Unsupervised Learning of Adaptive Codebooks for Deep Feedback Encoding in FDD Systems

Nurettin Turan*, Michael Koller*, Samer Bazzi†, Wen Xu†, and Wolfgang Utschick*
*Professur für Methoden der Signalverarbeitung, Technische Universität München, 80290 Munich, Germany
†Huawei Technologies Duesseldorf GmbH, 80992 Munich, Germany
Email: {nurettin.turan,michael.koller,utschick}@tum.de,{samer.bazzi,wen.dr.xu}@huawei.com

***Abstract*—In this work, we propose a joint adaptive codebook construction and feedback generation scheme in frequency division duplex (FDD) systems. Both unsupervised and supervised deep learning techniques are used for this purpose. Based on a recently discovered equivalence of uplink (UL) and downlink (DL) channel state information (CSI) in terms of neural network learning, the codebook and associated deep encoder for feedback signaling is based on UL data only. Subsequently, the feedback encoder can be offloaded to the mobile terminals (MTs) to generate channel feedback there as efficiently as possible, without any training effort at the terminals or corresponding transfer of training and codebook data. Numerical simulations demonstrate the promising performance of the proposed method.**

***Index Terms*—Projected gradient descent, Lloyd-Max quantization, neural network classification, feedback codebook design, frequency division duplexing.**

I. Introduction

Massive multiple-input multiple-output (MIMO) technology is one of the most prominent directions to scale up capacity and throughput in modern communication systems [1]. In particular, the multi-antennas support at the base station (BS) makes simple techniques such as spatial multiplexing and beamforming very efficient regarding the spectrum or the bandwidth utilization. In order to take full advantage of this technology, the BS must have the best possible channel estimation. However, considering the typically stringent delay requirements in wireless mobile communication systems, the CSI has to be acquired in very short regular time intervals. A variety of solution approaches developed for this purpose are based on time division duplex (TDD) mode.

On the other hand, in FDD mode, the BS and the MT transmit in the same time slot but at different frequencies. This breaks the reciprocity between UL CSI and DL CSI, unlike in TDD systems, and makes it difficult for the network operators with FDD licenses to obtain an accurate DL CSI estimate for transmit signal processing [2]. An obvious solution to the problem is to either extrapolate the DL CSI from the estimate of the UL CSI at the BS, or to transfer the DL CSI estimated at the MT to the BS directly or in a highly compressed version. However, the most common solution in practice is to avoid the direct feedback of the CSI and use only selection indices that determine an element from a finite set of channel properties or from a finite set of predefined beamformer or precoder configurations. The latter method is also studied in this paper.

In particular, we propose to construct a finite codebook $\mathcal{Q}$ of different precoder configurations at the BS using an unsupervised learning method. Although the codebook covers precoding matrices for the DL operation, the construction of the codebook is solely based on a set of UL channels $\mathcal{H}^{\mathrm{UL}}$ which have been collected at the BS. Following the codebook construction, a deep neural network (DNN) classifier $f_{\mathrm{DNN}}(\cdot)$ is trained at the BS again solely based on the UL samples, which assigns the index $k^\star$ of the most appropriate precoding matrix to a given DL observation. The exclusive use of UL data for the training of function blocks, which are exclusively intended for the DL channel, follows the recent results in [3], [4]. The proposed approach is then to subsequently offload this deep feedback encoder to each MT in the cell. See Fig. 1 for a sketch of the proposed overall concept.

Thus, the core idea of our scheme is that the neural network encoder trained on UL data at the BS can be applied to DL data without any further adaptation from any mobile device to which the encoder is offloaded. Training on the MT is no longer necessary at all, making it possible to quickly update the encoder on the MT at any time and place with an updated version of the codebook, e.g., when moving from one cell to another or for different locations of the MT in the cell. Based on the presented simulation results, we are able to demonstrate the promising performance of the proposed technique.

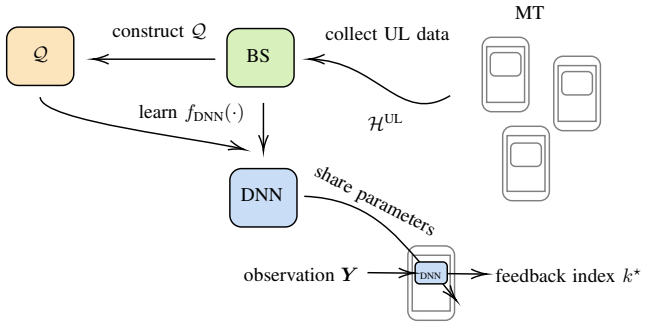


Fig. 1: Structure of the proposed approach with codebook construction and learning of the DNN classifier at the BS, which is then subsequently offloaded to the MT.

