Learning the MMSE Channel Estimator

David Neumann, Thomas Wiese, *Student Member, IEEE*, and Wolfgang Utschick, *Senior Member, IEEE*

***Abstract*—We present a method for estimating conditionally Gaussian random vectors with random covariance matrices, which uses techniques from the field of machine learning. Such models are typical in communication systems, where the covariance matrix of the channel vector depends on random parameters, e.g., angles of propagation paths. If the covariance matrices exhibit certain Toeplitz and shift-invariance structures, the complexity of the minimum mean squared error (MMSE) channel estimator can be reduced to $\mathcal{O}(M \log M)$ floating point operations, where M is the channel dimension. While in the absence of structure the complexity is much higher, we obtain a similarly efficient (but suboptimal) estimator by using the MMSE estimator of the structured model as a blueprint for the architecture of a neural network. This network learns the MMSE estimator for the unstructured model, but only within the given class of estimators that contains the MMSE estimator for the structured model. Numerical simulations with typical spatial channel models demonstrate the generalization properties of the chosen class of estimators to realistic channel models.**

***Index Terms*—Channel estimation, MMSE estimation, machine learning, neural networks, spatial channel model.**

I. Introduction

ACCURATE channel estimation is a major challenge in the next generation of wireless communication networks, e.g., in cellular massive MIMO [1], [2] or millimeter-wave [3], [4] networks. In setups with many antennas and low signal to noise ratios (SNRs), errors in the channel estimates are particularly devastating, because the array gain cannot be fully realized. Since a large array gain is essential in such setups, there is currently a lot of research going on concerning the modeling and verification of massive MIMO and/or millimeter wave channels [5], [6] and the question how these models can aid channel estimation [7].

For complicated stochastic models, the minimum mean squared error (MMSE) estimates of the channel cannot be calculated in closed form. A common strategy to obtain computable estimators is to restrict the estimator to a certain class of functions and then find the best estimator in that class. For example, we could restrict the estimator to the class of linear operators. The linear MMSE (LMMSE) estimator is then represented by the optimal linear operator, i.e., the linear operator that minimizes the mean squared error (MSE). In some special cases, the matrix that represents the optimal linear estimator can be calculated in closed form; in other cases, it has to be calculated numerically. We know that the LMMSE estimator is the MMSE estimator for jointly Gaussian distributed random variables. Nonetheless, it is often used in non-Gaussian settings and performs well for all kinds of distributions that are not too different from a Gaussian.

In the same spirit, we present a class of low-complexity channel estimators, which contain a convolutional neural network (CNN) as their core component. These CNN-estimators are composed of convolutions and some simple nonlinear operations. The CNN-MMSE estimator is then the CNN-estimator with optimal convolution kernels such that the resulting estimator minimizes the MSE. These optimal kernels have to be calculated numerically, and this procedure is called *learning*. Just as the LMMSE estimator is optimal for jointly Gaussian random variables, the CNN-MMSE estimator is optimal for a specific idealized channel model (essentially a single-path model as described by the ETSI 3rd Generation Partnership Project (3GPP) [8]). In numerical simulations, we find that the CNN-MMSE estimator works fine for the channel models proposed by the 3GPP, even though these violate the assumptions under which the CNN-MMSE estimator is optimal.

Once we have learned the CNN-MMSE estimator from real or simulated channel realizations, the computational complexity required to calculate a channel estimate is only $\mathcal{O}(M \log M)$ floating point operations (FLOPS). Despite this low complexity, the performance of the CNN-MMSE estimator does not trail far behind that of the unrestricted MMSE estimator, which is very complex to compute. Since the learning procedure is performed off-line, it does not add to the complexity of the estimator.

One assumption of the idealized channel model mentioned above is that the covariance matrices have Toeplitz structure. This assumption has also motivated other researchers to propose estimators that exploit this structure. For example, in [4] and [9], methods from the area of compressive sensing are used to approximate the channel vector as a linear combination of k steering vectors where k is much smaller than the number of antennas M. With these methods, a complexity of $\mathcal{O}(M \log M)$ floating point operations can be achieved if efficient implementations are used. Although of similar complexity, the proposed CNN-MMSE estimator significantly outperforms the compressive-sensing-based estimators in our simulations.

In [10], the maximum likelihood estimator of the channel covariance matrix within the class of all positive semi-definite Toeplitz matrices is constructed. This estimated covariance

Manuscript received June 14, 2017; revised October 27, 2017; accepted January 7, 2018. Date of publication January 30, 2018; date of current version April 19, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Marco Lops. This paper was presented at the 21st International ITG Workshop on Smart Antennas, Berlin, Germany, March 15–17, 2017. *(Corresponding author: David Neumann.)*

The authors are with the Professur für Methoden der Signalverarbeitung, Technische Universität München, München 80290, Germany (e-mail: d.neumann@tum.de; thomas.wiese@tum.de; utschick@tum.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2018.2799164

matrix is then used to estimate the actual channel. However, even the low-complexity version of this covariance matrix estimator relies on the solution of a convex program with M variables, i.e., its complexity is polynomial in the number of antennas. There also exists previous work on learning-based channel estimation [11]–[14], but with completely different focus in terms of system model and estimator design.

