Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 683030, 13 pages
doi:10.1155/2008/683030

Research Article

Feedback Quantization for Linear Precoded Spatial Multiplexing

Claude Simon and Geert Leus

Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

Correspondence should be addressed to Claude Simon, c.simon@tudelft.nl

Received 15 June 2007; Revised 19 October 2007; Accepted 8 January 2008

Recommended by David Gesbert

This paper gives an overview and a comparison of recent feedback quantization schemes for linear precoded spatial multiplexing systems. In addition, feedback compression methods are presented that exploit the time correlation of the channel. These methods can be roughly divided into two classes. The first class tries to minimize the data rate on the feedback link while keeping the performance constant. This class is novel and relies on entropy coding. The second class tries to optimize the performance while using the maximal data rate on the feedback link. This class is presented within the well-developed framework of finite-state vector quantization. Within this class, existing as well as novel methods are presented and compared.

Copyright © 2008 C. Simon and G. Leus. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

An attractive scheme to make spatial multiplexing more robust against rank deficient channels, and to reduce the receiver complexity, is linear precoding. The linear precoding matrix is a function of the channel state information (CSI), which is, in general, only available at the receiver. Thus, the required information to calculate the precoding matrix must be fed back to the transmitter over a feedback link, which is assumed to be data-rate limited. An important approach to improve the performance of linear precoded spatial multiplexing is optimizing the exploitation of the limited data rate on the feedback link.

The notion of linear precoding was introduced in [1], where the optimal linear precoder that minimizes the symbol mean square error for linear receivers under different constraints was derived. The bit-error-rate (BER) optimal precoder was introduced in [2], and the capacity optimal precoder in [3]. The first use of partial CSI at the transmitter was presented in [4], where the Lloyd algorithm is used to quantize the CSI. Other approaches focused on feeding back the mean of the channel [5], or the covariance matrix of the channel [6]. An overview of the achievable channel capacity with limited channel knowledge can be found in [7]. Schemes that directly select a quantized precoder from a codebook at the receiver, and feed back the precoder index to the transmitter have been independently proposed in [8, 9]. There, the authors proposed to design the precoder codebooks to maximize a subspace distance between two codebook entries, a problem which is known as the Grassmannian line packing problem. The advantage of directly quantizing the precoder is that the unitary precoder matrix [1] has less degrees of freedom than the full CSI matrix, and is thus more efficient to quantize. Several subspace distances to design the codebooks were proposed in [10], where the selected subspace distance depends on the function used to quantize the precoding matrix. In [11], a precoder quantization design criterion was presented that maximizes the capacity of the system and also the corresponding codebook design. A quantization function that directly minimizes the uncoded BER was proposed in [12].

This paper presents existing and novel schemes for linear precoding in the well-known vector quantization framework. We present the most popular selection and distortion criteria used for linear precoding, but also novel techniques like entropy coding, and finite state vector quantization. Further, we show how these schemes can be adapted to changing channel statistics, that is, to nonstationary sources.

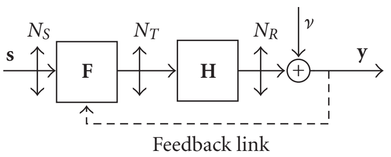


FIGURE 1: System model of the linear precoded spatial multiplexing MIMO system with limited feedback.

Notation

We use capital boldface letters to denote matrices, for example, $\mathbf{A}$, and small boldface letters to denote vectors, for example, $\mathbf{a}$. The Frobenius norm and the 2-norm of a matrix $\mathbf{A}$ are denoted as $\|\mathbf{A}\|_F$ and $\|\mathbf{A}\|_2$, respectively. $E(\cdot)$ denotes expectation and $P(\cdot)$ probability. $[\mathbf{A}]_{m,n}$ is the element in the mth row and nth column of $\mathbf{A}$. The $n \times n$ identity matrix is denoted as $\mathbf{I}_n$, and $\mathcal{U}_{m\times n}$ is the set of unitary $m \times n$ matrices. $\text{tr}(\mathbf{A})$ is the trace of $\mathbf{A}$, and $\det(\mathbf{A})$ the determinant of $\mathbf{A}$.

2. SYSTEM MODEL

Throughout the paper, we assume a narrowband spatial multiplexing MIMO system with N_T transmit and N_R receive antennas, transmitting $N_S \le \min(N_T, N_R)$ symbol streams, as depicted in Figure 1. The system equation at time instant n is

$$\mathbf{y}[n] = \mathbf{H}[n]\mathbf{F}[n]\mathbf{s}[n] + \boldsymbol{\nu}[n], \tag{1}$$

where $\mathbf{y}[n] \in \mathbb{C}^{N_R\times 1}$ is the received vector, $\boldsymbol{\nu}[n] \in \mathbb{C}^{N_R\times 1}$ is the additive noise vector, $\mathbf{s}[n] \in \mathbb{C}^{N_S\times 1}$ is the data symbol vector, $\mathbf{H}[n] \in \mathbb{C}^{N_R\times N_T}$ is the channel matrix, and $\mathbf{F}[n] \in \mathbb{C}^{N_T\times N_S}$ is the linear precoding matrix. We assume the data symbol vector $\mathbf{s}[n]$ is zero mean spatially and temporally white distributed over a complex finite alphabet, for example, the entries belong to a QAM alphabet $\mathcal{A}$, and the noise vector $\boldsymbol{\nu}[n]$ is zero mean spatially and temporally white complex Gaussian distributed. The channel matrix $\mathbf{H}[n]$ is zero mean possibly spatially and temporally correlated complex Gaussian distributed. The spatial correlation can be modeled using [13], $\mathbf{H}[n] = \mathbf{R}_r^{1/2}\mathbf{H}_w[n]\mathbf{R}_t^{1/2}$, where $\mathbf{R}_r$ is the receive covariance matrix, $\mathbf{R}_t$ the transmit covariance matrix, and $\mathbf{H}_w[n]$ the possibly temporally correlated channel matrix. We assume without loss of generality that the symbols and the noise have unit variance.

The singular value decomposition (SVD) of $\mathbf{H}[n]$ is defined as $\mathbf{H}[n] = \overline{\mathbf{U}}[n]\overline{\boldsymbol{\Sigma}}[n]\overline{\mathbf{V}}^H[n]$, where $\overline{\mathbf{U}}[n] \in \mathcal{U}_{N_R\times N_R}$, $\overline{\mathbf{V}}[n] \in \mathcal{U}_{N_T\times N_T}$, and $\overline{\boldsymbol{\Sigma}}[n]$ is a real nonnegative diagonal $N_R \times N_T$ matrix (the diagonal starts in the top left corner) with nonincreasing diagonal entries. The columns of $\overline{\mathbf{U}}[n]$ and $\overline{\mathbf{V}}[n]$ are called the left and right singular vectors, respectively, whereas the diagonal entries of $\overline{\boldsymbol{\Sigma}}[n]$ are the corresponding singular values. Only focusing on the N_S strongest modes of the channel (the ones with the largest singular values), let us define $\mathbf{U}[n] = [\overline{\mathbf{U}}[n]]_{:,1:N_S} \in \mathcal{U}_{N_R\times N_S}$, $\mathbf{V}[n] = [\overline{\mathbf{V}}[n]]_{:,1:N_S} \in \mathcal{U}_{N_T\times N_S}$, and $\boldsymbol{\Sigma}[n] = [\overline{\boldsymbol{\Sigma}}[n]]_{1:N_S,1:N_S}$, where $[\mathbf{A}]_{a:b,c:d}$ selects the submatrix of $\mathbf{A}$ on the rows a to b and the columns c to d, and the range indices are omitted when all rows or columns should be selected.

Many studies have been carried out to derive the optimal precoding matrix for a certain performance measure, see [1–3, 14]. In general, the optimal precoding matrix looks like

$$\mathbf{F}_{\text{opt}}[n] = \mathbf{V}[n]\boldsymbol{\Theta}[n]\mathbf{M}[n], \tag{2}$$

where $\boldsymbol{\Theta}[n] \in \mathbb{C}^{N_S\times N_S}$ is a diagonal power loading matrix, and $\mathbf{M}[n] \in \mathcal{U}_{N_S\times N_S}$ is a unitary mixing matrix. For some performance measures, the mixing matrix is arbitrary, whereas for other performance measures its value matters. In any case, it has been shown that for low-rate feedback channels, it is better not to feed back the power loading matrix and to stick to feeding back a unitary precoder [15]. That is why we will limit the precoding matrix $\mathbf{F}$ to be unitary, $\mathbf{F} \in \mathcal{U}_{N_T\times N_S}$.

The maximum data rate on the feedback link is R bits per channel use, and the feedback is assumed to be instantaneous and error free. We consider two different types of feedback channels: a dedicated feedback channel and a nondedicated feedback channel. A dedicated feedback channel is only used to transmit the precoder index to the transmitter, whereas a nondedicated feedback channel is also used for data transmission. The transmission is organized in a blockwise fashion, that is, feedback is only possible at the beginning of each new block, and every block has a duration of T_f. We assume the channel is perfectly known at the beginning of every block.

