France
Email: sheng.yang@centralesupelec.fr

***Abstract*—We propose novel compression algorithms for time-varying channel state information (CSI) in wireless communications. The proposed schemes combine (lossy) vector quantisation and (lossless) compression. The vector quantisation technique is based on data-adapted parametrised companders applied on each component of the normalised vector. Then, the sequences of quantisation indices are compressed according to estimated distributions computed with a context-tree approach. The algorithms have low complexity, are linear-time in spatial dimension and time duration, and can be implemented in an online fashion. We run numerical experiments to demonstrate the effectiveness of the proposed algorithms in such scenarios.**

I. Introduction

In wireless communication systems, efficiently representing the channel state information (CSI) is crucial for storage and dissemination. Typically, in the downlink transmission from a base station (BS) with multiple antennas to multiple users, beamforming techniques rely on precise CSI at the transmitter side [1]. For the BS to acquire the CSI, however, it usually requires that each user feeds back the CSI measurements in a timely and accurate fashion. How to reduce the bandwidth cost of such feedback traffic, which is highly non-negligible, is becoming a crucial problem. This is essentially a lossy data compression problem.

The spatial correlation inherent to the antenna structures has been exploited to reduce CSI dimension in recent works using deep learning and compressed sensing techniques (see, e.g., [2], [3] and the references therein), while the temporal correlation of CSI measurements is less exploited for feedback. Indeed, if the sequence of the quantised symbols is stationary, it can be losslessly compressed up to the entropy rate of the underlying process. A possible approach for CSI compression is therefore to apply any universal compression algorithm [4], [5], such as Lempel-Ziv [6], [7] (known as LZ77 and LZ78), to the quantisation indices.

Another universal compressor is the *context-tree weighting* (CTW) algorithm [8], which learns the distribution of a given sequence in an efficient way. This distribution can then be used to compress the sequence in combination with arithmetic coding, achieving the Rissanen lower bound [8]. A modification of CTW yields the *context-tree maximising* (CTM) algorithm [9], which can produce the maximum *a posteriori* (MAP) probability tree model. Connections with Bayesian inference have been explored in [4], [10].

However, directly applying these algorithms to compress quantisation indices in an online fashion presents some difficulties. First, the output bit-stream is of variable length, making the feedback difficult to implement. Second, in Lempel-Ziv methods, the input symbol block is also of variable length, as it depends on parsing the original sequence. Finally, arithmetic coding has to be carefully implemented so as to deal with digital computers finite precision constraints [11]. Trying to avoid such difficulties motivates us to propose new compression algorithms adapted to applications such as the communication scenarios considered here.

In this work, we focus on the problem of online lossy compression of a sequence of CSI vectors and propose a two-step compression procedure. First, a new vector quantisation technique, based on a class of parametrised companders, is applied on the components of the normalised vector. The quantisation is composed of a non-linear transformation, followed by a uniform quantiser. The companders can be designed and updated with available empirical data. In particular, we consider the widely used μ-law compander and a new one, the β-law compander, inspired by the beta distribution. Then, we compress the sequence of quantisation indices using a context-tree-based approach. We propose two solutions: 1) to directly apply CTW with arithmetic coding, or 2) to apply CTM to estimate the conditional distribution of the upcoming symbol at each time instant and use this probability to compress the symbol. In the latter case, we encode each symbol with a fixed number of levels to limit the fluctuation of the encoded bit-flow—a desirable property in communication systems. In addition, the algorithms have low complexity, are linear-time in both the spatial dimension and time duration, and can be implemented in an online fashion.

This paper is organised as follows. In Section II we present the system model and review basic concepts of vector quantisation and context-tree representation. Our CSI compression algorithm is described in Section III. The simulation of CSI acquisition is analysed in Section IV, followed by some conclusions. Due to the space limitation, we omit some important details that can be found in the extended version in [12], where implementation codes are also available.

Notation: Vectors ($\boldsymbol{v}$) are denoted by bold italic lower-case letters. Random variables (X) are in non-italic upper-case. L_2 vector norms are denoted by $\|\boldsymbol{v}\|$. Logarithms are to the base 2. We denote $[n] \coloneqq \{1, \dots, n\}$.