In summary, our main contributions are the following:

- We derive the MMSE channel estimator for conditionally normal channel models, i.e., the channel is normally distributed given a set of parameters, which are also modelled as random variables.
- We show how the complexity of the MMSE estimator can be reduced to $\mathcal{O}(M \log M)$ if the channel covariance matrices are Toeplitz and have a shift-invariance structure.
- We use the structure of this MMSE estimator to define the CNN estimators, which have $\mathcal{O}(M \log M)$ complexity.
- We describe how the variables of the neural network can be optimized/learned for a general channel model using stochastic gradient methods.
- We introduce a hierarchical learning algorithm, which helps to avoid local optima during the learning procedure.

A. Notation

The transpose and conjugate transpose of $\boldsymbol{X}$ are denoted by $\boldsymbol{X}^{\mathrm{T}}$ and $\boldsymbol{X}^{\mathrm{H}}$, respectively. The trace of a square matrix $\boldsymbol{X}$ is denoted by $\mathrm{tr}(\boldsymbol{X})$ and the Frobenius norm of $\boldsymbol{X}$ is denoted by $\|\boldsymbol{X}\|_F$. Two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{C}^{M \times M}$ are asymptotically equivalent, which we denote by $\boldsymbol{A} \asymp \boldsymbol{B}$, if

$$\lim_{M \to \infty} \|\boldsymbol{A} - \boldsymbol{B}\|_F^2 / M = 0. \tag{1}$$

We write $\exp(\boldsymbol{x})$ and $|\boldsymbol{x}|^2$ to denote element-wise application of $\exp(\cdot)$ and $|\cdot|^2$ to the elements of $\boldsymbol{x}$. The kth entry of the vector $\boldsymbol{x}$ is denoted by $[\boldsymbol{x}]_k$; similarly, for a matrix $\boldsymbol{X}$, we write $[\boldsymbol{X}]_{mn}$ to denote the entry in the mth row and nth column. The circular convolution of two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{C}^M$ is denoted by $\boldsymbol{a} * \boldsymbol{b} \in \mathbb{C}^M$. Finally, $\mathrm{diag}(\boldsymbol{x})$ denotes the square matrix with the entries of the vector $\boldsymbol{x}$ on its diagonal and $\mathrm{vec}(\boldsymbol{X})$ is the vector obtained by stacking all columns of the matrix $\boldsymbol{X}$ into a single vector.

II. Conditionally Normal Channels

We consider a base station with M antennas, which receives uplink training signals from a single-antenna terminal. We assume a frequency-flat, block-fading channel, i.e., we get independent observations in each coherence interval. After correlating the received training signals with the pilot sequence transmitted by the mobile terminal, we get observations of the form

$$\boldsymbol{y}_t = \boldsymbol{h}_t + \boldsymbol{z}_t, \quad t = 1, \ldots, T \tag{2}$$

with the channel vectors $\boldsymbol{h}_t$ and additive Gaussian noise $\boldsymbol{z}_t \sim \mathcal{N}_{\mathbb{C}}(\boldsymbol{0}, \boldsymbol{\Sigma})$. For the major part of this work, we assume that the noise covariance is a scaled identity $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$ with *known* σ^2. The channel vectors are assumed to be conditionally Gaussian distributed given a set of parameters $\boldsymbol{\delta}$, i.e., $\boldsymbol{h}_t | \boldsymbol{\delta} \sim \mathcal{N}_{\mathbb{C}}(\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\delta}})$. In contrast to the fast-fading channel vectors, the covariance matrix $\boldsymbol{C}_{\boldsymbol{\delta}}$ is assumed to be constant over the T channel coherence intervals. That is, T denotes the coherence interval of the covariance matrix in number of channel coherence intervals.