3. VECTOR QUANTIZATION

The data-rate-limited feedback link requires quantization of the channel matrix, resulting in a unitary precoder. The simplest approach is to use memoryless VQ, which quantizes every channel matrix $\mathbf{H}[n]$ separately. Hence, we can drop the time index n everywhere in this section. In memoryless VQ, we select a unitary $N_T \times N_S$ matrix $\mathbf{F}_i$ from a codebook $\mathcal{C} = \{\mathbf{F}_1, \dots, \mathbf{F}_K\}$ that minimizes or maximizes a given selection function S. We will denote $Q(\mathbf{H})$ as the quantized version of the channel matrix, but note that it actually represents the unitary precoder. More specifically, for a given selection function S and a given codebook $\mathcal{C}$, $Q(\mathbf{H})$ can be defined as

$$Q(\mathbf{H}) = \arg\min_{\mathbf{F}\in\mathcal{C}}/\max S(\mathbf{H}, \mathbf{F}), \tag{3}$$

where we take the minimum or the maximum depending on the selection function S. The quantization process can be further separated into an encoding step and a decoding step. The encoder α maps the channel into one of K precoder indices, which for simplicity reasons can be represented by the set $\mathcal{I} = \{1, 2, \dots, K\}$:

$$\alpha(\mathbf{H}) = \arg\min_{i\in\mathcal{I}}/\max S(\mathbf{H}, \mathbf{F}_i). \tag{4}$$

The decoder β simply maps the precoder index into one of the K precoders:

$$\beta(i) = \mathbf{F}_i. \tag{5}$$

Table 1: Example of a 4-entry ($K = 4$) codebook for a nondedicated and dedicated feedback link.

Precoders	Bitwords nondedicated	Bitwords dedicated
$\mathbf{F}_1$	$w_1 = 00$	$w_1 = /$
$\mathbf{F}_2$	$w_2 = 01$	$w_2 = 0$
$\mathbf{F}_3$	$w_3 = 10$	$w_3 = 1$
$\mathbf{F}_4$	$w_4 = 11$	$w_4 = 00$

So we actually have

$$Q(\mathbf{H}) = \beta(\alpha(\mathbf{H})). \tag{6}$$

Note that the index $i \in \mathcal{I}$ is transmitted over the feedback channel as a bitword w_i. What type of bitwords we have to feed back strongly depends on the type of feedback link: dedicated or nondedicated. In case of a nondedicated feedback channel, the transmitter has to be able to differentiate between a bitword and the data. This means the bitwords should be instantaneously decodable and thus prefix-free (PF), that is, a bitword can not contain any other bitword as a prefix. This is not the case in a dedicated feedback channel, where we can use non-prefix-free (NPF) bitwords. If the quantizer is well designed, all precoders $\mathbf{F}_i$ have more or less the same probability. Under that assumption, we can think of two ways to design our bitwords w_i. For a nondedicated feedback link, we can take K equal-length PF bitwords, leading to a feedback rate of $\lceil \log_2 K \rceil$ bits per channel use. For a dedicated feedback link, however, we can take any K bitwords with the smallest average length, leading to an average feedback rate of $1/K\sum_{i=1}^{K}\lfloor \log_2 i \rfloor$. An example is given in Table 1, where we assume a codebook with $K = 4$ entries. Next we focus on a number of selection functions for linear precoding, and we discuss the design of precoder codebooks.

3.1. Precoder selection

In this section, we will give an overview of some common selection functions S that have been proposed in recent literature. Whether we have to minimize or to maximize the selection function will be clear from the context. In [10], selection criteria are derived based on different performance measures. Optimizing the performance of the maximum likelihood (ML) receiver is related to maximizing the minimum Euclidean distance between any two possible noiseless received vectors:

$$S_{\mathrm{ML}}(\mathbf{H},\mathbf{F}) = \min_{\mathbf{s}_1,\mathbf{s}_2\in\mathcal{A}^{N_S\times 1},\, \mathbf{s}_1\neq\mathbf{s}_2} \left\|\mathbf{H}\mathbf{F}(\mathbf{s}_1-\mathbf{s}_2)\right\|_2. \tag{7}$$

For linear receivers, two performance measures are considered in [10], the minimum SNR on the substreams and the trace or determinant of the MSE matrix. Maximizing the first measure for the zero forcing (ZF) receiver is related to maximizing the minimal singular value (MSV) of the effective channel $\mathbf{HF}$:

$$S_{\mathrm{MSV}}(\mathbf{H},\mathbf{F}) = \lambda_{\min}\{\mathbf{HF}\}, \tag{8}$$

where $\lambda_{\min}\{\mathbf{A}\}$ denotes the MSV of the matrix $\mathbf{A}$. Minimizing the second measure for the minimum mean square error (MMSE) receiver, leads to minimizing the following selection function;

$$S_{\mathrm{MSE}}(\mathbf{H},\mathbf{F}) = m\left(\mathbf{I}_{N_S} + \mathbf{F}^H\mathbf{H}^H\mathbf{H}\mathbf{F}\right)^{-1}, \tag{9}$$

where $m = \mathrm{tr}$ or $m = \det$. Finally, [10] also proposes to maximize the mutual information (MI) between the transmitted symbol vector $\mathbf{s}$ and the received symbol vector $\mathbf{y}$ over the effective channel $\mathbf{HF}$:

$$S_{\mathrm{MI}}(\mathbf{H},\mathbf{F}) = \log_2\det\left(\mathbf{I}_{N_S} + \mathbf{F}^H\mathbf{H}^H\mathbf{H}\mathbf{F}\right). \tag{10}$$

It has been shown in [10] that the above performance measures can be associated to a subspace distance between the right singular vectors of $\mathbf{H}$, collected in $\mathbf{V}$, and $\mathbf{F}$. As such, this subspace distance could also be used as selection function to be minimized. The performance of the ML receiver, the minimum SNR on the substreams for the ZF receiver, and the trace of the MSE matrix for the MMSE receiver are all related to the projection 2-norm distance:

$$S_{\mathrm{P2}}(\mathbf{H},\mathbf{F}) = d_{\mathrm{P2}}(\mathbf{V},\mathbf{F}) = \left\|\mathbf{V}\mathbf{V}^H - \mathbf{F}\mathbf{F}^H\right\|_2, \tag{11}$$

whereas the determinant of the MSE matrix for the MMSE receiver and the MI criterion can be connected to the Fubini-Study distance:

$$S_{\mathrm{FS}}(\mathbf{H},\mathbf{F}) = d_{\mathrm{FS}}(\mathbf{V},\mathbf{F}) = \arccos\left|\det(\mathbf{V}^H\mathbf{F})\right|. \tag{12}$$

Next to minimizing those subspace distances, minimizing the chordal distance is also used as selection criterion,

$$\begin{aligned} S_C(\mathbf{H},\mathbf{F}) = d_C(\mathbf{V},\mathbf{F}) &= 1/\sqrt{2}\left\|\mathbf{V}\mathbf{V}^H - \mathbf{F}\mathbf{F}^H\right\|_F \\ &= \sqrt{\mathrm{tr}\left(\mathbf{I}_{N_S} - \mathbf{V}^H\mathbf{F}\mathbf{F}^H\mathbf{V}\right)}. \end{aligned} \tag{13}$$

This function is related to the performance of an orthogonal space-time block code (OSTBC) that is used on top of the precoder [16].

For all the above selection criteria (for the ML criterion this is only approximately true), the optimal unitary precoder is given by $\mathbf{VM}$, where $\mathbf{M}$ is an arbitrary $N_S \times N_S$ unitary matrix, that is, $\mathbf{M} \in \mathcal{U}_{N_S\times N_S}$. This unitary ambiguity can be a problem when we are interested in other performance measures, such as uncoded bit-error-rate (BER), for instance. We know that in that case, the actual structure of the ambiguity matrix becomes important [12]. One solution could of course be to simply minimize the BER:

$$S_{\mathrm{BER}}(\mathbf{H},\mathbf{F}) = \mathrm{BER}\,(\mathbf{H},\mathbf{F}). \tag{14}$$

However, this is often difficult to compute. A simpler solution might be to encode $\mathbf{V}$ using VQ and to adopt the optimal (or a suboptimal) unitary mixing matrix $\mathbf{M}$ according to [12]. Hence in that case we do not use $\mathbf{F}_i$ but $\mathbf{F}_i\mathbf{M}$ as a precoder at the transmitter. We could encode $\mathbf{V}$ for instance by minimizing the Frobenius norm between $\mathbf{V}$ and $\mathbf{F}$ [16].

$$\begin{aligned} S_F(\mathbf{H},\mathbf{F}) = d_F(\mathbf{V},\mathbf{F}) &= \|\mathbf{V}-\mathbf{F}\|_F \\ &= \sqrt{2\mathrm{tr}\left(\mathbf{I}_{N_S} - \Re(\mathbf{V}^H\mathbf{F})\right)}. \end{aligned} \tag{15}$$

This selection function is however not invariant to a phase shift of the singular vectors collected in $\mathbf{V}$. That is why, the Frobenius norm has been extended to the so-called modified Frobenius norm [17],

$$\begin{aligned} S_{\mathrm{MF}}(\mathbf{H},\mathbf{F}) = d_{\mathrm{MF}}(\mathbf{V},\mathbf{F}) &= \underset{\mathbf{\Theta}\in\mathcal{D}_{N_S}}{\operatorname{argmin}} \|V\mathbf{\Theta}-\mathbf{F}\|_F \\ &= \left\|\mathbf{V}\operatorname{diag}(\mathbf{V}^H\mathbf{F})\operatorname{diag}(|\mathbf{V}^H\mathbf{F}|)^{-1}-\mathbf{F}\right\|_F \qquad (16) \\ &= \sqrt{2\operatorname{tr}(\mathbf{I}_{N_S}-|\mathbf{V}^H\mathbf{F}|)}, \end{aligned}$$

where $\mathcal{D}_n \subset \mathcal{U}_{n\times n}$ is the set of all diagonal unitary $n \times n$ matrices. Notice how through the use of the real or absolute value of $\mathbf{V}^H\mathbf{F}$, instead of the product $\mathbf{V}^H\mathbf{F}\mathbf{F}^H\mathbf{V}$ in (13), we truly encode $\mathbf{V}$ instead of its subspace. Let us now discuss the codebook design.