II. System Model

The DL received signal of a single user MIMO system can be expressed as $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{y} \in \mathbb{C}^{N_{\mathrm{rx}}}$ is the receive

vector, $\boldsymbol{x} \in \mathbb{C}^{N_{\mathrm{tx}}}$ is the transmit vector sent over the MIMO channel $\boldsymbol{H} \in \mathbb{C}^{N_{\mathrm{rx}} \times N_{\mathrm{tx}}}$, and $\boldsymbol{n} \sim \mathcal{N}_{\mathbb{C}}(\boldsymbol{0}, \sigma_n^2 \boldsymbol{I}_{N_{\mathrm{rx}}})$ denotes the additive white Gaussian noise (AWGN). In this paper, we consider system configurations with $N_{\mathrm{rx}} < N_{\mathrm{tx}}$. If both the transmitter and receiver know the channel perfectly, and assuming input data with Gaussian distribution, the capacity of the MIMO channel is [5], [6]:

$$C = \max_{\boldsymbol{Q} \succeq \boldsymbol{0}, \mathrm{tr}\, \boldsymbol{Q} \leq \rho} \log_2 \det \left(\boldsymbol{I} + \frac{1}{\sigma_n^2} \boldsymbol{H} \boldsymbol{Q} \boldsymbol{H}^{\mathrm{H}} \right), \qquad (1)$$

where $\boldsymbol{Q} \in \mathbb{C}^{N_{\mathrm{tx}} \times N_{\mathrm{tx}}}$ is the transmit covariance matrix. This assumes a transmit vector given by $\boldsymbol{x} = \boldsymbol{Q}^{1/2} \boldsymbol{s}$ with $\mathrm{E}[\boldsymbol{s}\boldsymbol{s}^{\mathrm{H}}] = \boldsymbol{I}_{N_{\mathrm{tx}}}$ [7]. The capacity-achieving transmit covariance matrix $\boldsymbol{Q}^\star$ of the link between the BS and a MT can be found by decomposing the channel into N_{rx} parallel streams and employing water-filling [8].

In FDD systems, channel reciprocity can generally not be assumed, e.g., [7]. For this reason, only the MT could compute the optimal transmit covariance matrix $\boldsymbol{Q}^\star$ if it estimated the DL CSI. This makes some form of feedback from the MT to the BS necessary. Ideally, the user would feed the complete DL CSI back to the BS, which in general is considered to be infeasible. Instead, a low-rate feedback link is used to transmit a small number of M bits back to the BS.

Typically, the M feedback bits are used for encoding an index into a set of covariance matrices. That is, the MT and BS share a codebook $\mathcal{Q} = \{\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_K\}$ of $K = 2^M$ pre-computed transmit covariance matrices, the MT is assumed to estimate the DL channel $\boldsymbol{H}$ and then uses it to determine the best codebook entry $\boldsymbol{Q}_{k^\star}$ in terms of maximum achievable rate, i.e.,

$$k^\star = \operatorname*{arg\,max}_{k \in \{1,\ldots,K\}} \log_2 \det \left(\boldsymbol{I} + \frac{1}{\sigma_n^2} \boldsymbol{H} \boldsymbol{Q}_k \boldsymbol{H}^{\mathrm{H}} \right), \qquad (2)$$

or other practically useful selection criteria. Finally, the feedback consists of the index $k^\star$ encoded by M bits, and the BS employs the transmit covariance matrix $\boldsymbol{Q}_{k^\star}$ for data transmission. Section IV describes an algorithm to obtain a codebook $\mathcal{Q}$.

III. Channel Model and Data Generation

We consider an FDD system below $6\,\mathrm{GHz}$ and assume a frequency gap of $200\,\mathrm{MHz}$ between UL and DL. Version 2.2 of the QuaDRiGa channel simulator [9], [10] is used to generate CSI for the UL and DL scenarios.

We simulate two urban macrocell (UMa) single carrier scenarios: one with $(N_{\mathrm{tx}}, N_{\mathrm{rx}}) = (16, 4)$ and one with $(N_{\mathrm{tx}}, N_{\mathrm{rx}}) = (32, 16)$. In both cases, the UL carrier frequency is $2.53\,\mathrm{GHz}$ and the DL carrier frequency is $2.73\,\mathrm{GHz}$. The BS is equipped with a uniform linear array (ULA) with "3GPP-3D" antennas, and the MT consists of a ULA assuming "omni-directional" antennas. The BS is placed at a height of $25\,\mathrm{m}$ and covers a sector of $120°$, where the minimum distance of the MT location to the BS is $35\,\mathrm{m}$ and the maximum distance to the BS is $500\,\mathrm{m}$. In 80% of the cases, the MT is located indoors at different floor levels, and in the case of outdoor locations the MT's height is $1.5\,\mathrm{m}$ in accordance with [11].

As outlined in [9], many parameters such as path-loss, delay, and angular spreads, path-powers for each subpath, and antenna patterns are different in the UL and DL domain. However, the following parameters are identical in the UL and DL domain: BS location and the MT locations, propagation cluster delays and angles for each multi-path component (MPC), and the spatial consistency of the large scale fading parameters. QuaDRiGa models MIMO channels as

$$\boldsymbol{H} = \sum_{\ell=1}^{L} \boldsymbol{G}_\ell e^{-2\pi j f_c \tau_\ell}, \qquad (3)$$

where ℓ is the path number, and the number of MPCs L depends on whether there is line of sight (LOS), non-line of sight (NLOS), or outdoor-to-indoor (O2I) propagation: $L_{\mathrm{LOS}} = 37$, $L_{\mathrm{NLOS}} = 61$ or $L_{\mathrm{O2I}} = 37$. The carrier frequency is denoted by f_c and the ℓ-th path delay by τ_ℓ. The MIMO coefficient matrix $\boldsymbol{G}_\ell$ consists of one complex entry for each antenna pair, which comprises the attenuation of a path, the antenna radiation pattern weighting, and the polarization [12]. The data is not post-processed: we work with channels including the path gain.