II. Problem Formulation and Preliminaries

A. Main Problem

We consider a network composed of a transmitter (e.g., base station) and N_r receivers (e.g., mobile users). Assume that the CSI between the transmitter and receiver k at time t can be described by a complex vector $\boldsymbol{h}_k[t] \in \mathbb{C}^{N_\mathrm{t}\times 1}$, for $k \in [N_\mathrm{r}]$. For different purposes (e.g., feedback, storage), each receiver is required to represent its state sequence using as few bits as possible, for a given distortion constraint. This is known as the lossy source coding problem [5]. In most practical scenarios, the norm of the vectors $\boldsymbol{h}_k[t]$ is less important than the direction. Therefore, our goal here is to compress the normalised vector $\boldsymbol{h}_k[t]/\|\boldsymbol{h}_k[t]\|$.

B. Vector Quantisation

A *vector quantiser* [13] of dimension p and size M, is a mapping $q : \mathbb{R}^p \to \mathcal{C}$, with $\mathcal{C} \coloneqq \{\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{M-1}\} \subset \mathbb{R}^p$, that associates each vector $\boldsymbol{x} \in \mathbb{R}^p$ to a codeword $\hat{\boldsymbol{x}} \coloneqq q(\boldsymbol{x}) = \boldsymbol{y}_k$, for some $k \in \{0, 1, \ldots, M-1\}$. For a sequence of vector symbols $\boldsymbol{x}_1^n \coloneqq \boldsymbol{x}_1\boldsymbol{x}_2\cdots\boldsymbol{x}_n$, we can apply vector-by-vector quantisation, generating a sequence of quantised vectors $\hat{\boldsymbol{x}}_1^n \coloneqq \hat{\boldsymbol{x}}_1\hat{\boldsymbol{x}}_2\cdots\hat{\boldsymbol{x}}_n$ and a sequence of quantisation indices $k_1^n \coloneqq k_1k_2\cdots k_n$, where $\hat{\boldsymbol{x}}_i = \boldsymbol{y}_{k_i}$, for each $i \in [n]$.

Two important parameters to assess the performance of a vector quantiser are the quantisation rate and the mean distortion. The *quantisation rate*, defined as $R \coloneqq (\log M)/p$, is an indicator of the cost to describe the vector, while the *mean distortion* measures the error induced by the quantisation. We use, as distortion measure between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, the mean squared chordal distance (MSCD), defined as

$$\mathrm{MSCD}(\boldsymbol{x}, \hat{\boldsymbol{x}}) \coloneqq 1 - \mathbb{E}\left[\frac{|\langle \boldsymbol{x}, \hat{\boldsymbol{x}}\rangle|^2}{\|\boldsymbol{x}\|^2\|\hat{\boldsymbol{x}}\|^2}\right]. \tag{1}$$

C. Variable-Order Markov Chain and Context Tree

Let $x_i^j \coloneqq x_ix_{i+1}\cdots x_j$ be a scalar sequence over an alphabet $\mathcal{A} \coloneqq \{0, 1, \ldots, m-1\}$, generated by a source with probability distribution P. We denote $l(x_i^j) \coloneqq j - i + 1$ the length of sequence x_i^j. A *variable-order Markov chain* with order or memory D (also called *bounded memory tree source*) is a random process for which $P(x_i|x_{-\infty}^{i-1}) = P(x_i|x_{i-D}^{i-1})$. Our interest in Markov chains comes from the fact that any stationary ergodic source can be approximated by a Markov chain with sufficiently large order [4], [5].