The parameters, which describe, for example, angles of propagation paths, are also considered as random variables with distribution $\boldsymbol{\delta} \sim p(\boldsymbol{\delta})$, which is known. In summary, we have

$$\boldsymbol{y}_t | \boldsymbol{h}_t \sim \mathcal{N}_{\mathbb{C}}(\boldsymbol{h}_t, \boldsymbol{\Sigma}) \tag{3}$$

with *known* noise covariance matrix $\boldsymbol{\Sigma}$ and hierarchical prior

$$\boldsymbol{h}_t | \boldsymbol{\delta} \sim \mathcal{N}_{\mathbb{C}}(\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\delta}}), \quad \boldsymbol{\delta} \sim p(\boldsymbol{\delta}). \tag{4}$$

Example: Conditionally normal channels appear in typical channel models for communication scenarios, e.g., in those defined by the 3GPP for cellular networks [8]. There, the covariance matrices are of the form

$$\boldsymbol{C}_{\boldsymbol{\delta}} = \int_{-\pi}^{\pi} g(\theta; \boldsymbol{\delta}) \boldsymbol{a}(\theta) \boldsymbol{a}(\theta)^{\mathrm{H}} d\theta, \tag{5}$$

where $g(\theta; \boldsymbol{\delta}) \geq 0$ is a power density function corresponding to the parameters $\boldsymbol{\delta}$ and where $\boldsymbol{a}(\theta)$ denotes the array manifold vector of the antenna array at the base station for an angle θ. As an example, in the 3GPP urban micro and urban macro scenarios, $g(\theta; \boldsymbol{\delta})$ is a superposition of several scaled probability density functions of a Laplace-distributed random variable with standard deviations 2° and 5°, respectively. The Laplace density models the scattering of the received power around the center of the propagation path. Its standard deviation is denoted as the per-path angular spread.

III. MMSE Channel Estimation

Our goal is to estimate $\boldsymbol{h}_t$ for each t given all observations $\boldsymbol{Y} = \left[\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T\right]$ and knowledge of the model (3), (4). For a fixed parameter $\boldsymbol{\delta}$, the MMSE estimator can be given analytically, since conditioned on $\boldsymbol{\delta}$, the observation $\boldsymbol{y}_t$ is jointly Gaussian distributed with the channel vector $\boldsymbol{h}_t$. Also, given the parameters $\boldsymbol{\delta}$, the observations of different coherence intervals are independent. The conditional MMSE estimate of the channel vector $\boldsymbol{h}_t$ is [7], [15]

$$\mathrm{E}[\boldsymbol{h}_t | \boldsymbol{Y}, \boldsymbol{\delta}] = \boldsymbol{W}_{\boldsymbol{\delta}} \boldsymbol{y}_t \tag{6}$$

with

$$\boldsymbol{W}_{\boldsymbol{\delta}} = \boldsymbol{C}_{\boldsymbol{\delta}} (\boldsymbol{C}_{\boldsymbol{\delta}} + \boldsymbol{\Sigma})^{-1}. \tag{7}$$

Given the parameters $\boldsymbol{\delta}$, the estimate of $\boldsymbol{h}_t$ only depends on $\boldsymbol{y}_t$. Since the parameters $\boldsymbol{\delta}$ and, thus, the covariance matrix $\boldsymbol{C}_{\boldsymbol{\delta}}$ are unknown random variables, the MMSE estimator for our system model is given by

$$\hat{\boldsymbol{h}}_t = \mathrm{E}[\boldsymbol{h}_t \mid \boldsymbol{Y}] \tag{8}$$

$$= \mathrm{E}[\, \mathrm{E}[\boldsymbol{h}_t \mid \boldsymbol{Y}, \boldsymbol{\delta}] \mid \boldsymbol{Y}] \tag{9}$$

$$= \mathrm{E}[\boldsymbol{W}_{\boldsymbol{\delta}} \boldsymbol{y}_t \mid \boldsymbol{Y}] \tag{10}$$

$$= \mathrm{E}[\boldsymbol{W}_{\boldsymbol{\delta}} \mid \boldsymbol{Y}] \boldsymbol{y}_t \tag{11}$$

$$= \widehat{\boldsymbol{W}}_{\star}(\boldsymbol{Y})\, \boldsymbol{y}_t \tag{12}$$

where we use the law of total expectation and (6) to get the final result. We note that the observations are filtered by the MMSE estimate $\widehat{\boldsymbol{W}}_\star$ of the filter $\boldsymbol{W}_{\boldsymbol{\delta}}$. Hence, the main difficulty of the non-linear channel estimation lies with the calculation of $\widehat{\boldsymbol{W}}_\star$ from the observations $\boldsymbol{Y}$.

Using Bayes' theorem to express the posterior distribution of $\boldsymbol{\delta}$ as

$$p(\boldsymbol{\delta}|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\boldsymbol{\delta})p(\boldsymbol{\delta})}{\int p(\boldsymbol{Y}|\boldsymbol{\delta})p(\boldsymbol{\delta})d\boldsymbol{\delta}} \tag{13}$$

we can write the MMSE filter as

$$\widehat{\boldsymbol{W}}_\star = \int p(\boldsymbol{\delta}|\boldsymbol{Y})\boldsymbol{W}_{\boldsymbol{\delta}} d\boldsymbol{\delta} = \frac{\int p(\boldsymbol{Y}|\boldsymbol{\delta})\boldsymbol{W}_{\boldsymbol{\delta}}\, p(\boldsymbol{\delta})d\boldsymbol{\delta}}{\int p(\boldsymbol{Y}|\boldsymbol{\delta})\, p(\boldsymbol{\delta})d\boldsymbol{\delta}}. \tag{14}$$