We generate datasets with 30×10^3 channels for both the UL and DL of each scenario. Defining a signal-to-noise ratio (SNR) as $\mathrm{SNR} = E[\|\boldsymbol{H}\boldsymbol{x}\|_2^2] / E[\|\boldsymbol{n}\|_2^2]$ and assuming $\boldsymbol{Q} = \frac{\rho}{N_{\mathrm{tx}}} \boldsymbol{I}$ (i.e., uniform power allocation), it holds: $\mathrm{SNR} = \frac{\rho / N_{\mathrm{tx}} ||\boldsymbol{H}||_F^2}{\sigma_n^2 N_{\mathrm{rx}}}$. In our simulations, we have a noise variance of $\sigma_n^2 = -114\,\mathrm{dBm}$ and a transmit power of $\rho = 15\,\mathrm{dBm}$. We then select all DL channels together with the corresponding UL channels (with the same MT locations) which fall within the SNR range of $[-10\,\mathrm{dB}, 20\,\mathrm{dB}]$ to constitute our datasets. This leads to the following pairs of UL and DL datasets for the two considered scenarios:

$$\mathcal{H}_{4\times16}^{\mathrm{UL}}, \mathcal{H}_{4\times16}^{\mathrm{DL}}, \mathcal{H}_{16\times32}^{\mathrm{UL}}, \text{ and } \mathcal{H}_{16\times32}^{\mathrm{DL}}, \qquad (4)$$

with cardinalities $|\mathcal{H}_{4\times16}^{\mathrm{UL}}| = |\mathcal{H}_{4\times16}^{\mathrm{DL}}| = 18773$ and $|\mathcal{H}_{16\times32}^{\mathrm{UL}}| = |\mathcal{H}_{16\times32}^{\mathrm{DL}}| = 16148$.

The UL channels have a dimension of 16×4 or 32×16, respectively, depending on the scenario. The sets $\mathcal{H}_{4\times16}^{\mathrm{UL}}$ and $\mathcal{H}_{16\times32}^{\mathrm{UL}}$ contain transposed versions of the respective channels, i.e, with dimensions 4×16 or 16×32.

IV. Unsupervised Codebook Design

In this section, we first briefly describe how we can construct a codebook given a set of training channels. Then, we show that we can learn a codebook for the DL domain by using a training set solely consisting of UL channel data.

A. Iterative Llyod Clustering Algorithm

Let $\mathcal{H} = \{\boldsymbol{H}_n\}_{n=1}^{T_{\mathrm{train}}}$ be a set of collected training channel matrices. The goal is to obtain a corresponding codebook $\mathcal{Q} = \{\boldsymbol{Q}_k\}_{k=1}^{K}$ of $K = 2^M$ transmit covariance matrices. To this end, we employ an iterative Lloyd clustering algorithm which alternates between two stages until a convergence criterion is

met [13]. We write $\{\boldsymbol{Q}_k^{(i)}\}_{k=1}^K$ for the codebook in iteration i. Further, we define the spectral efficiency

$$r(\boldsymbol{H}, \boldsymbol{Q}) = \log_2 \det \left(\boldsymbol{I} + \frac{1}{\sigma_n^2} \boldsymbol{H}\boldsymbol{Q}\boldsymbol{H}^{\mathrm{H}} \right) \tag{5}$$

to be the cost criterion of interest. Then, the two stages of iteration i are expressed as follows:

1) Divide the training set $\mathcal{H}$ into K clusters $\mathcal{V}_k^{(i)}$:

$$\mathcal{V}_k^{(i)} = \{\boldsymbol{H} \in \mathcal{H} \mid r(\boldsymbol{H}, \boldsymbol{Q}_k^{(i)}) \geq r(\boldsymbol{H}, \boldsymbol{Q}_j^{(i)}), k \neq j\}. \tag{6}$$

2) Find new covariance matrices or update the so called "cluster centers":

$$\boldsymbol{Q}_k^{(i+1)} = \underset{\boldsymbol{Q} \succeq \boldsymbol{0}}{\arg\max} \frac{1}{|\mathcal{V}_k^{(i)}|} \sum_{\boldsymbol{H} \in \mathcal{V}_k^{(i)}} r(\boldsymbol{H}, \boldsymbol{Q}) \tag{7}$$

$$\text{subject to} \quad \operatorname{trace}(\boldsymbol{Q}) \leq \rho \quad \text{and} \quad \operatorname{rank} \boldsymbol{Q} \leq N_{\mathrm{rx}}.$$

Whereas (6) is easily obtained by computing the spectral efficiency (5) of each training channel matrix with every codebook element, we solve the optimization problem (7) with projected gradient descent (PGD) as it can be found in [14]. PGD starts from an initialization—in our case it is the scaled identity matrix—and updates $\boldsymbol{Q}$ using an unconstrained gradient step:

$$\boldsymbol{Q} \leftarrow \boldsymbol{Q} + \alpha \boldsymbol{g}_{\boldsymbol{Q}}. \tag{8}$$