The statistical behaviour of a variable-order Markov chain is described by a *context set* $\mathcal{S}$ (also known as *suffix set* or *model*), which is defined as a subset of $\bigcup_{i=0}^{D}\mathcal{A}^i$ that is proper (i.e., no element in $\mathcal{S}$ is a proper suffix of any other) and complete (i.e., each $x_{-\infty}^n$ has a suffix in $\mathcal{S}$, which is unique by properness). The *context function* $c : \mathcal{A}^D \to \mathcal{S}$ maps each length-D context x_{i-D}^{i-1} to a suffix $c(x_{-\infty}^{i-1}) = c(x_{i-D}^{i-1}) = x_{i-j}^{i-1}$, $j \le D$. Furthermore, each suffix $s \in \mathcal{S}$ is associated with a parameter $\boldsymbol{\theta}_s \coloneqq (\theta_s(0), \theta_s(1), \ldots, \theta_s(m-1))$, where $\theta_s(j) \coloneqq P(j|s)$. The *parameter vector* $\Theta \coloneqq \{\boldsymbol{\theta}_s : s \in \mathcal{S}\}$ groups all parameters in the context set $\mathcal{S}$. Therefore, the Markov chain is completely characterised by the couple $(\mathcal{S}, \Theta)$. We use $\mathcal{C}_D$ to denote the class of all context sets of order up to D. Finally, we define $L_D(\mathcal{S}) \coloneqq |\{s \in \mathcal{S} : \ l(s) = D\}|$ the number of contexts with length D.

Since the context set $\mathcal{S}$ is proper, its elements can be represented as leaf nodes of a tree $\mathcal{T}_D$, called *context tree*, i.e., $\mathcal{S} \subseteq \mathcal{T}_D$. For a given sequence x_1^n, each leaf node $s \in \mathcal{S}$ is associated with a *counter* $\boldsymbol{a}_s \coloneqq \boldsymbol{a}_s(x_1^n) \coloneqq (a_s(0), a_s(1), \ldots, a_s(m-1))$, where $a_s(j)$ stores the number of times that symbol $j \in \mathcal{A}$ follows context s in x_1^n. The counter of each inner node of the tree is recursively defined as the sum of the counters of its children nodes, i.e., $\boldsymbol{a}_s \coloneqq \sum_{j\in\mathcal{A}}\boldsymbol{a}_{js}$, $\forall\, s \in \mathcal{T}_D \setminus \mathcal{S}$. In particular, we use the empty string λ to denote the root of the tree.

With the above definitions and the Markov property for a D-th order Markov chain, if both $\mathcal{S}$ and Θ are known, the probability of a sequence can be written as [10]

$$P(x_1^n|x_{D-1}^0, \mathcal{S}, \Theta) = \prod_{s\in\mathcal{S}}\prod_{j\in\mathcal{A}}\theta_s(j)^{a_s(j)}. \tag{2}$$

If only the model $\mathcal{S}$ is known, but not its parameters Θ, the *marginal distribution* of a sequence x_1^n, given its past x_{1-D}^0 and model $\mathcal{S}$, is

$$P(x_1^n|x_{1-D}^0, \mathcal{S}) = \int P(x_1^n|x_{1-D}^0, \mathcal{S}, \Theta)\pi(\Theta|\mathcal{S})\,\mathrm{d}\Theta, \tag{3}$$

assuming the distribution of the parameters, $\pi(\Theta|\mathcal{S})$, is known. While this distribution is unknown in general, using the so-called *Jeffrey's prior* is asymptotically optimal in the minimax sense [4]. This choice corresponds to setting $\pi(\Theta|\mathcal{S})$ to be the Dirichlet distribution with parameters $\left(\frac{1}{2}, \cdots, \frac{1}{2}\right)$. In this case, the distribution (3) can be simplified to the so-called *Krichevsky–Trofimov (KT) distribution*, which can be easily computed as

$$P(x_1^n|x_{1-D}^0, \mathcal{S}) = \prod_{s\in\mathcal{S}} P_e(\boldsymbol{a}_s), \tag{4}$$

where

$$P_e(\boldsymbol{a}_s) = \frac{\prod_{j=0}^{m-1}\left(\frac{1}{2}\right)\left(\frac{3}{2}\right)\cdots\left(a_s(j)-\frac{1}{2}\right)}{\left(\frac{m}{2}\right)\left(\frac{m}{2}+1\right)\cdots\left(\frac{m}{2}+M_s-1\right)}, \quad s \in \mathcal{T}_D, \tag{5}$$

with $M_s \coloneqq \sum_{j=0}^{m-1} a_s(j)$.