The MMSE estimation in (12), (14) can be interpreted as follows. We first calculate $\widehat{\boldsymbol{W}}_\star$ as a convex combination of filters $\boldsymbol{W}_{\boldsymbol{\delta}}$ with weights $p(\boldsymbol{\delta}|\boldsymbol{Y})$ for known covariance matrices $\boldsymbol{C}_{\boldsymbol{\delta}}$ and then apply the resulting filter $\widehat{\boldsymbol{W}}_\star$ to the observation.

By manipulating $p(\boldsymbol{Y}|\boldsymbol{\delta})$, we obtain the following expression for the MMSE filter, which shows that $\widehat{\boldsymbol{W}}_\star$ depends on $\boldsymbol{Y}$ only through the scaled sample covariance matrix

$$\widehat{\boldsymbol{C}} = \frac{1}{\sigma^2}\sum_{t=1}^{T} \boldsymbol{y}_t \boldsymbol{y}_t^{\mathrm{H}}. \tag{15}$$

Lemma 1: If the noise covariance matrix is $\boldsymbol{\Sigma} = \sigma^2\,\mathbf{I}$, the MMSE filter $\widehat{\boldsymbol{W}}_\star$ in (14) can be calculated as

$$\widehat{\boldsymbol{W}}_\star(\widehat{\boldsymbol{C}}) = \frac{\int \exp\left(\mathrm{tr}(\boldsymbol{W}_{\boldsymbol{\delta}}\widehat{\boldsymbol{C}}) + T\log|\,\mathbf{I} - \boldsymbol{W}_{\boldsymbol{\delta}}|\right)\boldsymbol{W}_{\boldsymbol{\delta}}\, p(\boldsymbol{\delta})d\boldsymbol{\delta}}{\int \exp\left(\mathrm{tr}(\boldsymbol{W}_{\boldsymbol{\delta}}\widehat{\boldsymbol{C}}) + T\log|\,\mathbf{I} - \boldsymbol{W}_{\boldsymbol{\delta}}|\right)\, p(\boldsymbol{\delta})d\boldsymbol{\delta}} \tag{16}$$

with $\widehat{\boldsymbol{C}}$ given by (15) and $\boldsymbol{W}_{\boldsymbol{\delta}}$ given by (7).

Proof: See Appendix A. ■

Note that the scaled sample covariance matrix $\widehat{\boldsymbol{C}}$ is a sufficient statistic to calculate the MMSE filter $\widehat{\boldsymbol{W}}_\star$. Moreover, if we define $\widehat{\boldsymbol{H}} = [\hat{\boldsymbol{h}}_1, \ldots, \hat{\boldsymbol{h}}_T]$, we see from $\widehat{\boldsymbol{H}} = \widehat{\boldsymbol{W}}_\star(\widehat{\boldsymbol{C}})\,\boldsymbol{Y}$ that we use all data to construct the sample covariance matrix $\widehat{\boldsymbol{C}}$ and the filter $\boldsymbol{W}_{\boldsymbol{\delta}}$ and then apply the resulting filter to each observation individually to calculate the channel estimate. This structure is beneficial for applications in which we are only interested in the estimate of the most recent channel vector. In such a case, we can apply an adaptive method to track the scaled sample covariance matrix. That is, given the most recent observation $\boldsymbol{y}$, we apply the update

$$\widehat{\boldsymbol{C}} \leftarrow \alpha\widehat{\boldsymbol{C}} + \beta\boldsymbol{y}\boldsymbol{y}^{\mathrm{H}} \tag{17}$$

with suitable $\alpha, \beta > 0$ and then calculate the channel estimate

$$\hat{\boldsymbol{h}} = \widehat{\boldsymbol{W}}_\star(\widehat{\boldsymbol{C}})\,\boldsymbol{y}. \tag{18}$$

IV. MMSE Estimation and Neural Networks

For arbitrary prior distributions $p(\boldsymbol{\delta})$, the MMSE filter as given by Lemma 1 cannot be evaluated in closed form. To make the filter computable, we need the following assumption.

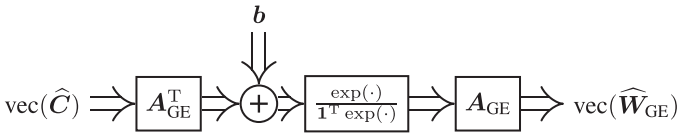

Fig. 1. Block diagram of the gridded estimator $\widehat{\boldsymbol{W}}_{\mathrm{GE}}$.