3.2. Codebook design

In general, a codebook design aims at finding a set of precoders $\mathcal{C}$ that minimizes some average distortion,

$$D_{\mathrm{av}} = \int_{\mathbb{C}^{N_R\times N_T}} D(\mathbf{H},Q(\mathbf{H}))p(\mathbf{H})d\mathbf{H}, \qquad (17)$$

where $D(\mathbf{H},Q(\mathbf{H}))$ is the distortion between $\mathbf{H}$ and $Q(\mathbf{H})$, and $p(\mathbf{H})$ is the probability density function (PDF) of the channel matrix $\mathbf{H}$. The distortion function D can take many different forms depending on the performance measure we are interested in (as was the case for the selection function). In [10], it has been shown that if we are interested in the performance of the ML receiver, the minimum SNR on the substreams for the ZF receiver, or the trace of the MSE matrix for the MMSE receiver, we can take as distortion function, the squared projection 2-norm distance between $\mathbf{V}$ and $Q(\mathbf{H})$: $D_{\mathrm{P2}}(\mathbf{H},Q(\mathbf{H})) = d_{\mathrm{P2}}^2(\mathbf{V},Q(\mathbf{H}))$. On the other hand, if we care about the determinant of the MSE matrix for the MMSE receiver or the MI, we should take the squared Fubini-Study distance between $\mathbf{V}$ and $Q(\mathbf{H})$ as distortion function, $D_{\mathrm{FS}}(\mathbf{H},Q(\mathbf{H})) = d_{\mathrm{FS}}^2(\mathbf{V},Q(\mathbf{H}))$. Finally, the distortion function related to the performance of an orthogonal space-time block code (STBC) that is used on top of the precoder is presented in [16] as $D_C(\mathbf{H},Q(\mathbf{H})) = d_C^2(\mathbf{V},Q(\mathbf{H})) = \operatorname{tr}(\mathbf{I}_{N_S} - \mathbf{V}^H Q(\mathbf{H})Q(\mathbf{H})^H\mathbf{V})$. The reason why squared subspace distances are used as distortion functions (and not the performance measures themselves) is because they lead to simpler design procedures as detailed later on.

In [11], an alternative and more exact distortion measure for the MI is proposed, namely, the capacity loss introduced by quantization,

$$D_{\mathrm{CL}}(\mathbf{H},Q(\mathbf{H})) = \operatorname{tr}(\mathbf{\Lambda}-\mathbf{\Lambda}\mathbf{V}^H Q(\mathbf{H})Q(\mathbf{H})^H\mathbf{V}), \qquad (18)$$

where $\mathbf{\Lambda} = (\mathbf{I}_{N_S}+\mathbf{\Sigma}^2)^{-1}\mathbf{\Sigma}^2$. Note that this distortion function converges to the squared chordal distance D_C when the diagonal elements of $\mathbf{\Sigma}^2$ go to infinity.

All the above distortion functions are invariant to a left multiplication of the precoder with a unitary matrix. As already indicated in the previous section, this could create a problem when performance measures like the uncoded BER are considered. Taking the distortion function equal to the BER, that is, $D_{\mathrm{BER}}(\mathbf{H},Q(\mathbf{H})) = \mathrm{BER}(\mathbf{H},Q(\mathbf{H}))$ leads to a difficult codebook design. But as before, we could take the squared Frobenius norm or squared modified Frobenius norm between $\mathbf{V}$ and $Q(\mathbf{H})$ as a distortion function to solve this complexity problem, $D_F(\mathbf{H},Q(\mathbf{H})) = 2\operatorname{tr}(\mathbf{I}_{N_S} - \Re(\mathbf{V}^H Q(\mathbf{H})))$, $D_{\mathrm{MF}}(\mathbf{H},Q(\mathbf{H})) = 2\operatorname{tr}(\mathbf{I}_{N_S} - |\mathbf{V}^H Q(\mathbf{H})|)$. In this case, our goal is again to feedback $\mathbf{V}$, and we will not use the precoder $Q(\mathbf{H})$ but $Q(\mathbf{H})\mathbf{M}$ at the transmitter, where $\mathbf{M}$ is the optimal (or a suboptimal) unitary mixing matrix [12].

Now, the question is how we can solve (17) for a certain distortion function. We can basically distinguish between three different approaches: Grassmannian subspace packing, the generalized Lloyd (GL) algorithm, and the Monte-Carlo (MC) algorithm.

3.2.1. Grassmannian subspace packing

In case the distortion function is a subspace distance and the channel is spatially white, we can simplify (17) by means of a Grassmannian subspace packing problem. In such a problem, the objective is to find a set of unitary precoders that maximizes the minimal subspace distance between them [10, 16],

$$\max_{\mathcal{C}} \min_{\substack{\mathbf{F}_i,\mathbf{F}_j\in\mathcal{C}\\ \mathbf{F}_i\neq\mathbf{F}_j}} d(\mathbf{F}_i,\mathbf{F}_j), \qquad (19)$$

where d is any of the subspace distances we discussed above. Of course, such a codebook can also be used when the channel is not spatially white, but the performance will decrease with an increased spatial correlation of the channel.

3.2.2. Generalized Lloyd algorithm

The generalized Lloyd (GL) algorithm tries to solve (17) by iteratively optimizing the encoder and the decoder [18, 19]. For a given decoder β, the encoder is optimized by taking the precoder index leading to the smallest distortion (the so-called nearest neighbor condition):

$$\alpha(\mathbf{H}) = \underset{i\in\mathcal{I}}{\operatorname{argmin}}\, D(\mathbf{H},\beta(i)), \qquad (20)$$

thereby splitting the space of channel matrices into K channel regions $\mathcal{R}_i$, $i \in \mathcal{I}$;

$$\mathcal{R}_i = \{\mathbf{H} : D(\mathbf{H},\mathbf{F}_i) \le D(\mathbf{H},\mathbf{F}_j),\ \mathbf{F}_i,\mathbf{F}_j \in \mathcal{C},\ \mathbf{F}_i\neq\mathbf{F}_j\}. \quad (21)$$

On the other hand, for a given encoder α, the decoder β is optimized by taking the centroid of the related channel region (the so-called centroid condition),

$$\beta(i) = \underset{\mathbf{F}\in\mathcal{U}_{N_T\times N_S}}{\operatorname{argmin}} \int_{\mathcal{R}_i} D(\mathbf{H},\mathbf{F})p(\mathbf{H})d\mathbf{H}. \qquad (22)$$

Although not rigorously proven, the GL algorithm converges to a local minimum, which might not necessarily be the global minimum. To avoid working with the continuous channel distribution, the GL algorithm makes use of a set

Table 2: Example of feedback compression through entropy coding.

Codebook	$P(Q(\mathbf{H}[n]) = \mathbf{F}_i \mid Q(\mathbf{H}[n-1]) = \mathbf{F}_8)$	Huffman code	NPF code
$\mathbf{F}_8$	0.25	01	/
$\mathbf{F}_2$	0.20	11	0
$\mathbf{F}_7$	0.18	000	1
$\mathbf{F}_4$	0.16	001	00
$\mathbf{F}_3$	0.10	101	01
$\mathbf{F}_6$	0.08	1000	10
$\mathbf{F}_5$	0.02	10010	11
$\mathbf{F}_1$	0.01	10011	000

of training channels $\mathcal{T} = \{\mathbf{H}^{(r)}\}$, where r is the realization index. This set can be interpreted as the discrete channel distribution that approximates the continuous one. The more training vectors in the set, the better the approximation. Computing the exact centroid based on $\mathcal{T}$ is not always easy [20]. For the squared subspace distances as well as the capacity loss distortion function in (18), closed form expressions for the centroid exist. However, for the BER and even the squared Frobenius norm or squared modified Frobenius norm, a closed form expression does not exist. For those distortion functions, we simply apply a brute force (approximate) centroid computation by exhaustively searching the best possible candidate among the set of matrices $\mathbf{V}^{(r)}$ for which $\mathbf{H}^{(r)}$ belongs to the related region.

3.2.3. Monte-Carlo algorithm

Another interesting approach is the pure Monte-Carlo based design. Instead of trying to optimize an existing codebook, this design randomly generates codebooks, checks the average distortion (17) of these codebooks, and keeps the best one. As for the GL algorithm, we will make use of the set of training channels $\mathcal{T}$ to approximate the continuous channel distribution. Although this algorithm becomes computationally expensive for large dimensions, for small dimensions we have observed that the MC algorithm is a very good alternative to Grassmannian subspace packing or the GL algorithm.