Finally, if the model $\mathcal{S}$ is also unknown, then we shall marginalise over $\mathcal{S}$ with a given prior distribution π_D on all models $\mathcal{S}$ of maximal depth D. Fixing $\gamma \in]0, 1[$ and

$$\pi_D(\mathcal{S}) \coloneqq (1-\gamma)^{\frac{|\mathcal{S}|-1}{m-1}}\gamma^{|\mathcal{S}|-L_D(\mathcal{S})}, \tag{6}$$

we obtain a mixture of different distributions (4), corresponding to the coding distribution of CTW [4], [10]:

$$Q_n(x_1^n|x_{1-D}^0) \coloneqq \sum_{\mathcal{S}\in\mathcal{C}_D}\pi_D(\mathcal{S})\prod_{s\in\mathcal{S}}P_e(\boldsymbol{a}_s). \tag{7}$$

Not only is this coding distribution universal for the class of stationary ergodic sources (i.e., it asymptotically achieves optimal coding rate irrespective of the source distribution), but also it can be recursively computed so that the complexity is linear in n [8].

The CTM algorithm [9] comes from a modification of the CTW algorithm and can be used to compute the maximum *a posteriori* model $\mathcal{S}$ for a given sequence x_{1-D}^n.

Definition 1. For $\gamma \in]0,1[$, the *maximised probability* P_m^s of each node $s \in \mathcal{T}_D$ with length $d = l(s)$ is

$$P_m^s \coloneqq \begin{cases} \max\{\gamma P_e(\boldsymbol{a}_s), (1-\gamma)\prod_{j=0}^{m-1} P_m^{js}\}, & 0 \le d < D, \\ P_e(\boldsymbol{a}_s), & d = D, \end{cases} \tag{8}$$

and the *maximising tree* $\mathcal{S}_m^s$ is obtained by pruning the descendants of the nodes s where the maximum is achieved by the first term.

Lemma 1 (See [10])**.** *The maximised coding distribution P_m^λ of the root node $\lambda \in \mathcal{T}_D$ satisfies*

$$P_m^\lambda = \max_{\mathcal{S}\in\mathcal{C}_D} \pi_D(\mathcal{S}) \prod_{s\in\mathcal{S}} P_e(\boldsymbol{a}_s). \tag{9}$$

We find then that the maximising tree $\mathcal{S}_m^\lambda$, which is associated to the maximised probability P_m^λ, corresponds to the maximum *a posteriori* model:

$$\mathcal{S}_m^\lambda = \arg\max_{\mathcal{S}\in\mathcal{C}_D} P(\mathcal{S}|x) = \arg\max_{\mathcal{S}\in\mathcal{C}_D} \pi_D(\mathcal{S}) \prod_{s\in\mathcal{S}} P_e(\boldsymbol{a}_s). \tag{10}$$

III. Proposed Scheme

A. Quantisation

The vector quantisation that we propose consists in vector normalisation, decomposition into real components, and individual scalar quantisation based on parametric companders.

1) Vector Normalisation: In this step, the input vector $\boldsymbol{x} = [x(1) \cdots x(N_\mathrm{t})]$ is normalised by the component with the largest absolute value, i.e., $\bar{\boldsymbol{x}} \coloneqq \boldsymbol{x}/x(i^*)$ where $i^* \coloneqq \arg\max_i |x(i)|$. Note that $\bar{x}(i^*) = 1$, while the other normalised components are complex in general, with absolute value in $[0,1]$. The i^*-th component can skip the following steps and be directly assigned a special quantisation index indicating it as the strongest component.

2) Decomposition: Before quantisation, each complex component should be decomposed into real values. We consider the polar decomposition into amplitude and phase, since these components are usually less correlated in wireless applications, thus providing a less 'redundant' representation.