The gradient $\boldsymbol{g}_{\boldsymbol{Q}}$ of the objective function in (7) with respect to $\boldsymbol{Q}$, namely

$$\boldsymbol{g}_{\boldsymbol{Q}} = \frac{1}{\sigma_n^2 \ln(2)} \sum_{\boldsymbol{H} \in \mathcal{V}_k^{(i)}} \boldsymbol{H}^{\mathrm{H}} \left(\boldsymbol{I} + \frac{1}{\sigma_n^2} \boldsymbol{H}\boldsymbol{Q}\boldsymbol{H}^{\mathrm{H}} \right)^{-1} \boldsymbol{H}, \tag{9}$$

leaves the updated $\boldsymbol{Q}$ still positive-semidefinite for arbitrary step sizes $\alpha \geq 0$. The other constraints in (7) are thereafter enforced via a projection step of $\boldsymbol{Q}$ onto the constraint set. The projection is done via water-filling. The two stages consisting of unconstrained gradient update followed by projection are iterated until convergence. In our simulations, an Armijo rule controls the step size. More details about this version of PGD can be found in [14].

B. Codebook Construction—UL versus DL Data

We split the two sets $\mathcal{H}_{4\times16}^{\mathrm{UL}}$, $\mathcal{H}_{4\times16}^{\mathrm{DL}}$ into a training set with $T_{\mathrm{train}} = 10^4$ samples, and the remaining samples constitute an evaluation set:

$$\mathcal{H}_{4\times16}^{\mathrm{UL,train}}, \mathcal{H}_{4\times16}^{\mathrm{UL,eval}}, \mathcal{H}_{4\times16}^{\mathrm{DL,train}}, \text{ and } \mathcal{H}_{4\times16}^{\mathrm{DL,eval}}. \tag{10}$$

However, the UL evaluation set $\mathcal{H}_{4\times16}^{\mathrm{UL,eval}}$ is not relevant for our considerations and the following transmit strategies are always evaluated on $\mathcal{H}_{4\times16}^{\mathrm{DL,eval}}$.

Since the distances of the MT locations to the BS range from $35\,\mathrm{m}$ to $500\,\mathrm{m}$ and since we cover an SNR range of $[-10\,\mathrm{dB}, 20\,\mathrm{dB}]$, the quality of the channels varies greatly. For this reason, we display the distribution of the spectral efficiencies of the channels in the set $\mathcal{H}_{4\times16}^{\mathrm{DL,eval}}$ for various transmit strategies via box plots. Every boxplot in Fig. 2 highlights the median, the first, and the third quartile, and the whiskers represent an interquartile range of 1.5.

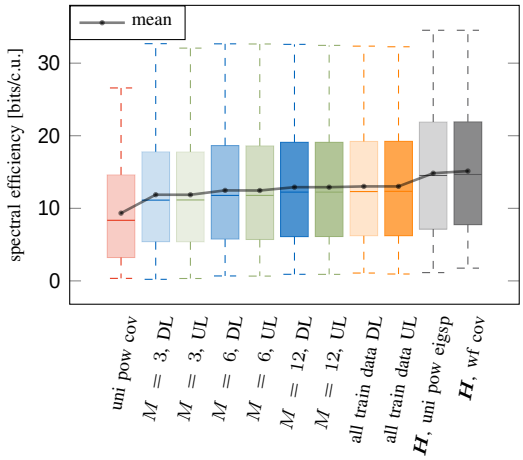


Fig. 2: Spectral efficiencies corresponding to codebooks of different sizes M, constructed using either UL or DL training data.

i) The boxplot labled "`uni pow cov`" represents uniform power allocation where the transmit covariance matrix is simply given by $\boldsymbol{Q} = \frac{\rho}{N_{\mathrm{tx}}} \boldsymbol{I}$. In this case, no channel knowledge or codebook is used.

ii) As a performance upper bound, Fig. 2 shows the spectral efficiency distribution obtained using the optimal transmit strategy, where the capacity-achieving transmit covariance matrix is computed for every channel via water-filling ("$\boldsymbol{H}$, `wf cov`").

iii) Moreover, Fig. 2 depicts the "$\boldsymbol{H}$, `uni pow eigsp`" transmit strategy, where a transmit covariance matrix is calculated by allocating equal power on the eigenvectors of the channel. That is, the channel is decomposed into N_{rx} parallel streams and $\frac{\rho}{N_{\mathrm{rx}}}$ power is allocated on each stream. Note that this and the optimal approach are infeasible because the BS would require full knowledge of the DL channel.

iv) Furthermore, Fig. 2 contains codebook based transmit strategies with $M \in \{3, 6, 12\}$ bit. Recall, that a feedback index is selected via (2) using the DL channel. For reasons of comparability, we use either UL or DL channels as training data, i.e, either $\mathcal{H}_{4\times16}^{\mathrm{UL,train}}$ or $\mathcal{H}_{4\times16}^{\mathrm{DL,train}}$, to construct the codebooks with the procedure described in Section IV-A. Already with a codebook of $M = 3$ bits, higher spectral efficiencies can be achieved as compared to the "`uni pow cov`" transmit strategy.

v) With an increasing codebook size, the spectral efficiency increases and almost reaches the performance of the "`all train data UL`" or "`all train data DL`" transmit strategies, which constitute a natural upper bound on codebook based approaches with the respective UL or DL dataset, and which are computed as follows. For each channel sample in $\mathcal{H}_{4\times16}^{\mathrm{UL,train}}$ or $\mathcal{H}_{4\times16}^{\mathrm{DL,train}}$, the corresponding capacity achieving transmit covariance matrix is calculated and is used as a codebook entry. Thus, in this case, we have a huge codebook with $T_{\mathrm{train}} = 10^4$ entries.