4. FEEDBACK COMPRESSION THROUGH ENTROPY CODING

This section explores methods to compress the feedback requirements on the feedback link, without sacrificing performance. It uses variable-rate codes to encode highly probable precoder matrices with small bitwords and less probable precoder matrices with longer bitwords. This is called entropy coding [18]. However, as we already indicated in Section 3, if the memoryless VQ is well designed, all precoders $\mathbf{F}_i$ have more or less the same probability. We therefore try to exploit the time correlation of the channel and make use of the transition probabilities between precoders instead of the occurrence probabilities. Hence, instead of assigning a bitword w_i to a precoder $\mathbf{F}_i$, we assign a bitword $w_{i,j}$ to a precoder $\mathbf{F}_i$ if the previous precoder was the precoder $\mathbf{F}_j$. Our goal then is to minimize the average length

$$\sum_{i=1}^{K} l(w_{i,j}) P(Q(\mathbf{H}[n]) = \mathbf{F}_i \mid Q(\mathbf{H}[n-1]) = \mathbf{F}_j), \quad (23)$$

where $l(w_{i,j})$ is the length of the bitword $w_{i,j}$ and $P(Q(\mathbf{H}[n]) = \mathbf{F}_i \mid Q(\mathbf{H}[n-1]) = \mathbf{F}_j)$ is the transition probability from $\mathbf{F}_j$ to $\mathbf{F}_i$. Depending on the type of feedback channel, we obtain a different solution for (23). For a nondedicated feedback link, or in other words for PF bitwords, the solution of (23) is given by the Huffman code [21]. For a dedicated feedback link, or in other words for NPF bitwords, the solution of (23) is simply given by selecting any K bitwords with the smallest possible average length, and assigning the longest (smallest) bitwords to the lowest (highest) transition probabilities.

An example of a codebook for a dedicated feedback link and a nondedicated feedback link is depicted in Table 2. The transition probabilities are estimated through Monte-Carlo simulations. This example assumes that the previous quantized precoder is $Q(\mathbf{H}[n-1]) = \mathbf{F}_8$. Due to the time correlation of the channel, the most probable precoder in this example at time instant n is then again $\mathbf{F}_8$. Thus, the most probable precoder matrix $\mathbf{F}_8$ gets a short bitword assigned, whereas the precoders with lower probabilities get longer bitwords assigned.

Please note that for OFDM, where several precoder matrices for different tones are transmitted at the same time instant, the individual precoding matrices do not need to be instantaneously decodable. They can be jointly encoded, for example, through the use of arithmetic coding.

The scheme can be extended to incorporate error correcting codes to make it robust against errors on the feedback channel.

The above techniques rely on the exact knowledge or the knowledge of the order of the transition probabilities between the past precoder $Q(\mathbf{H}[n-1])$ and the actual precoder $Q(\mathbf{H}[n])$. Unfortunately, a closed form expression of the transition probabilities is not known, and difficult to derive due to the nonlinearity of the quantization. For the special case of known channel statistics, they can be estimated offline through a Monte-Carlo approach [22]. However, in practice the underlying channel statistics are unknown, or are changing at runtime. The next section provides a solution to this problem.

4.1. Adaptive entropy coding

In [23], we introduced a novel scheme to adaptively estimate the transition probabilities. The presented scheme is able to estimate the transition probabilities at runtime, and to adapt to changing channel statistics. The algorithm starts by assuming that all the different transitions are equiprobable. Then it counts the different transitions at both the decoder and the encoder, and updates the transition probabilities after each new feedback. Assuming a transition between the precoder $\mathbf{F}_j$ and the precoder $\mathbf{F}_k$ happens, the transition probability $P_{k,j}[n] = P(Q(\mathbf{H}[n]) = \mathbf{F}_k \mid Q(\mathbf{H}[n-1]) = \mathbf{F}_j)$ is updated as [18]

$$\begin{aligned} P_{k,j}[n] &= \frac{(N-1)P_{k,j}[n-1]+1}{N}, \\ P_{i,j}[n] &= \frac{(N-1)P_{i,j}[n-1]}{N} \quad \text{for } i \neq k. \end{aligned} \tag{24}$$

The factor N controls how fast or how accurate the probabilities are estimated. Larger values of N lead to a smaller increase or decrease after each iteration, and thus, to a slower, but more accurate estimation.

Instead of updating the transition probabilities, one can also directly update the Huffman code, in the case of a nondedicated feedback link [24–26]. However, the effect is very similar to the two-step approach of first updating the transition probabilities and then computing the new Huffman code.

5. FINITE-STATE VECTOR QUANTIZATION (FSVQ)

In this section, we will look at a number of methods to improve the performance exploiting the maximal data rate of R bits per channel use on the feedback channel. We will present the different methods in the well-developed framework of finite-state vector quantization (FSVQ), and we closely follow [18].

Before introducing FSVQ, let us consider a so-called switched VQ, consisting of a finite number of memoryless VQs and a classifier that periodically decides which memoryless VQ is best and feeds back the index of this VQ to the decoder. The decision of the classifier is generally based on an estimate of the statistics of the channel. An example of this approach is given in [27], where the different memoryless VQ codebooks are constructed by rotating and scaling a specific root codebook. The drawback of this approach is of course the additional feedback overhead due to the fact that the classifier periodically feeds back the index of the best memoryless VQ.

FSVQ solves this problem since it does not require any additional side information. An FSVQ has some built-in mechanism to determine which of the memoryless VQs should be used to transform the current channel into a quantization index. It is the current state that determines which memoryless VQ to employ, and that is why the related codebook is called the state codebook. The current state together with the obtained quantization index then determines the next state through the so-called next-state function. This is explained in more detail next.

Suppose we have a set of K states, which without loss of generality can be denoted as $\mathcal{S} = \{1, 2, \dots, K\}$. Every state $s \in \mathcal{S}$ is related to a state codebook $\mathcal{C}_s = \{\mathbf{F}_{1,s}, \mathbf{F}_{2,s}, \dots, \mathbf{F}_{N,s}\}$. The encoder α maps the current channel and state into one of N quantization indices, which for simplicity reasons can be represented by the set $\mathcal{I} = \{1, 2, \dots, N\}$. Assume for instance that at time instant n the channel and state are given by $\mathbf{H}[n]$ and $s[n]$, respectively, then we can describe our encoder as

$$\alpha(\mathbf{H}[n], s[n]) = \underset{i \in \mathcal{I}}{\operatorname{argmin}}\, S(\mathbf{H}[n], \mathbf{F}_{i,s[n]}), \tag{25}$$

where S is one of the selection functions described in Section 3.1. The decoder β simply maps the current quantization index and state into one of the N precoders of the related state codebook. Assume for instance that at time instant n the quantization index and state are given by $i[n]$ and $s[n]$, respectively, then our decoder can be expressed as

$$\beta(i[n], s[n]) = \mathbf{F}_{i[n], s[n]}. \tag{26}$$

So the overall quantization procedure can be written as

$$Q(\mathbf{H}[n], s[n]) = \beta(\alpha(\mathbf{H}[n], s[n]), s[n]). \tag{27}$$

Finally, we need a mechanism that tells us how to go from one state to the next. This is obtained by the next-state function. Keeping in mind that both the encoder and decoder should be able to track the state, the next-state function f can only be guided by the quantization index. Assume that at time instant n the current quantization index and state are given by $i[n]$ and $s[n]$, respectively, then the next-state function can be expressed as follows:

$$s[n+1] = f(i[n], s[n]). \tag{28}$$

An FSVQ is now completely determined by the state space $\mathcal{S} = \{1, 2, \dots, K\}$, the state codebooks $\mathcal{C}_s = \{\mathbf{F}_{1,s}, \mathbf{F}_{2,s}, \dots, \mathbf{F}_{N,s}\}$ for all $s \in \mathcal{S}$, the next state function f, and the initial state $s[0]$. Note that the union of all state codebooks is called the super codebook $\mathcal{C} = \bigcup_{s \in \mathcal{S}} \mathcal{C}_s$, which contains no more than KN precoders.

As in memoryless VQ, we can consider two ways to assign bitwords w_i to the indices $i \in \mathcal{I}$. We can use N equal-length PF bitwords (for a nondedicated feedback link), with a feedback rate of $\lceil \log_2 N \rceil$ bits per channel use, or N increasing-length NPF bitwords (for a dedicated feedback link), with an average feedback rate of $1/N \sum_{i=1}^{N} \lfloor \log_2 i \rfloor$. This assignment is again based on the assumption that for a certain state s, the precoders $\mathbf{F}_{i,s}$ have more or less the same probability.

Two special classes of FSVQs are the labeled-state and the labeled-transition FSVQs. Basically, every FSVQ can always be represented in either form and as a result, these classes are not restrictive. In a labeled-state FSVQ, the states are basically labeled by the quantized precoders, and the quantized precoder that is produced depends on the arrival state. In other words, the labeled-state FSVQ decoder β only depends on the next state:

$$\beta(i[n], s[n]) = \mathbf{F}_{i[n],s[n]} = \phi(f(i[n], s[n])) = \phi(s[n+1]). \tag{29}$$

In a labeled-transition FSVQ, not the states but the state transitions are labeled by the quantized precoders, and the selected quantized precoder is determined not by the arrival state but by both the departure state and the arrival state. Hence, the labeled-transition FSVQ decoder β depends on the current as well as on the next state:

$$\begin{aligned}\beta(i[n], s[n]) &= \mathbf{F}_{i[n],s[n]} = \psi(s[n], f(i[n], s[n])) \\ &= \psi(s[n], s[n+1]).\end{aligned} \tag{30}$$

As will be illustrated later on, the design of an FSVQ is often based on an initial classifier that classifies channels into states. Such a classifier could for instance be a simple memoryless VQ with a codebook $\mathcal{C}_{\text{class}} = \{\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_K\}$ that assigns a state $s \in \mathcal{S}$ to a channel $\mathbf{H}[n]$ using the function g,

$$g(\mathbf{H}[n]) = \underset{s\in\mathcal{S}}{\operatorname{argmin}}\, S_{\text{class}}(\mathbf{H}[n], \mathbf{F}_s), \tag{31}$$

where the selection function S_{class} is one of the functions introduced in Section 3.1, and could possibly be different from the selection function S chosen in the encoder (25). We will come back to this issue in Section 5.2.