3) Quantisation with Parametric Companders: The amplitude and phase are quantised separately with different scalar quantisers of M_abs and M_ang quantisation levels, respectively.

If the input is uniformly distributed, then a uniform quantiser is optimal. In general, however, uniform quantisation can be far from optimal in the rate-distortion sense [5]. Let X be a random variable representing the input, following some distribution P over the support interval $[0,1]$. The idea of using a compander is to apply a non-linear and non-decreasing mapping $g : [0,1] \to [0,1]$ to the signal (*compression*) before quantising it, so that the signal is more 'uniform' in the image space. To recover the signal, the inverse mapping $g^{-1} : [0,1] \to [0,1]$ is used (*expansion*). It is practical to use parametric companders, i.e., (differentiable) maps g that can be described by a few number of parameters. One of the widely used such companders is the μ-law compander, which is parametrised by a value $\mu > 0$. Note that, as compared to the Lloyd quantiser [5], compander-based quantisers have much lower complexity of quantisation and representation.

TABLE I
Two Compander Functions.

Compander	Parameters	pdf $g'(x)$
μ-law	$\mu > 0$	$\dfrac{\mu}{(1+\mu x)\ln(1+\mu)}$
β-law	$\alpha > 0,\ \beta > 0$	$\dfrac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$

In this work, we propose a data-driven design of a compander parametrised by some θ (which can contain multiple scalar parameters). Assume that we have a set of training data $x_1, \ldots, x_n$. Our design is a two-step procedure: 1) uniformisation of the data, and 2) adjustment of the compander parameter.

We assume that the training data are formed by independent samples from some distribution P. If we knew the cumulative distribution function (cdf) F_P of P, we could apply the mapping F_P so that $F_P(x_1), \ldots, F_P(x_n)$ are samples from a uniform distribution. If, however, we are restricted to a class of companders $\{g_\theta,\ \theta \in \mathcal{Q}\}$ for some set $\mathcal{Q}$, then we have to approximate F_P with g_θ. Since a compander as defined above is non-decreasing from 0 to 1, it is equivalent to a cdf. Thus, a sensible criterion for the approximation is through the Kullback-Leibler divergence:

$$\theta^* = \arg\min_{\theta\in\mathcal{Q}} D(P \,\|\, g_\theta) = \arg\max_{\theta\in\mathcal{Q}} \mathbb{E}_P[\log(g'_\theta(\mathrm{X}))]. \tag{11}$$

Remarkably, this is equivalent to maximising the differential entropy of $g_\theta(\mathrm{X})$. Since the uniform distribution maximises differential entropy among all bounded support distributions [5], the criterion (11) returns indeed the best 'uniformiser'. Note that, since g_θ is a cdf, g'_θ is the corresponding probability density function (pdf).

The true distribution of the data is, nevertheless, unknown in most practical scenarios. But we can adapt the probabilistic criterion (11) into a data-driven one by replacing the expectation with the sample mean:

$$\arg\max_{\theta\in\mathcal{Q}} \frac{1}{n}\sum_{i=1}^{n} \log\left(g'_\theta(x_i)\right). \tag{12}$$

In this paper, we consider the μ-law compander and another one that we call β-law compander, as shown in Table I. The β-law compander is equivalent to the beta cdf, parametrised by $\alpha > 0$ and $\beta > 0$. An attractive feature of the β-law compander is that its pdf is log-concave in (α, β) [14, Theorem 6], so that the maximisation (12) can be easily solved.

The first step (uniformising the input) is not enough in the sense of rate-distortion. We also need to adjust the parameter to balance the distortion generated in different intervals, which is the role of the second step. While the exact solution is hard

to find, we provide a heuristic, yet efficient, way to make the adjustment.

If we assume that the distortion generated in the interval i is proportional to the squared length Δ_i^2 of the interval, then the average distortion is proportional to $\sum_{i=0}^{M-1} N_i \Delta_i^2$ where N_i is the number of samples inside interval i. Starting with the solution given by step 1, all N_i's are comparable (since it is roughly uniform), and the largest interval contributes the most to the average distortion. Similarly, the smallest interval contributes the least. The idea is therefore to reduce the largest interval until

$$N_S \Delta_S^2 \geq N_L \Delta_L^2, \tag{13}$$

where 'S' and 'L' stand for the 'smallest' and 'largest' intervals, respectively.