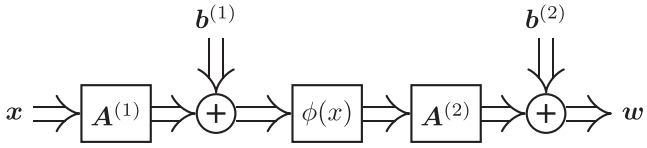

Fig. 2. Neural network with two layers and activation function $\phi(x)$.

Assumption 1: The prior $p(\boldsymbol{\delta})$ is discrete and uniform, i.e., we have a grid $\{\boldsymbol{\delta}_i : i = 1, \ldots, N\}$ of possible values for $\boldsymbol{\delta}$ and

$$p(\boldsymbol{\delta}_i) = \frac{1}{N}\,, \quad \forall i = 1, \ldots, N. \tag{19}$$

Under this assumption, we can evaluate the MMSE estimator of $\boldsymbol{W}_{\boldsymbol{\delta}}$ as

$$\widehat{\boldsymbol{W}}_{\mathrm{GE}}(\widehat{\boldsymbol{C}}) = \frac{\frac{1}{N}\sum_{i=1}^{N} \exp\left(\mathrm{tr}(\boldsymbol{W}_{\boldsymbol{\delta}_i}\widehat{\boldsymbol{C}}) + b_i\right)\boldsymbol{W}_{\boldsymbol{\delta}_i}}{\frac{1}{N}\sum_{i=1}^{N} \exp\left(\mathrm{tr}(\boldsymbol{W}_{\boldsymbol{\delta}_i}\widehat{\boldsymbol{C}}) + b_i\right)} \tag{20}$$

where $\boldsymbol{W}_{\boldsymbol{\delta}_i}$ is obtained by evaluating (7) for $\boldsymbol{\delta} = \boldsymbol{\delta}_i$ and

$$b_i = T\log|\,\mathbf{I} - \boldsymbol{W}_{\boldsymbol{\delta}_i}|. \tag{21}$$

If Assumption 1 does not hold, e.g., if $p(\boldsymbol{\delta})$ describes a continuous distribution, expression (20) is only approximately true if the grid points $\boldsymbol{\delta}_i$ are chosen as random samples from $p(\boldsymbol{\delta})$. In this case, the estimator (20) is a heuristic, suboptimal estimator, which neglects that the true distribution of $\boldsymbol{\delta}$ is continuous. We refer to this estimator as *gridded estimator* (GE). By the law of large numbers, the approximation error vanishes as the number of samples N is increased, but this also increases the complexity of the channel estimation.

We can improve the performance of the estimator for a fixed N by interpreting $\boldsymbol{W}_{\boldsymbol{\delta}_i}$ and $\boldsymbol{b}_i$ as variables that can be optimized instead of using the values in (7) and (21). This is the idea underlying the learning-based approaches, which we describe in the following.

Let us first analyze the structure of the gridded estimator. If we consider the vectorization of $\widehat{\boldsymbol{W}}_{\mathrm{GE}}(\widehat{\boldsymbol{C}})$, i.e.,

$$\mathrm{vec}\left(\widehat{\boldsymbol{W}}_{\mathrm{GE}}(\widehat{\boldsymbol{C}})\right) = \boldsymbol{A}_{\mathrm{GE}}\frac{\exp\left(\boldsymbol{A}_{\mathrm{GE}}^{\mathrm{T}}\mathrm{vec}\left(\widehat{\boldsymbol{C}}\right) + \boldsymbol{b}\right)}{\mathbf{1}^{\mathrm{T}}\exp\left(\boldsymbol{A}_{\mathrm{GE}}^{\mathrm{T}}\mathrm{vec}\left(\widehat{\boldsymbol{C}}\right) + \boldsymbol{b}\right)} \tag{22}$$

where $\boldsymbol{A}_{\mathrm{GE}} = [\mathrm{vec}\left(\boldsymbol{W}_{\boldsymbol{\delta}_1}\right), \ldots, \mathrm{vec}\left(\boldsymbol{W}_{\boldsymbol{\delta}_N}\right)] \in \mathbb{C}^{M^2 \times N}$ and $\boldsymbol{b} = [b_1, \ldots, b_N]$, we see that the function (20) can be visualized as the block diagram shown in Fig. 1. A slightly more general structure is depicted in Fig. 2, which is readily identified as a common structure of a feed-forward neural network (NN) with two *linear layers*, which are connected by a nonlinear *activation function*. The gridded estimator $\widehat{\boldsymbol{W}}_{\mathrm{GE}}$ is a special case

of the neural network in Fig. 2, which uses the *softmax* function

$$\phi(\boldsymbol{x}) = \frac{\exp(\boldsymbol{x})}{\mathbf{1}^{\mathrm{T}} \exp(\boldsymbol{x})} \tag{23}$$

as activation function and the specific choices $\boldsymbol{A}^{(1)} = \boldsymbol{A}_{\mathrm{GE}}^{\mathrm{T}}$, $\boldsymbol{A}^{(2)} = \boldsymbol{A}_{\mathrm{GE}}$, $\boldsymbol{b}^{(1)} = \boldsymbol{b}$ and $\boldsymbol{b}^{(2)} = \mathbf{0}$ for the variables.