In the next few subsections, we will describe a few methodologies to design the state codebooks and the next state functions based on the initial classifier. In the first subsection, we will discuss some labeled-state FSVQ designs. These are basically existing designs, although they have not always been introduced in the framework of FSVQ or in the context of time-correlated channels. In the second subsection, we describe the so-called omniscient design, which is a completely novel feedback compression method. Note that it is still possible to iteratively improve the obtained state codebooks, given the next-state function, as illustrated in [18, page 536]. However, this generally only shows marginal performance gains over the initial designs, and thus we will not consider it in this work.

5.1. Labeled-state FSVQ designs

In this section, we discuss a few labeled-state FSVQ feedback designs, where each state $s \in \mathcal{S}$ is labeled with the precoder $\mathbf{F}_s$ from the classifier codebook $\mathcal{C}_{\text{class}}$. Hence, the decoder β is then simply given by

$$\beta(i[n], s[n]) = \phi(s[n+1]) = \mathbf{F}_{s[n+1]}. \tag{32}$$

In that case, the super codebook $\mathcal{C}$ corresponds to the classifier codebook $\mathcal{C}_{\text{class}}$, and the state codebooks $\mathcal{C}_s$ are subsets of the classifier codebook $\mathcal{C}_{\text{class}}$. Below wedescribe a few popular methods to determine the state codebooks and next-state function.

Table 3: Example of transition probabilities and precoder distances assuming the previous state was $s = 8$.

s'	$P(g(\mathbf{H}[n]) = s' \mid g(\mathbf{H}[n-1]) = 8)$	$D(\mathbf{F}_{s'}, \mathbf{F}_8)$
1	0.0380	1, 6292
2	0.0200	1, 4550
3	0.0132	1, 3461
4	0.0365	1, 2801
5	0.0250	1, 1548
6	0.0397	1, 3112
7	0.0232	1, 4487
8	0.8045	0

5.1.1. Conditional histogram design

For the conditional histogram design, the next states of a current state s are the N states s' that have the highest probability to be reached from state s in terms of the initial classifier. Hence, the state codebook $\mathcal{C}_s$ is the set of N precoders $\mathbf{F}_{s'}$ corresponding to the N states s' that have the highest transition probability $P(g(\mathbf{H}[n]) = s' \mid g(\mathbf{H}[n-1]) = s)$. If we define, without loss of generality, $\mathbf{F}_{i,s}$ as the precoder $\mathbf{F}_{s'}$ of the state s' with the ith highest transition probability $P(g(\mathbf{H}[n]) = s' \mid g(\mathbf{H}[n-1]) = s)$, then the next-state function $f(i, s)$ is simply given by this state s'. Note that the transition probabilities can be computed as in Section 4, but the adaptive approach can not be used here because the decoder does not have knowledge about the current channel. An example is given in Table 3, where we assume that the current state is $s = 8$. Assuming the state codebooks have size $N = 4$, the state codebook $\mathcal{C}_8$ is given by $\mathcal{C}_8 = \{\mathbf{F}_8, \mathbf{F}_6, \mathbf{F}_1, \mathbf{F}_4\}$. Although presented in a different framework, a similar approach has been proposed in [22].

5.1.2. Nearest neighbor design

For the nearest neighbor design, the next states of a current state s are not the N states s' that have the highest transition probability, but the N states s' that have the closest precoder to the precoder of state s in terms of some distance d, which could be a subspace distance, the Frobenius norm d_F, or the modified Frobenius norm d_{MF}, although the latter are not strictly speaking distances. Hence, the state codebook $\mathcal{C}_s$ is the set of N precoders $\mathbf{F}_{s'}$ that have the smallest distance $d(\mathbf{F}_{s'}, \mathbf{F}_s)$. If we define, without loss of generality, $\mathbf{F}_{i,s}$ as the precoder $\mathbf{F}_{s'}$ of the state s' with the ith smallest distance $d(\mathbf{F}_{s'}, \mathbf{F}_s)$, then the next-state function $f(i, s)$ is simply given by this state s'. Again looking at the example in Table 3, we now see that the state codebook $\mathcal{C}_8$ is given by $\mathcal{C}_8 = \{\mathbf{F}_8, \mathbf{F}_5, \mathbf{F}_4, \mathbf{F}_6\}$.

In the context of orthogonal frequency division multiplexing (OFDM), this approach has already been proposed in [28] to compress the feedback of the precoders on the different subcarriers.

5.1.3. Discussion

The problem of both the conditional histogram design and the nearest neighbor design is that if K/N is large and the time correlation of the channel is small, the optimal transition might be not one of the N most likely ones or not one of the N transitions with the smallest distance between precoders. This could lead to a so-called *derailment problem*. Taking a smaller K/N is a possible solution, but it either leads to a lower performance (decreasing K) or a higher feedback rate (increasing N). As suggested in [18, page 540], the derailment problem could also be solved by periodic reinitialization.

5.2. Omniscient design

In this section, we present a novel feedback compression method, based on what in the field of vector quantization is known as the *omniscient* design [18]. In general, the omniscient design provides the best performance of all the FSVQ design approaches [18].

To explain the omniscient design, let us assume that the next-state function is not determined by the current quantization index and state, but simply by the current channel, for instance by means of the classifier function g,

$$s[n+1] = g(\mathbf{H}[n]). \tag{33}$$

The state codebook $\mathcal{C}_s$ for a state s can then be designed by minimizing some average distortion:

$$D_{\mathrm{av},s} = \int_{\mathbb{C}^{N_R\times N_T}} D(\mathbf{H}, Q(\mathbf{H}, s))\, p(\mathbf{H}[n] \mid g(\mathbf{H}[n-1]) = s)\, d\mathbf{H}, \tag{34}$$

where $D(\mathbf{H}, Q(\mathbf{H}, s))$ is the distortion between $\mathbf{H}$ and $Q(\mathbf{H}, s)$, and $p(\mathbf{H}[n] \mid g(\mathbf{H}[n-1]) = s)$ is the conditional probability density function of $\mathbf{H}[n]$ given $g(\mathbf{H}[n-1]) = s$, or equivalently, given the current state $s[n] = s$. Any of the distortion functions presented in Section 3.2 can be considered. We can now solve (34) by the GL algorithm or the MC algorithm, as was done in Sections 3.2.2 and 3.2.3. This requires a set of training channels $\mathcal{T}_s$. To construct $\mathcal{T}_s$, we first generate a large set of pairs of consecutive channels based on the channel statistics, $\mathcal{P} = \{(\mathbf{H}^{(r)}[n-1], \mathbf{H}^{(r)}[n])\}$, where r is the realization index. From this set $\mathcal{P}$ we construct $\mathcal{T}_s$ as the set of channels $\mathbf{H}^{(r)}[n]$ for which $g(\mathbf{H}^{(r)}[n-1]) = s$, that is, $\mathcal{T}_s = \{\mathbf{H}^{(r)}[n] \mid (\mathbf{H}^{(r)}[n-1], \mathbf{H}^{(r)}[n]) \in \mathcal{P}$ and $g(\mathbf{H}^{(r)}[n-1]) = s\}$. The problem of this approach is that the decoder can not track the state, because it does not have access to the current channel. Hence, it is assumed here that the decoder is *omniscient* and we actually do not have an FSVQ. Thus, we should replace $\mathbf{H}[n]$ in the next-state function by its estimate $\widehat{\boldsymbol{H}}[n]$ that is computed based on the quantized precoder $Q(\mathbf{H}[n], s[n])$ known to the decoder. As an estimate, we could for instance consider $\widehat{\boldsymbol{H}}[n] = [Q(\mathbf{H}[n], s[n]), \mathbf{0}_{N_T\times(N_R-N_S)}]^H$. This is of course not a good channel estimate for equalization, but it is good in terms of the N_S largest right singular vectors collected in $\mathbf{V}[n]$. Hence, if the classifier g is designed based on a selection function $\mathcal{S}_{\mathrm{class}}$ that only depends on $\mathbf{V}[n]$, then $g(\widehat{\boldsymbol{H}}[n])$ is a good approximation of $g(\mathbf{H}[n])$. That is why we often choose S_{class} based on a subspace distance (S_{P2}, S_{FS}, or S_C), the Frobenius norm (S_F), or the modified Frobenius norm (S_{MF}), irrespective of what is chosen as selection function S in the encoder (25). So, we keep the idealized state codebooks $\mathcal{C}_s$ but we change the next-state function into

$$s[n+1] = g(\widehat{\boldsymbol{H}}[n]) = f(i[n], s[n]). \tag{35}$$

This way we obtain an FSVQ. When K/N gets smaller and the time correlation of the channel gets larger, that is, when the regions related to the classifier codebook $\mathcal{C}_{\mathrm{class}}$ get larger compared to the regions related to the state codebooks $\mathcal{C}_s$, the approximation gets better. On the other hand, however, for a fixed N, it is sometimes worth to increase K to benefit from an increased knowledge about the past.

In [18], it is mentioned that the omniscient design leads to a labeled-transition FSVQ, because given a current state, every possible quantization index leads to a different next state. However, this is not necessarily true. Different quantization indices could sometimes lead to the same next state, and thus in general we do not have a labeled-transition FSVQ.