Although the presented compander design is based on training data, we can also start with a uniform compander and update it regularly when more data are available. A great advantage of the parametric compander design is the negligible communication overhead of the (few) quantisation parameters.

Remark 1. *It is well known that, followed by entropic encoding, a uniform quantiser is asymptotically optimal in the high-rate regime. We emphasise, however, that here we do not operate in the high-rate regime unlike many other applications. More importantly, a large alphabet size would make the following context-tree-based compression highly inefficient. Hence, a carefully designed quantiser is crucial for the overall performance.*

After the quantisation is done, one has to compress the sequence of quantisation indices. One way to do that is to directly apply CTW with arithmetic coding to this sequence. In the following subsections, we describe an alternative solution that limits the fluctuation of the output bit-stream.

B. Tree Estimation

Given a scalar sequence k_1^n, we use the CTM algorithm (cf. Section II-C) to find the maximum *a posteriori* tree model $\hat{\mathcal{S}}$ that describes that sequence. This algorithm consists in building the same tree $\mathcal{T}_D$ as in CTW algorithm, followed by a pruning procedure as described in Definition 1. Both the computational and storage complexity of CTM algorithm are known to be $O(nmD)$, i.e., linear with sequence length n, alphabet size m and maximum tree depth D, cf. [10].

When training data are available, we can apply the CTM algorithm on the training data to estimate the MAP model $\hat{\mathcal{S}}$, and use it to estimate symbol probabilities and encode the incoming sequence. This, however, is not necessary: we could initialise the full tree $\mathcal{T}_D$ with empty counters, keep updating the counters with incoming data, and regularly prune a copy of this tree to have an updated estimate of the MAP model $\hat{\mathcal{S}}$.

C. Coding Distribution and Encoding

Once a tree model $\hat{\mathcal{S}}$ is estimated, we can encode a sequence k_1^n according to the probabilities issued from that model. Note that, given a model $\hat{\mathcal{S}}$ and past symbols k_{1-D}^0, the estimated probability of a sequence k_1^n can be computed via the KT estimator, using (4) and (5). In particular, denoting $s \coloneqq c(k_{i-D}^{i-1})$, we can compute the probabilities $\hat{P}(\cdot) = P(\cdot|\hat{\mathcal{S}})$ that the next symbol is $k_i = j$, for all $j \in \mathcal{A}$, as

$$\begin{aligned}\hat{P}(j|k_{i-D}^{i-1}) &= \frac{\hat{P}(k_{i-D}^{i})}{\hat{P}(k_{i-D}^{i-1})} = \frac{\prod_{s' \in \hat{\mathcal{S}}} P_e(\boldsymbol{a}_{s'}(k_1^i))}{\prod_{s' \in \hat{\mathcal{S}}} P_e(\boldsymbol{a}_{s'}(k_1^{i-1}))} \\ &= \frac{P_e(\boldsymbol{a}_s(k_1^i))}{P_e(\boldsymbol{a}_s(k_1^{i-1}))} = \frac{a_s(j) + \frac{1}{2}}{\frac{m}{2} + \sum_{j' \in \mathcal{A}} a_s(j')}. \end{aligned} \tag{14}$$

With $\hat{P}$, one may apply arithmetic coding to encode k_i. But the encoded bits would have a variable length depending on both $\hat{P}$ and k_i. Reducing the fluctuation of the coded bit length is important for practical communication systems. Here, we propose an encoding scheme with three possible codeword lengths, as described below.