To formulate the learning problem mathematically, we define the set of all functions that can be represented by the NN in Fig. 2 as

$$\mathcal{W}_{\mathrm{NN}} = \Big\{ \boldsymbol{f}(\cdot) : \mathbb{C}^{M^2} \mapsto \mathbb{C}^{M^2},\ \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}^{(2)} \phi(\boldsymbol{A}^{(1)} \boldsymbol{x} + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}, \boldsymbol{A}^{(1)} \in \mathbb{C}^{N \times M^2}, \boldsymbol{A}^{(2)} \in \mathbb{C}^{M^2 \times N}, \boldsymbol{b}^{(1)} \in \mathbb{C}^{N}, \boldsymbol{b}^{(2)} \in \mathbb{C}^{M^2} \Big\}. \tag{24}$$

The MSE of a given estimator $\widehat{\boldsymbol{W}}(\cdot)$, which takes the scaled covariance matrix $\widehat{\boldsymbol{C}}$ as input, is given by

$$\varepsilon(\widehat{\boldsymbol{W}}(\cdot)) = \mathrm{E}[\|\boldsymbol{H} - \widehat{\boldsymbol{W}}(\widehat{\boldsymbol{C}})\, \boldsymbol{Y}\|_F^2]. \tag{25}$$

The optimal neural network, i.e., the NN-MMSE estimator, is given as the function in the set $\mathcal{W}_{\mathrm{NN}}$ that minimizes the MSE,

$$\mathrm{vec}\,(\widehat{\boldsymbol{W}}_{\mathrm{NN}}(\cdot)) = \underset{\mathrm{vec}\,(\widehat{\boldsymbol{W}}(\cdot)) \in \mathcal{W}_{\mathrm{NN}}}{\arg\min}\ \varepsilon(\widehat{\boldsymbol{W}}(\cdot)). \tag{26}$$

Since we assume that the dimension N and the activation function $\phi(\cdot)$ are fixed, the variational problem in (26) is simply an optimization over the variables $\boldsymbol{A}^{(\ell)}$ and $\boldsymbol{b}^{(\ell)}$, $\ell = 1, 2$.

If we choose the softmax function as activation function, and if Assumption 1 is fulfilled, we have

$$\varepsilon(\widehat{\boldsymbol{W}}_{\mathrm{GE}}(\cdot)) = \varepsilon(\widehat{\boldsymbol{W}}_{\mathrm{NN}}(\cdot)) = \varepsilon(\widehat{\boldsymbol{W}}_{\star}(\cdot)) \tag{27}$$

since, in this case, the gridded estimator is the MMSE estimator, $\widehat{\boldsymbol{W}}_{\mathrm{GE}}(\cdot) = \widehat{\boldsymbol{W}}_{\star}(\cdot)$, and because $\mathrm{vec}\,(\widehat{\boldsymbol{W}}_{\mathrm{GE}}(\cdot)) \in \mathcal{W}_{\mathrm{NN}}$. In general, we have the relation

$$\varepsilon(\widehat{\boldsymbol{W}}_{\mathrm{GE}}(\cdot)) \geq \varepsilon(\widehat{\boldsymbol{W}}_{\mathrm{NN}}(\cdot)) \geq \varepsilon(\widehat{\boldsymbol{W}}_{\star}(\cdot)). \tag{28}$$

The optimization problem (26) is a typical learning problem for a neural network with a slightly unusual cost function. Due to the expectation in the objective function, we have to revert to stochastic gradient methods to find (local) optima for the variables of the neural network. Unlike the *gridded estimator* (20), which relies on analytic expressions for the covariance matrices $\boldsymbol{C}_{\boldsymbol{\delta}}$, the *neural network estimator* merely needs a large data set $\{(\boldsymbol{H}_1, \boldsymbol{Y}_1), (\boldsymbol{H}_2, \boldsymbol{Y}_2), \ldots\}$ of channel realizations and corresponding observations to optimize the variables. In fact, we could also take samples of channel vectors and observations from a measurement campaign to learn the NN-MMSE estimator for the "true" channel model. This requires that the SNR during the measurement campaign is significantly larger than the SNR in operation. If, as assumed, the noise covariance matrix is known, the observations can then be generated by adding noise to the channel measurements.