Remarkably, the considered codebooks show comparable

performance in the two cases where either UL or DL training data is used for the codebook construction. In particular, the boxplots in Fig. 2 show very similar distributions of the spectral efficiencies of the corresponding UL/DL pairs. Whereas the instantaneous UL and DL channel realizations may differ, one may now reasonably conjecture that the totality of the collected UL and DL channels share relevant statistical properties. This conjecture was in particular investigated in [3], [4] for a large range of frequency gaps between the UL and DL domain and corroborated with the help of hypothesis tests.

C. Codebook Performance with Estimated Channels

In this section, we investigate how the distribution of the spectral efficiencies is affected when a codebook entry is selected with the help of estimated channels.

In the pilot transmission phase, the DL received signal is:

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{P} + \boldsymbol{N} \in \mathbb{C}^{N_{\mathrm{rx}} \times n_p}, \tag{11}$$

where n_p is the number of transmitted pilots, $\boldsymbol{N} = [\boldsymbol{n}_1, \boldsymbol{n}_2, \ldots, \boldsymbol{n}_{n_p}] \in \mathbb{C}^{N_{\mathrm{rx}} \times n_p}$ collects noise samples with

$$\boldsymbol{n}_p \sim \mathcal{N}_{\mathbb{C}}(\boldsymbol{0}, \sigma_n^2 \boldsymbol{I}_{N_{\mathrm{rx}}}), \tag{12}$$

$p \in \{1, 2, \ldots, n_p\}$, and where the pilot matrix $\boldsymbol{P}$ is given by

$$\boldsymbol{P} = \sqrt{\frac{\rho}{N_t}} \begin{bmatrix} 1 & 1 & \ldots & 1 \\ 1 & W_{n_p} & \ldots & W_{n_p}^{n_p-1} \\ \vdots & \vdots & & \vdots \\ 1 & W_{n_p}^{N_{\mathrm{tx}}-1} & \ldots & W_{n_p}^{(N_{\mathrm{tx}}-1)(n_p-1)} \end{bmatrix} \in \mathbb{C}^{N_{\mathrm{tx}} \times n_p}, \tag{13}$$

with $W_{n_p} = e^{\mathrm{j}2\pi/n_p}$. The pilot matrix $\boldsymbol{P}$ is a submatrix of the discrete Fourier transform (DFT) matrix. If $n_p \geq N_{\mathrm{tx}}$, the least squares (LS) channel estimate in the pilot transmission phase can be obtained as $\boldsymbol{H}_{\mathrm{LS}} = \boldsymbol{Y}\boldsymbol{P}^{\dagger}$ with the pseudoinverse $\boldsymbol{P}^{\dagger} = \boldsymbol{P}^{\mathrm{H}}(\boldsymbol{P}\boldsymbol{P}^{\mathrm{H}})^{-1}$.

As an alternative to least squares, it is common to assume that channels exhibits a certain structure. Using the vectorized channel $\boldsymbol{h} = \mathrm{vec}(\boldsymbol{H})$, we express this as $\boldsymbol{h} \approx \boldsymbol{D}\boldsymbol{t}$, where $\boldsymbol{D} = \boldsymbol{D}_{\mathrm{rx}} \otimes \boldsymbol{D}_{\mathrm{tx}}$ is a *dictionary* with oversampled DFT matrices $\boldsymbol{D}_{\mathrm{rx}}$ and $\boldsymbol{D}_{\mathrm{tx}}$ (cf., e.g., [15]), because we have ULAs at the transmitter and receiver side. A compressive sensing algorithm like orthogonal matching pursuit (OMP) [16] can now be used to obtain a sparse vector $\boldsymbol{t}$, and the channel estimate then computes to $\mathrm{vec}(\boldsymbol{H}_{\mathrm{OMP}}) = \boldsymbol{D}\boldsymbol{t}$. Since the sparsity order is not known but the algorithm's performance crucially depends on it, we use a genie-aided approach to obtain an upper bound on the performance. To this end, we use the true channel to choose the optimal sparsity for OMP.

Fig. 3(a) shows the spectral efficiency distributions of the scenario with $N_{\mathrm{tx}} = 16$ and $N_{\mathrm{rx}} = 4$ again. The codebooks are the same as in the previous section ($M = 6$)—trained using either $\mathcal{H}_{4\times16}^{\mathrm{UL,train}}$ or $\mathcal{H}_{4\times16}^{\mathrm{DL,train}}$—, and the same evaluation set $\mathcal{H}_{4\times16}^{\mathrm{DL,eval}}$ is used.