Fix two integers $q_1, q_2 \leq \log m$ such that $m_1 \coloneqq 2^{q_1}$, $m_2 \coloneqq 2^{q_2}$, and $m_1 + m_2 \leq m$. If k_i is among the m_1 most probable symbols according to $\hat{P}$ (tie could be broken with a fixed rule), then the encoded bit string $\mathbf{c}_i$ is 0 followed by q_1 bits indicating the position of k_i in the list of the m_1 most probable symbols. Otherwise, if k_i is among the next m_2 most probable symbols, the encoded bit string $\mathbf{c}_i$ is 10 followed by q_2 bits indicating the position of k_i in the second list. Finally, if k_i is not among the $m_1 + m_2$ most probable symbols, the encoded bit string $\mathbf{c}_i$ is 11 followed by q_2 bits corresponding to the index $\tilde{k}_i$ from a lower resolution quantiser with size m_3. Hence, in our scheme, we also need to keep a lower resolution quantiser to apply on least probable symbols. It follows that the codeword length is either $1 + q_1$, $2 + q_2$ or $2 + \lceil \log m_3 \rceil$.

D. More Implementation Details

Some more implementation details are omitted and can be found in the long version in [12].

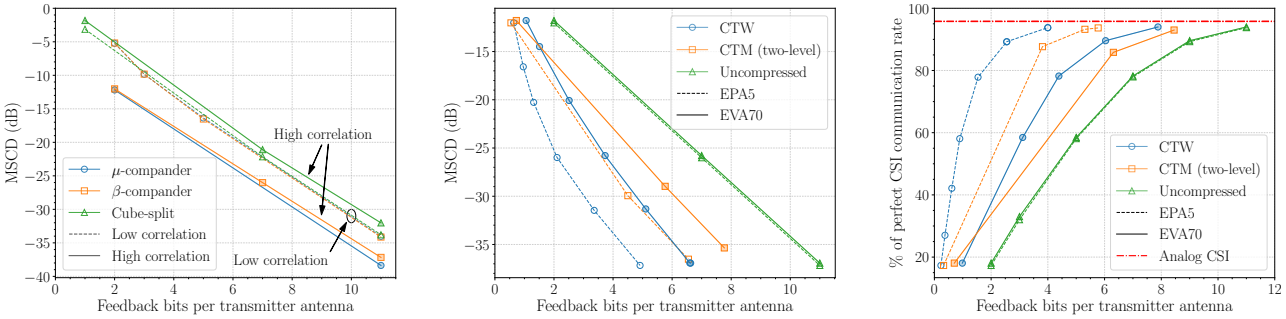
First, the bit allocation between the amplitude and phase quantisations can be optimised to minimise the overall distortion on the complex symbol. We can show that a rule of thumb is to use two more bits on the phase than on the amplitude.

Then, for practical uses, we have multiple trees, each one corresponding to a quantised sequence (amplitude or phase) of a given user and antenna. While each tree provides the marginal distribution of the given sequence, all the marginal distributions can be jointly used to encode the parallel streams together, in order to improve the coding rate.

IV. Simulation Results and Conclusions

We use the MATLAB LTE Toolbox to simulate an LTE MIMO downlink channel, with $N_\mathrm{t} = N_\mathrm{r} = 4$. We consider both the low mobility (EPA5, Doppler 5 Hz) and high mobility (EVA70, Doppler 70 Hz) scenarios, with either low or high correlation between antennas at the base station. In our implementations, we use $D = 2$, $\gamma = 0.5$ and $q_1 = 0$.

We consider three quantisation schemes: the μ-law compander, the β-law compander, and the cube-split quantiser [15]. Interestingly, the cube-split quantiser can be regarded as a complex compander adapted to the distribution of normalised complex Gaussian vectors. For each quantisation scheme, we



(a) MSCD distortion, EPA5. (b) MSCD distortion, high antenna correlation. (c) Communication rate, low antenna correlation.

Fig. 1. Simulation results.

consider three scenarios: no compression, compression with ideal CTW using arithmetic coding [8], and compression with the two-level resolution CTM scheme. The ideal CTW case is simply evaluated with $\frac{1}{n}\left(\lceil -\log Q_n(x_1^n|x_{1-D}^0)\rceil + 1\right)$.