The basic structure of the NN-MMSE estimator is depicted in Fig. 3. The learning of the optimal variables $\boldsymbol{A}^{(\ell)}$ and $\boldsymbol{b}^{(\ell)}$ is performed off-line and needs to be done only once. During operation, the channel estimates are obtained by first forming the scaled sample covariance matrix $\widehat{\boldsymbol{C}}$, which is then fed into the neural network $\widehat{\boldsymbol{W}}_{\mathrm{NN}}(\cdot)$. Finally, the output $\widehat{\boldsymbol{W}}_{\mathrm{NN}}(\widehat{\boldsymbol{C}})$ of the neural network is applied as a linear filter to the observations $\boldsymbol{Y}$ to get the channel estimates $\widehat{\boldsymbol{H}}$.

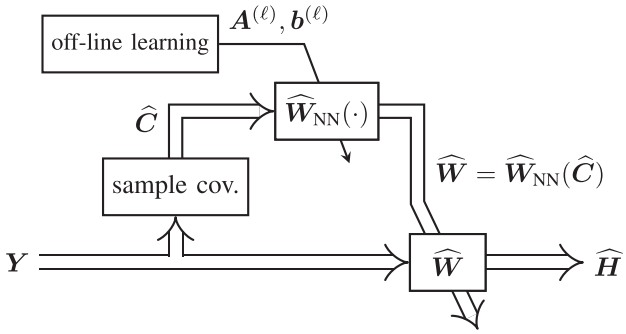

Fig. 3. Channel estimator with embedded neural network.

With proper initialization and sufficient quality of the training data, the neural network estimator is guaranteed to outperform the gridded estimator, which has the same computational complexity. However, there are two problems with this learning approach, which we address in the following sections. First, finding the optimal neural network $\widehat{\boldsymbol{W}}_{\mathrm{NN}}$ is too difficult, because the number of variables is huge and the optimization problem is not convex. Second, even if the optimal variables were known, the computation of the channel estimate $\hat{\boldsymbol{h}}_t = \widehat{\boldsymbol{W}}_{\mathrm{NN}}(\widehat{\boldsymbol{C}})\ \boldsymbol{y}_t$ is too complex: Evaluating the output of the neural network $\widehat{\boldsymbol{W}}_{\mathrm{NN}}(\widehat{\boldsymbol{C}})$ needs $\mathcal{O}(M^2 N)$ floating point operations due to the matrix-vector products (cf. Fig. 2). For example, if the grid size N needs to scale linearly with the number of antennas M to obtain accurate estimates, the computational complexity scales as $\mathcal{O}(M^3)$, which is too high for practical applications.

V. Low-Complexity MMSE Estimation

With Assumption 1 the gridded estimator $\widehat{\boldsymbol{W}}_{\mathrm{GE}}$ in (20) is the MMSE estimator. In the following, we introduce additional assumptions, which help to simplify $\widehat{\boldsymbol{W}}_{\mathrm{GE}}$. With these assumptions, we get a fast channel estimator, i.e., one with a computational complexity of only $\mathcal{O}(M \log M)$. Just as for the gridded estimator, the fast estimator is no longer the MMSE estimator if the assumptions are violated. However, in analogy to Section IV, the structure of this fast estimator motivates the convolutional neural network (CNN) estimator presented in Section VI.

Our approach to reduce the complexity of $\widehat{\boldsymbol{W}}_{\mathrm{GE}}$ can be broken down into two steps. First, we exploit common structure of the covariance matrices, which occur for commonly used array geometries. In a second step, we use an approximated shift-invariance structure, which is present in a certain channel model with only a single path of propagation. With those two steps, we reduce the computational complexity from $\mathcal{O}(M^2 N)$ to $\mathcal{O}(M \log M)$.

A. A Structured MMSE Estimator

In the first step, we replace the filters $\boldsymbol{W}_{\boldsymbol{\delta}_i}$ in (20) with structured matrices that use only $\mathcal{O}(M)$ variables. Specifically, we make the following assumption.

Assumption 2: The filters $\boldsymbol{W}_{\boldsymbol{\delta}_i}$ can be decomposed as

$$\boldsymbol{W}_{\boldsymbol{\delta}_i} = \boldsymbol{Q}^{\mathrm{H}} \operatorname{diag}(\boldsymbol{w}_i)\boldsymbol{Q} \tag{29}$$

with a common matrix $\boldsymbol{Q} \in \mathbb{C}^{K\times M}$ and vectors $\boldsymbol{w}_i \in \mathbb{R}^K$ where $\mathcal{O}(K) = \mathcal{O}(M)$.

Note that the requirement $\mathcal{O}(K) = \mathcal{O}(M)$ ensures the desired dimensionality reduction and $\boldsymbol{w}_i \in \mathbb{R}^K$ leads to self-adjoint filters. Combining Assumptions 1 and 2, we get the following result.