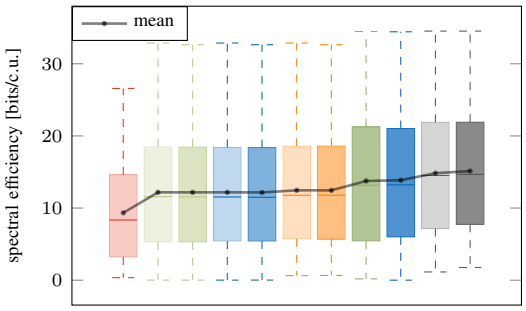
i) The boxplots labeled "$\boldsymbol{H}$, `DL cb`" or "$\boldsymbol{H}$, `UL cb`" select the feedback index in (2) using the true DL channels $\boldsymbol{H}$, i.e., assuming perfect CSI. Consequently, this is an upper bound for index selection based on channel estimates.

ii) After receiving $n_p = 16$ pilots, the MT estimates the DL channel and uses this estimated channel to select a codebook entry via (2). The corresponding boxplots are denoted by "$\boldsymbol{H}_{\mathrm{LS}}$, `DL cb`" or "$\boldsymbol{H}_{\mathrm{LS}}$, `UL cb`" and "$\boldsymbol{H}_{\mathrm{OMP}}$, `DL cb`" or "$\boldsymbol{H}_{\mathrm{OMP}}$, `UL cb`", respectively, depending on whether LS or OMP channel estimates are used and whether the codebooks are constructed with UL or DL data. It can be seen that irrespective of whether we have constructed the codebook with DL or UL training data, the distributions of the spectral efficiencies hardly differ at all. OMP seems to perform slightly better than LS.

iii) Further, we depict approaches ("$\boldsymbol{H}_{\mathrm{LS}}$, `wf cov`" and "$\boldsymbol{H}_{\mathrm{OMP}}$, `wf cov`"), where we use the estimated channel to calculate a transmit covariance matrix with water-filling. These approaches are practically infeasible, since the BS would require knowledge of the estimated channels. OMP is again slightly better than LS.

iv) As a performance upper bound, we depict ("$\boldsymbol{H}$, `wf cov`"), where the optimal transmit strategy is employed. The approach "$\boldsymbol{H}$, `uni pow eigsp`" is again the transmit strategy, where a transmit covariance matrix is calculated by allocating equal power on the eigenvectors of the channel.

(a) $N_{\mathrm{tx}} = 16, N_{\mathrm{rx}} = 4, n_{\mathrm{p}} = 16$



(b) $N_{\mathrm{tx}} = 32, N_{\mathrm{rx}} = 16, n_{\mathrm{p}} = 32$

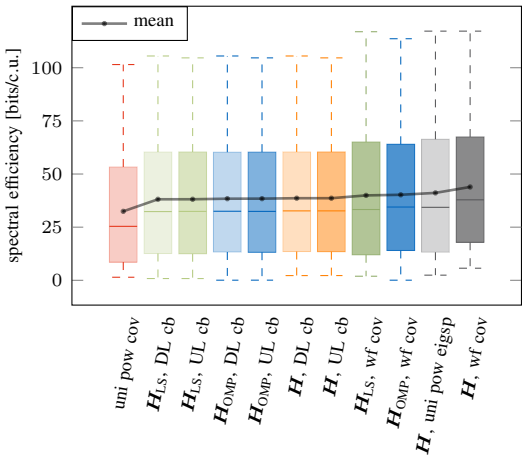


Fig. 3: Spectral efficiencies of different transmit strategies in two different scenarios. Codebooks with $M = 6$ bits are either constructed using UL or DL training data: "`UL cb`" or "`DL cb`". The LS and OMP channel estimates are $\boldsymbol{H}_{\mathrm{LS}}$ and $\boldsymbol{H}_{\mathrm{OMP}}$, and $\boldsymbol{H}$ refers to the true DL channel. The "`wf`" labels correspond to transmission with water-filling covariance matrices where complete channel knowledge would be necessary.

Recall, that these two approaches are also infeasible because the BS would require full knowledge of the DL channel.

Similar observations can be made in Fig. 3(b) where we have a different scenario: $N_{\text{tx}} = 32$, $N_{\text{rx}} = 16$, and $n_p = 32$. As before, we split the data into training sets $\mathcal{H}^{\text{UL,train}}_{16\times 32}$ and $\mathcal{H}^{\text{DL,train}}_{16\times 32}$ with 10^4 samples each and an evaluation set $\mathcal{H}^{\text{DL,eval}}_{16\times 32}$.

V. Deep Neural Network for Feedback Encoding

Instead of using the two stage process of first estimating a channel and then selecting a codebook entry via (2), we propose to find a function which directly maps from the noisy observations $\boldsymbol{Y}$ to the feedback index $k^\star$:

$$f : \mathbb{C}^{N_{\text{rx}}\times n_p} \to \{1, \cdots, 2^M\}, \quad \boldsymbol{Y} \mapsto f(\boldsymbol{Y}) = k^\star. \tag{14}$$

This is readily interpreted as a classification task. In particular, we propose to use a DNN f_{DNN} to approximate the function f and, thus, to perform the classification task.