In all cases, we assess the MSCD versus the average number of CSI bits per antenna, and, for low antenna correlation, we also assess the downlink communication sum rate with zero-forcing beamforming, evaluated approximately using the formula provided in [1, Eq. (20)], at 30 dB. The results are obtained with the best quantisation parameters (sizes of different codebooks) over those that we have tried.

Fig. 1a compares the performance of the different quantisers, with no compression, for low mobility scenario (EPA5). For low antenna correlation, the cube-split and the proposed quantisers achieve almost the same results. On the other hand, when antenna correlation is high, both proposed quantisers have similar performances and are noticeably better than the cube-split (which assumes uniformity of the distribution by design).

In Fig. 1b, we fix the β-law compander and study the performance of different compression methods, under high antenna correlation. The compression gains are significant and can reduce the CSI bits by up to half in the lower rate regime. For EPA5, in the higher rate regime, the two-level CTM scheme can reduce the feedback bits in 4.5 bits and is 1.5 bits away from the CTW performance, approximately. For EVA70, the gains are smaller, due to the lower time correlation. Nevertheless, in the higher rate regime, the two-level CTM can save more than 2 bits, and CTW, more than 4 bits. Furthermore, for EVA70 in the extreme low rate regime, the the two-level CTM outperforms CTW, thanks to the low-resolution quantiser.

Finally, Fig. 1c presents the communication rates for different compression schemes, using the β-law compander. The results are normalised by the rate achieved when perfect (i.e., noiseless) CSI knowledge is available. For both EPA5 and EVA70, we see that the communication rate converges much faster to the analog CSI rate (i.e., with no quantisation) when some of the proposed compression schemes are employed.

More importantly, the proposed schemes have low complexity, can be implemented in an online fashion, and are modular. In particular, the context-tree-based compression scheme can be applied on any other quantisers, including those recently designed with neural networks, e.g., [3]. Similarly, the proposed quantiser can be combined with any other lossless compression schemes.

References

[1] G. Caire, N. Jindal, M. Kobayashi, and N. Ravindran, "Multiuser MIMO achievable rates with downlink training and channel state feedback," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2845–2866, 2010.

[2] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Convolutional neural network-based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2827–2840, 2020.

[3] M. B. Mashhadi, Q. Yang, and D. Gündüz, "Distributed deep convolutional compression for massive MIMO CSI feedback," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2621–2633, 2021.

[4] E. Gassiat, *Universal Coding and Order Identification by Model Selection Methods*. Cham, Switzerland: Springer, 2018.

[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ, USA: Wiley, 2006.

[6] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inf. Theory*, vol. 23, no. 3, pp. 337–343, 1977.

[7] ——, "Compression of individual sequences via variable-rate coding," *IEEE Trans. Inf. Theory*, vol. 24, no. 5, pp. 530–536, 1978.

[8] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: basic properties," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 653–664, 1995.

[9] F. Willems, T. Tjalkens, and Y. Shtarkov, "Context-tree maximizing," in *Proc. 34th Annu. Conf. Inf. Sciences and Syst.*, Princeton, New Jersey, 2000, pp. TP6–7–TP6–12.

[10] I. Kontoyiannis, L. Mertzanis, A. Panotopoulou, I. Papageorgiou, and M. Skoularidou, "Bayesian context trees: Modelling and exact inference for discrete time series," *arXiv*, 2020. [Online]. Available: https://arxiv.org/pdf/2007.14900v1.pdf

[11] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, 1987.

[12] "Context-tree based CSI compression." [Online]. Available: https://miyamotohk.github.io/context-tree-compression

[13] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA, USA: Kluwer, 1992.

[14] S. S. Dragomir, R. P. Agarwal, and N. S. Barnett, "Inequalities for beta and gamma functions via some classical and new integral inequalities," *RGMIA Res. Rep. Collection*, vol. 2, no. 3, 1999.

[15] A. Decurninge and M. Guillaud, "Cube-split: Structured quantizers on the Grassmannian of lines," in *2017 IEEE Wireless Commun. and Netw. Conf. (WCNC)*, 2017, pp. 1–6.