5.3. Adaptive FSVQ

Unfortunately, it is not trivial to extend the FSVQ to adapt to changing channel characteristics, that is, to a nonstationary source. The adaptation of the state codebooks $\mathcal{C}_s$ has to rely on information that is available both at the encoder and the decoder. This shared information can for instance consist of the last l states $s[n], s[n-1], \ldots, s[n-l+1]$ and the last l quantized precoders $Q(\mathbf{H}[n], s[n]), Q(\mathbf{H}[n-1], s[n-1]), \ldots, Q(\mathbf{H}[n-l+1], s[n-l+1])$. We restrict our approach to such a window of l samples due to memory restrictions, and we forget past samples for which the channel might have different characteristics. Whenever the precoder is $Q(\mathbf{H}[n], s[n]) = \mathbf{F}_{i,s[n]}$, we know that the channel matrix $\mathbf{H}[n]$ lies in some region $\mathcal{R}_{i,s[n]}$. Assuming a realistic channel distribution, we can then define one or more random channel matrices that also lie in the region $\mathcal{R}_{i,s[n]}$. Finally, the FSVQ design algorithms mentioned previously can be used with the new training sequence to design the new state codebooks. Note that the state codebooks, and thus the quantizer regions, are recalculated from scratch after each feedback. Instead, we could also consider updating the codebook as done in competitive learning [29]. However, such techniques still have to be adapted to take the unitary constraint of the precoding matrix into account, and they are considered future work.

6. SIMULATIONS

In this section, we are providing numerical results for the different schemes and design approaches presented so far. We assume that $N_S = 2$ data streams are transmitted over $N_T = 4$ antennas. The receiver is equipped with $N_R = 2$ receive antennas, and QPSK modulation is used.

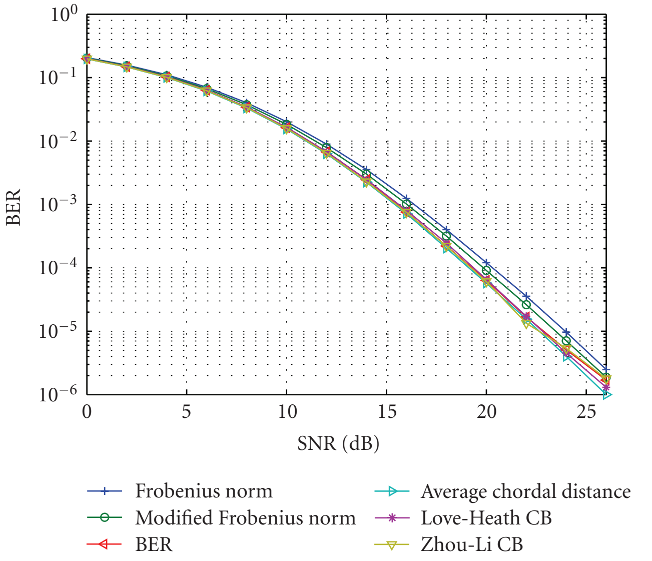


FIGURE 2: Comparison between different codebooks using the BER selection criterion ($N_S = 2$, $N_T = 4$, $N_R = 2$, $|\mathcal{C}| = 16$, ZF receiver).

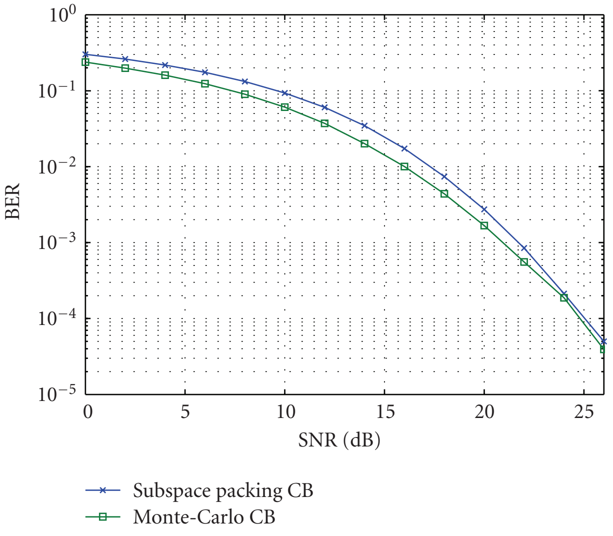


FIGURE 3: Comparison of different codebooks for memoryless VQ for a spatially correlated channel ($N_S = 2$, $N_T = 4$, $N_R = 4$, $|\mathcal{C}| = 4$, ZR receiver).

We start in Section 6.1 by comparing the BER performance for different codebooks using the BER criterion as selection function. Section 6.2 then shows the performance of Monte-Carlo and subspace packing codebooks for spatially correlated channels. In Section 6.3, the possible feedback compression gains of entropy coding over memoryless VQ are shown for time-correlated channels. Section 6.4 shows how fast the adaptive entropy coding schemes adapt to changing channel statistics. The following subsection then compares FSVQ to memoryless VQ, and it also compares the different FSVQ design approaches. Finally, Section 6.6 shows the duality between FSVQ and entropy coding.

6.1. *Memoryless VQ*

Figure 2 compares the performance of different codebook designs presented in Section 3.2. The BER is used as selection function (14). The Frobenius norm, the modified Frobenius norm, and the chordal distance codebook are using the Monte-Carlo algorithm to solve (17), using the respective squared distances as distortion function. The BER codebook is also designed using the Monte-Carlo algorithm. The Love-Heath codebook [10] and the Zhou-Li codebook [12] are designed to optimize (19) with the chordal distance as subspace distance. Love and Heath were using techniques from [30], and Zhou and Li were using the generalized Lloyd algorithm. The simulation shows that the performance of the different codebooks is similar, and even using the BER as a distortion function in the codebook design does not yield a noticeable performance gain.

6.2. *Codebook design for spatially correlated channels*

Figure 3 compares the performance of two codebooks for a spatially correlated channel. One codebook is designed using the Grassmannian subspace packing approach with the chordal distance, and the other codebook is designed using the Monte-Carlo algorithm with the squared modified Frobenius norm as distortion function. The channel is modeled using the measurements in [31], and the BER selection function (14) is used to choose the best codebook entry. We see that the Monte-Carlo codebook, which takes the channel correlation into account, outperforms the Grassmannian subspace packing codebook, which aims at spatially white channels.

6.3. *Entropy coding*

Figure 4 depicts the compression gains possible through entropy coding. The channel is modeled through Jakes' model with the Doppler spread fixed. The mean feedback rate is depicted as a function of the frame duration T_f. A small frame duration implies a highly correlated channel, whereas a longer frame duration implies a less correlated channel. The Huffman code is used as prefix-free code, and the simple binary numbering from Table 2 is used as the non-prefix-free code. The modified Frobenius norm (16) is used as selection function and the squared modified Frobenius norm as distortion function to design the codebook using the Monte-Carlo algorithm. The transition probabilities used to design the entropy codes are estimated through Monte-Carlo simulations.

We see that the prefix-free code achieves a mean feedback rate of 1 bit for highly correlated channels, whereas the non-prefix-free code can even achieve 0 bits, that is, no feedback is necessary. For longer frame durations, that is, uncorrelated channels, the mean feedback rate for the Huffman encoded bitwords converges to 4 bits, since the transitions between the different codewords become equiprobable, and then the Huffman code assigns equal-length bitwords to all the

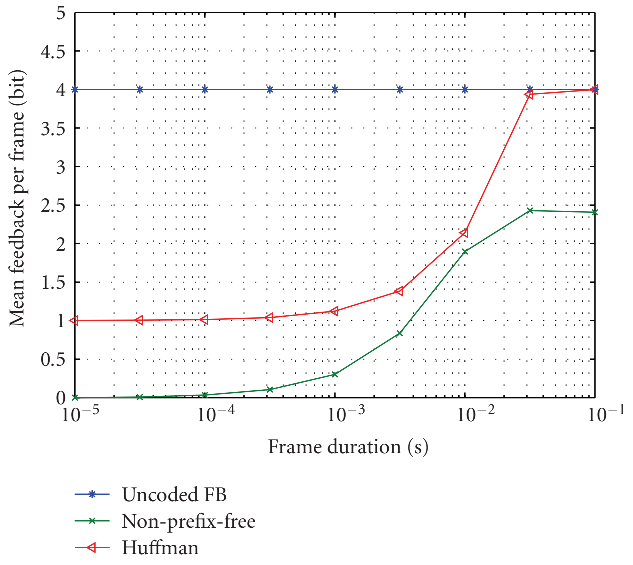


Figure 4: Feedback compression with entropy coding for different frame lengths ($N_S = 2$, $N_T = 2$, $f_D = 30$ Hz, $|\mathcal{C}| = 16$).

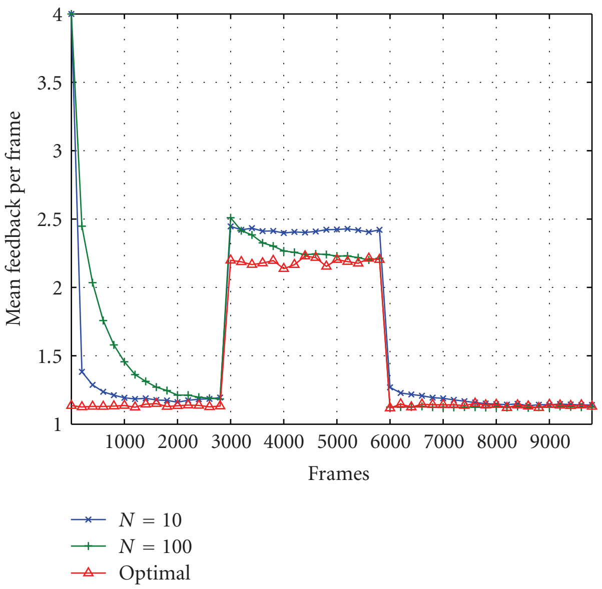


Figure 5: Tradeoff between adaptation speed and accuracy using a Huffman code ($f_D = 30$ Hz, $N_S = N_T = 2$, $|\mathcal{C}| = 16$).

precoders. The non-prefix-free code converges to 2.375 bits for uncorrelated channels since the transitions between the different codewords become equiprobable as well, and thus it assigns the binary numbering bitwords randomly.