The codebooks ($M = 6$) are the same as in the previous section—trained using either $\mathcal{H}^{\text{UL,train}}_{N_{\text{rx}}\times N_{\text{tx}}}$ or $\mathcal{H}^{\text{DL,train}}_{N_{\text{rx}}\times N_{\text{tx}}}$ for the two considered scenarios. We further use the four training data sets to generate labeled data for four different DNN approaches: either UL or DL with either $(N_{\text{tx}}, N_{\text{rx}}) = (16, 4)$ or $(N_{\text{tx}}, N_{\text{rx}}) = (32, 16)$. The labels are given by the optimal codebook indices determined via (2). The input/output pairs $\{(\boldsymbol{Y}_n, k^\star_n)\}_{n=1}^{T_{\text{train}}}$ then form the DNN training data sets.

Of the remaining data in $\mathcal{H}^{\text{UL}}_{N_{\text{rx}}\times N_{\text{tx}}}$ or $\mathcal{H}^{\text{DL}}_{N_{\text{rx}}\times N_{\text{tx}}}$ for the two scenarios, 2500 samples are split off to form four validation data sets, and, finally, the rest is used for test sets. Thus, the test sets consist of 6273 samples for $(N_{\text{tx}}, N_{\text{rx}}) = (16, 4)$ and 3648 samples for $(N_{\text{tx}}, N_{\text{rx}}) = (32, 16)$.

A. DNN Structure and Training Procedure

The considered DNN has two input "channels": one for the real part $\Re(\boldsymbol{Y})$ of the observation $\boldsymbol{Y}$, and one for the imaginary part $\Im(\boldsymbol{Y})$. In a first step, these two parts are normalized w.r.t. the Frobenius norm. We employ random search [17] to determine the hyperparameters (explained next) of the DNN.

The first modules of the DNN are convolutional modules, which consist of a convolutional layer, a batch normalization layer, and an activation function. We have D_{CM} such modules, where D_{CM} is randomly chosen from $[8, 20]$. Each of the convolutional layers has D_{K} kernels, where D_{K} is randomly chosen from $[32, 64]$. In the setting $(N_{\text{tx}}, N_{\text{rx}}) = (16, 4)$, we then flatten the features and obtain a vector of size $N_f = D_{\text{K}} N_{\text{rx}} n_p$. In the setting $(N_{\text{tx}}, N_{\text{rx}}) = (32, 16)$, two-dimensional max pooling by a factor of two is applied prior to flattening, yielding a $N_f = D_{\text{K}} \frac{N_{\text{rx}}}{2} \frac{n_p}{2}$-dimensional vector. Subsequently, we have fully connected layers—each followed by batch normalization and activation function—, decreasing the dimension from N_f to $512 \to 256 \to 128 \to 64 \to 64 = 2^M$.

The loss function is cross-entropy. We train for 300 epochs with a 5 epochs early stopping criterion. The activation function is the same in each layer, but randomly chosen from {ReLu, sigmoid, PReLU, Leaky ReLU, tanh, swish}. Further random parameters are: batch size $\in [20, 1000]$, learning rate $\in [10^{-6}, 10^{-3}]$, L_1 regularization $\in [10^{-6}, 10^{-3}]$, L_2 regularization $\in [10^{-6}, 10^{-3}]$, and exponential learning rate decay $\in [0.94, 1]$. The optimizer is Adam [18]. We run 100 random searches and pick out the best DNN.

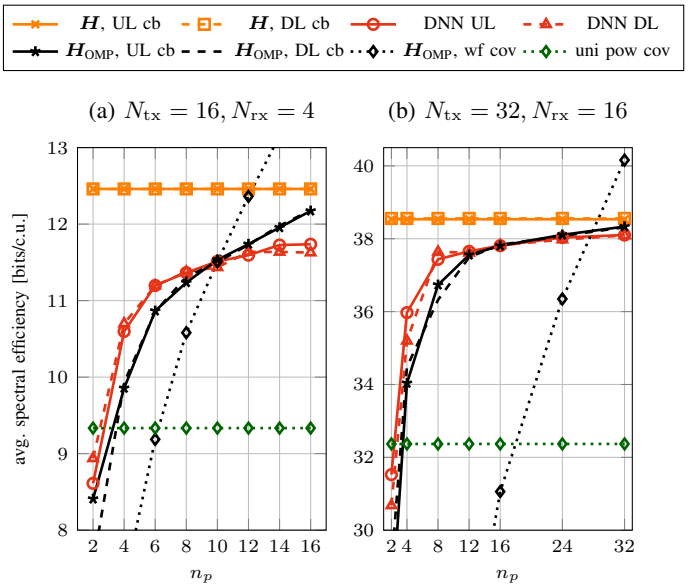


Fig. 4: Average spectral efficiencies in two different scenarios. Codebooks with $M = 6$ bits are constructed with UL or DL data and a DNN trained with UL or DL data is used for feedback encoding.