6.4. *Adaptive entropy coding*

The tradeoff between adaptation speed and accuracy for adaptive entropy coding is depicted in Figures 5 and 6. To depict the adaptation of the adaptive entropy coding to changing channel statistics, we changed the frame duration from 10^{-3} seconds to 10^{-2} seconds after 3000 frames, and back after another 3000 frames. The remaining simulation parameters are identically as in the previous subsection.

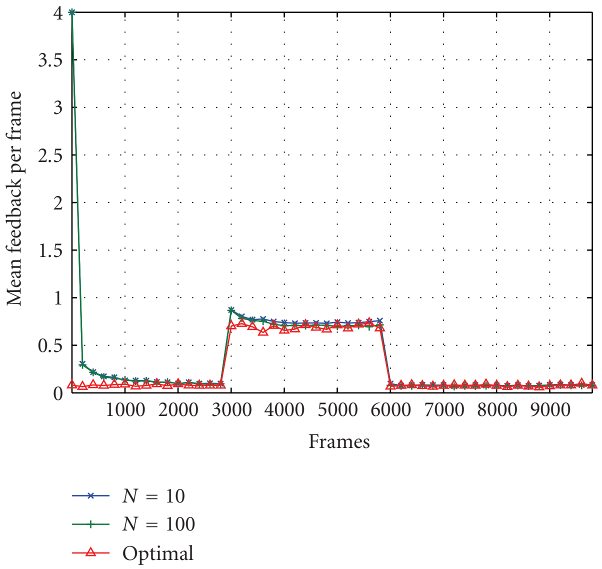


Figure 6: Tradeoff between adaptation speed and accuracy using a non-prefix-free code ($f_D = 30$ Hz, $N_S = N_T = 2$, $|\mathcal{C}| = 16$).

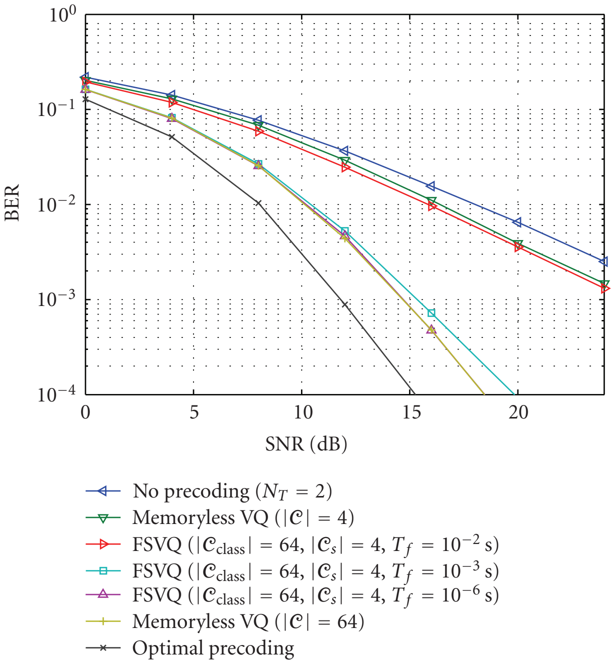


Figure 7: Comparison of several codebook design approaches ($N_S = 2$, $N_T = 4$, $N_R = 2$, $f_D = 30$ Hz, MMSE receiver).

Figure 5 assumes a nondedicated feedback channel. We see how the selection of the weighting factor N controls the tradeoff between performance and speed of the adaptive encoding process. For small N, the transition probabilities are estimated faster but less accurate, and for higher N, the estimation is slower but more accurate.

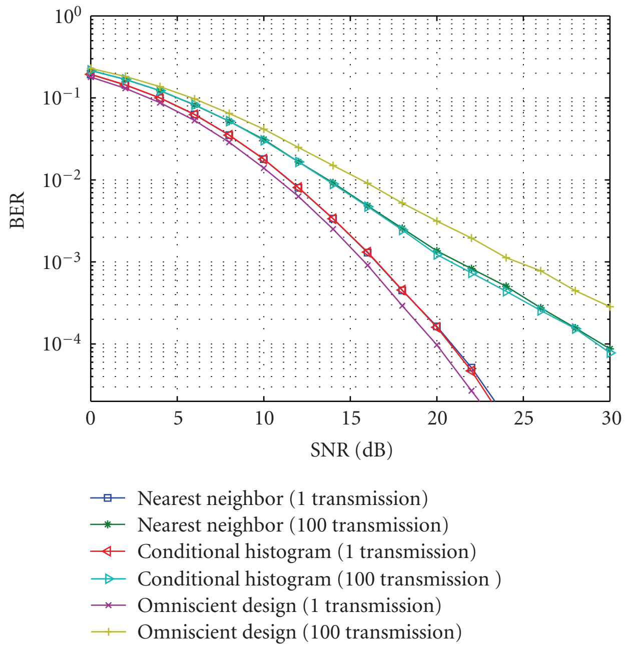


FIGURE 8: Comparison between the different FSVQ design approaches ($N_S = 2$, $N_T = 4$, $N_R = 2$, $|\mathcal{C}_{\text{class}}| = 16$, $|\mathcal{C}_s| = 4$, $T_f = 10^{-3}$ s, $f_D = 30$ Hz, SNR $= 10$ dB, ZF receiver).

Figure 6 shows a similar scenario, but for a dedicated feedback channel, where the bitwords are designed using the non-prefix-free code from Table 2. We see that the system quickly adapts to the changing frame lengths for both values of N, since the encoding of the bitwords does no longer depend on the exact transition probabilities but only on their order.

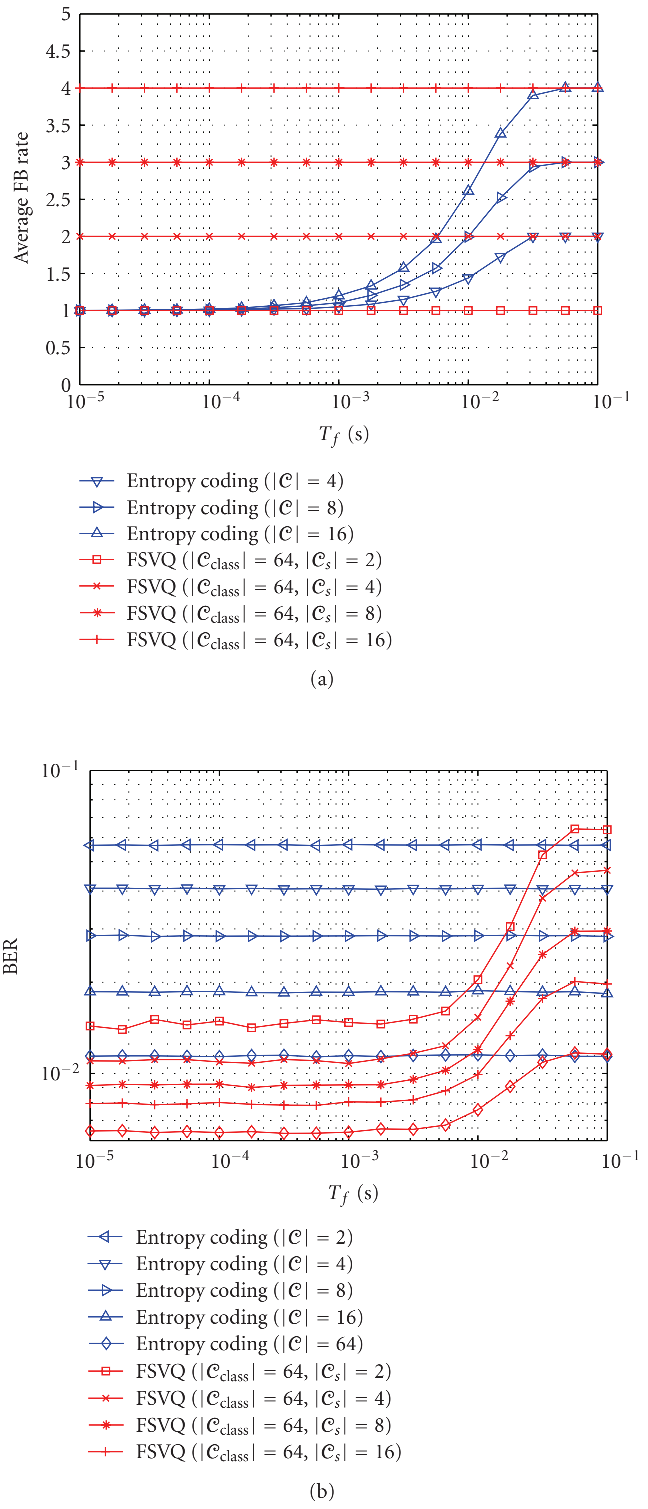
6.5. FSVQ

The performance of different state codebook designs is depicted in Figure 7. The FSVQs are created using the omniscient design. The different codebooks are designed with the squared modified Frobenius norm as distortion function, and the modified Frobenius norm (16) is used as selection function for the classifier (33) as well as for the quantization (27).

We see that the performance of the FSVQ highly depends on the time correlation of the channel. If the time correlation between the channels is high, the 2 bit feedback of a FSVQ has the same BER performance as the 4 bit memoryless VQ. However, for less correlated channels the performance drops to the same performance as the 2 bit memoryless VQ.

Different design approaches for FSVQ codebooks are shown in Figure 8. We simulate for the different design approaches the performance after 1 transmission and after 100 transmissions. We use the same distortion and selection functions as in the previous simulations.