Fig. 4 depicts spectral efficiencies over the number of pilots $n_p \leq N_{\text{tx}}$:

i) The curves labeled "$\boldsymbol{H}$, `DL cb`" and "$\boldsymbol{H}$, `UL cb`" select the feedback index with (2) using the true DL channels $\boldsymbol{H}$ and thus represent an upper bound for the curves labeled "$\boldsymbol{H}_{\text{OMP}}$, `DL cb`" and "$\boldsymbol{H}_{\text{OMP}}$, `UL cb`", where a codebook entry—constructed with DL or UL training data—is selected via (2) using OMP channel estimates. There is again not much difference between the codebooks constructed using UL or DL data. Interestingly, the curve "$\boldsymbol{H}_{\text{OMP}}$, `wf cov`", where we use an OMP channel estimate to calculate a transmit covariance matrix with water-filling, is worse than the codebook based approaches "$\boldsymbol{H}_{\text{OMP}}$, `DL cb`" and "$\boldsymbol{H}_{\text{OMP}}$, `UL cb`", if only a few pilots are transmitted. In contrast, the use of a codebook with pre-computed transmit covariance matrices proves advantageous when only a small number of pilots are transmitted. Recall, that the "$\boldsymbol{H}_{\text{OMP}}$, `wf cov`" approach is practically infeasible, since the BS would require knowledge of the estimated channel.

ii) The proposed DNN approach is labeled "`DNN DL`" and "`DNN UL`", depending on whether DL or UL training data has been used. Since the DNN directly maps the observations to a feedback index $k^\star$, the MT does not need to know the codebook, which is in contrast to the other approaches. Fig. 4 shows that for small and moderate numbers of pilots, the DNN outperforms the approaches "$\boldsymbol{H}_{\text{OMP}}$, `DL cb`" and "$\boldsymbol{H}_{\text{OMP}}$, `UL cb`", where first a channel is estimated and then a feedback index is chosen from the codebook via (2). Remarkably, the DNNs trained with UL or DL data perform equally well.

VI. Conclusion

We showed that in FDD systems, we can learn an adaptive codebook directly at the BS by gathering UL CSI. Further, a DNN feedback encoder can also be trained at the BS with the help of the codebook and UL CSI. The feedback encoder's weights and biases can be offloaded to all MTs within the coverage area of the BS. A MT can then directly select a feedback index from noisy observations via this DNN such that no channel estimation or knowledge of the codebook are necessary. The overall concept and the fact that both the codebooks and the feedback classifiers are learned exclusively at the BS makes the use of many different adaptive codebooks in the cell conceivable.

References

[1] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. on Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, 2010.

[2] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, "Massive MIMO is a reality–What is next? five promising research directions for antenna arrays," *Digital Signal Processing*, vol. 94, pp. 3 – 20, 2019, special Issue on Source Localization in Massive MIMO.

[3] W. Utschick, V. Rizzello, M. Joham, Z. Ma, and L. Piazzi, "Learning the CSI recovery in FDD systems," 2021, https://arxiv.org/abs/2104.01322.

[4] V. Rizzello and W. Utschick, "Learning the CSI denoising and feedback without supervision," 2021, https://arxiv.org/abs/2104.05002.

[5] A. Goldsmith, *Wireless Communications*. Cambridge Univ. Press, 2005.

[6] A. Goldsmith, S. Jafar, N. Jindal, and S. Vishwanath, "Capacity limits of MIMO channels," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 5, pp. 684–702, 2003.

[7] D. J. Love, R. W. Heath, V. K. N. Lau, D. Gesbert, B. D. Rao, and M. Andrews, "An overview of limited feedback in wireless communication systems," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 8, pp. 1341–1365, 2008.

[8] I. E. Telatar, "Capacity of multi-antenna gaussian channels," *Eur. Trans. on Telecommun.*, vol. 10, pp. 585–595, 1999.

[9] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "Quadriga: A 3-d multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, 2014.

[10] S. Jaeckel, L. Raschkowski, K. Börner, L. Thiele, F. Burkhardt, and E. Eberlein, "Quadriga: Quasi deterministic radio channel generator, user manual and documentation," Fraunhofer Heinrich Hertz Institute, Tech. Rep., v2.2.0, 2019.

[11] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), TR 38.901 V16.1.0, 2019.

[12] M. Kurras, S. Dai, S. Jaeckel, and L. Thiele, "Evaluation of the spatial consistency feature in the 3gpp geometry-based stochastic channel model," in *2019 IEEE Wireless Commun. and Netw. Conf. (WCNC)*, 2019, pp. 1–6.

[13] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[14] R. Hunger, D. A. Schmidt, M. Joham, and W. Utschick, "A general covariance-based optimization framework using orthogonal projections," in *2008 IEEE 9th Workshop on Signal Processing Advances in Wireless Communications*, 2008, pp. 76–80.

[15] A. Alkhateeb, G. Leus, and R. W. Heath, "Compressed sensing based multi-user millimeter wave systems: How many measurements are needed?" in *2015 IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, 2015, pp. 2909–2913.

[16] M. Gharavi-Alkhansari and T. Huang, "A fast orthogonal matching pursuit algorithm," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, vol. 3, 1998, pp. 1389–1392 vol.3.

[17] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. null, p. 281–305, Feb. 2012.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Int. Conf. on Learn. Representations*, 12 2014.