We see that the omniscient design performs best after 1 transmission, but it also suffers the most from the derailment problem, that is, its performance after 100 transmissions is worse than the nearest neighbor and the conditional histogram design. This effect can be counteracted through periodic reinitialization.



(a)

(b)

FIGURE 9: Comparison of adaptive entropy coding and FSVQ ($N_S = 2$, $N_T = 4$, $N_R = 2$, $f_D = 30$ Hz, SNR $= 10$ dB, MMSE receiver).

6.6. Comparison entropy coding and FSVQ

We compare the omniscient design with the entropy coding approach for a MIMO system with a nondedicated feedback link. Figure 9 shows the average feedback rate and the BER of the linear MMSE receiver as a function of the frame length T_f. The modified Frobenius norm is used as selection function, and the squared modified Frobenius norm is used as distortion function to design the codebooks. We consider codebooks for the entropy coding approach with $|\mathcal{C}| = 2, 4, 8$, and 16, whereas for the omniscient design we take $|\mathcal{C}_{\text{class}}| = 64$ and $|\mathcal{C}_s| = 2, 4, 8$, and 16. For the entropy coding approach, the BER is constant and the average feedback rate increases with an increasing Doppler spread. On the other hand, for the omniscient design, the average feedback rate is constant and the BER increases with an increasing Doppler spread. Hence, the question basically is how their average feedback rates (BERs) compare for the same BER (average feedback rate). To answer this question, let us take a look at a few examples. We see that the entropy coding approach with $|\mathcal{C}| = 8$ has the same average feedback rate as the omniscient design with $|\mathcal{C}_{\text{class}}| = 64$ and $|\mathcal{C}_s| = 4$ at $T_f \approx 0.01$ s. However, at this frame length, the first has a worse BER as the latter. Similarly, we see that the entropy coding approach with $|\mathcal{C}| = 8$ has the same BER as the omniscient design with $|\mathcal{C}_{\text{class}}| = 64$ and $|\mathcal{C}_s| = 4$ at $T_f \approx 0.02$ s. But at this frame length, the first has a higher average feedback rate as the latter. Other examples show the same behavior. Hence, we can conclude that for this particular set-up, the entropy coding approach is worse than the omniscient design.

7. CONCLUSION

In this paper, we presented existing and novel schemes exploiting limited feedback for linear precoded spatial multiplexing in the framework of vector quantization. We depicted the different selection and distortion functions to generate the codebooks, and to quantize the input. Further, we considered the problem of reducing the data rate on the feedback link, and the problem of optimizing the overall performance of the system, both for stationary and for nonstationary sources.

ACKNOWLEDGMENTS

This paper has been presented in part at the 2007 International Conference on Acoustics, Speech and Signal Processing (ICASSP), Honolulu, Hawaii, USA, the 2007 International Symposium on Signal Processing and its Applications (ISSPA), Sharjah, UAE, and the 2007 International ITG/IEEE Workshop on Smart Antennas (WSA), Vienna, Austria. This research was supported in part by NWO-STW under the VIDI program (DTC.6577).

REFERENCES

[1] A. Scaglione, P. Stoica, S. Barbarossa, G. B. Giannakis, and H. Sampath, "Optimal designs for space-time linear precoders and decoders," IEEE Transactions on Signal Processing, vol. 50, no. 5, pp. 1051–1064, 2002.

[2] D. P. Palomar, M. Bengtsson, and B. Ottersten, "Minimum BER linear transceivers for MIMO channels via primal decomposition," IEEE Transactions on Signal Processing, vol. 53, no. 8, pp. 2866–2882, 2005.

[3] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," European Transactions on Telecommunications, vol. 10, no. 6, pp. 585–595, 1999.

[4] A. Narula, M. J. Lopez, M. D. Trott, and G. W. Wornell, "Efficient use of side information in multiple-antenna data transmission over fading channels," IEEE Journal on Selected Areas in Communications , vol. 16, no. 8, pp. 1423–1436, 1998.

[5] E. Visotsky and U. Madhow, "Space-time transmit precoding with imperfect feedback," IEEE Transactions on Information Theory , vol. 47, no. 6, pp. 2632–2639, 2001.

[6] S. A. Jafar, S. Vishwanath, and A. Goldsmith, "Channel capacity and beamforming for multiple transmit and receive antennas with covariance feedback," in *Proceedings of IEEE International Conference on Communications (ICC '01)*, vol. 7, pp. 2266–2270, Helsinki, Finland, June 2001.

[7] A. Goldsmith, S. A. Jafar, N. Jindal, and S. Vishwanath, "Capacity limits of MIMO channels," IEEE Journal on Selected Areas in Communications , vol. 21, no. 5, pp. 684–702, 2003.

[8] D. J. Love, R. W. Heath Jr., and T. Strohmer, "Quantized maximum ratio transmission for multiple-input multiple-output wireless systems," in *Conference Record of the 36th Asilomar Conference on Signals, Systems, and Computers (ACSSC '06)*, vol. 1, pp. 531–535, Pacific Grove, Calif, USA, November 2002.

[9] K. K. Mukkavilli, A. Sabharwal, E. Erkip, and B. Aazhang, "On beamforming with finite rate feedback in multiple-antenna systems," IEEE Transactions on Information Theory , vol. 49, no. 10, pp. 2562–2579, 2003.

[10] D. J. Love and R. W. Heath Jr., "Limited feedback unitary precoding for spatial multiplexing systems," IEEE Transactions on Information Theory , vol. 51, no. 8, pp. 2967–2976, 2005.

[11] J. C. Roh and B. D. Rao, "Transmit beamforming in multiple-antenna systems with finite rate feedback: a VQ-based approach," IEEE Transactions on Information Theory , vol. 52, no. 3, pp. 1101–1112, 2006.

[12] S. Zhou and B. Li, "BER criterion and codebook construction for finite-rate precoded spatial multiplexing with linear receivers," IEEE Transactions on Signal Processing , vol. 54, no. 5, pp. 1653–1665, 2006.

[13] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, Cambridge, UK, 2003.

[14] H. Sampath, P. Stoica, and A. Paulraj, "Generalized linear precoder and decoder design for MIMO channels using the weighted MMSE criterion," IEEE Transactions on Communications , vol. 49, no. 12, pp. 2198–2206, 2001.

[15] J. C. Roh and B. D. Rao, "Channel feedback quantization methods for MISO and MIMO systems," in *Proceedings of the 15th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC '04)*, vol. 2, pp. 805–809, Barcelona, Spain, September 2004.

[16] D. J. Love and R. W. Heath Jr., "Limited feedback unitary precoding for orthogonal space-time block codes," IEEE Transactions on Signal Processing , vol. 53, no. 1, pp. 64–73, 2005.

[17] G. Leus, C. Simon, and N. Khaled, "Spatial multiplexing with linear precoding in time-varying channels with limited

feedback," in *Proceedings of the 14th European Signal Processing Conference (EUSIPCO '06)*, Florence, Italy, September 2006.

[18] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compressing*, Kluwer Academic Publishers, New York, NY, USA, 1995.

[19] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Transactions on Communications , vol. 28, no. 1, pp. 84–95, 1980.

[20] M. Sabin and R. M. Gray, "Global convergence and empirical consistency of the generalized Lloyd algorithm," IEEE Transactions on Information Theory , vol. 32, no. 2, pp. 148–155, 1986.

[21] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, NY, USA, 1991.

[22] K. Huang, B. Mondal, R. W. Heath Jr., and J. G. Andrews, "Multi-antenna limited feedback for temporally-correlated channels: feedback compression," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '06)*, pp. 1–5, San Francisco, Calif, USA, November 2006.

[23] C. Simon and G. Leus, "Adaptive feedback reduction for precoded spatial multiplexing MIMO systems," in *Proceedings of the International ITG/IEEE Workshop on Smart Antennas (WSA '07)*, Vienna, Austria, February 2007.

[24] J. S. Vitter, "Design and analysis of dynamic Huffman codes," Journal of the ACM , vol. 34, no. 4, pp. 825–845, 1987.

[25] D. E. Knuth, "Dynamic Huffman coding," Journal of Algorithms , vol. 6, no. 2, pp. 163–180, 1985.

[26] R. Gallagher, "Variations on a theme by Huffman," IEEE Transactions on Information Theory , vol. 24, no. 6, pp. 668–674, 1978.

[27] R. Samanta and R. W. Heath Jr., "Codebook adaptation for quantized MIMO beamforming systems," in *Conference Record of the 39th Asilomar Conference on Signals, Systems, and Computers (ACSSC '05)*, pp. 376–380, Pacific Grove, Calif, USA, October-November 2005.

[28] S. Zhou, B. Li, and P. Willett, "Recursive and trellis-based feedback reduction for MIMO-OFDM with rate-limited feedback," IEEE Transactions on Wireless Communications , vol. 5, no. 12, pp. 3400–3405, 2006.

[29] A. K. Krishnamurthy, S. C. Ahalt, D. E. Melton, and P. Chen, "Neural networks for vector quantization of speech and images," IEEE Journal on Selected Areas in Communications , vol. 8, no. 8, pp. 1449–1457, 1990.

[30] B. M. Hochwald, T. L. Marzetta, T. J. Richardson, W. Sweldens, and R. Urbanke, "Systematic design of unitary space-time constellations," IEEE Transactions on Information Theory , vol. 46, no. 6, pp. 1962–1973, 2000.

[31] A. van Zelst, "A compact representation of spatial correlation in MIMO radio channels," http://www.avzelst.nl/.