Theorem 1: Given Assumptions 1 and 2, the MMSE estimator of $\boldsymbol{W}_{\boldsymbol{\delta}}$ simplifies to

$$\widehat{\boldsymbol{W}}_{\mathrm{SE}}(\widehat{\boldsymbol{C}}) = \boldsymbol{Q}^{\mathrm{H}} \operatorname{diag}(\hat{\boldsymbol{w}}_{\mathrm{SE}}(\hat{\boldsymbol{c}}))\boldsymbol{Q} \tag{30}$$

where

$$\hat{\boldsymbol{c}} = \frac{1}{\sigma^2}\sum_{t=1}^{T} |\boldsymbol{Q}\boldsymbol{y}_t|^2 . \tag{31}$$

Moreover, the element-wise filter $\hat{\boldsymbol{w}}_{\mathrm{SE}}$ is given by

$$\hat{\boldsymbol{w}}_{\mathrm{SE}}(\hat{\boldsymbol{c}}) = \boldsymbol{A}_{\mathrm{SE}} \frac{\exp(\boldsymbol{A}_{\mathrm{SE}}^{\mathrm{T}}\hat{\boldsymbol{c}} + \boldsymbol{b})}{\mathbf{1}^{\mathrm{T}} \exp(\boldsymbol{A}_{\mathrm{SE}}^{\mathrm{T}}\hat{\boldsymbol{c}} + \boldsymbol{b})} \tag{32}$$

where the matrix

$$\boldsymbol{A}_{\mathrm{SE}} = [\boldsymbol{w}_i, \ldots, \boldsymbol{w}_N] \in \mathbb{R}^{K\times N} \tag{33}$$

contains the element-wise MMSE filters (29), and the entries of the vector

$$\boldsymbol{b} = [b_1, \ldots, b_N]^{\mathrm{T}} \in \mathbb{R}^N \tag{34}$$

are given by (21).

Proof: If we replace the filters $\boldsymbol{W}_{\boldsymbol{\delta}_i}$ in (20) with the parametrization in (29), we can simplify the trace expressions according to

$$\operatorname{tr}(\boldsymbol{W}_{\boldsymbol{\delta}_i}\widehat{\boldsymbol{C}}) = \operatorname{tr}(\boldsymbol{Q}^{\mathrm{H}} \operatorname{diag}(\boldsymbol{w}_i)\boldsymbol{Q}\widehat{\boldsymbol{C}}) \tag{35}$$

$$= \operatorname{tr}\left(\operatorname{diag}(\boldsymbol{w}_i)\frac{1}{\sigma^2}\sum_{t=1}^{T} \boldsymbol{Q}\boldsymbol{y}_t\boldsymbol{y}_t^{\mathrm{H}}\boldsymbol{Q}^{\mathrm{H}} \right) \tag{36}$$

$$= \boldsymbol{w}_i^{\mathrm{T}}\hat{\boldsymbol{c}} \tag{37}$$

as $\hat{\boldsymbol{c}}$ contains the diagonal elements of the matrix

$$\frac{1}{\sigma^2}\sum_{t=1}^{T} \boldsymbol{Q}\boldsymbol{y}_t\boldsymbol{y}_t^{\mathrm{H}}\boldsymbol{Q}^{\mathrm{H}}. \tag{38}$$

Consequently, the gridded estimator in (20) simplifies to

$$\widehat{\boldsymbol{W}}_{\mathrm{SE}}(\hat{\boldsymbol{c}}) = \frac{\sum_{i=1}^{N} \exp(\boldsymbol{w}_i^{\mathrm{T}}\hat{\boldsymbol{c}} + b_i)\boldsymbol{Q}^{\mathrm{H}} \operatorname{diag}(\boldsymbol{w}_i)\boldsymbol{Q}}{\sum_{i=1}^{N} \exp(\boldsymbol{w}_i^{\mathrm{T}}\hat{\boldsymbol{c}} + b_i)}$$

$$= \boldsymbol{Q}^{\mathrm{H}} \operatorname{diag}\left(\frac{\sum_{i=1}^{N} \exp(\boldsymbol{w}_i^{\mathrm{T}}\hat{\boldsymbol{c}} + b_i)\boldsymbol{w}_i}{\sum_{i=1}^{N} \exp(\boldsymbol{w}_i^{\mathrm{T}}\hat{\boldsymbol{c}} + b_i)} \right) \boldsymbol{Q}. \tag{39}$$

With the definitions of $\boldsymbol{A}_{\mathrm{SE}}$ and $\boldsymbol{b}$, we can write (39) as (30). ■

If Assumptions 1 and 2 hold, the MMSE estimates of the channel vectors using the *structured estimator* (SE) can be calculated as

$$\hat{\boldsymbol{h}}_t = \boldsymbol{Q}^{\mathrm{H}} \operatorname{diag}(\hat{\boldsymbol{w}}_{\mathrm{SE}}(\hat{\boldsymbol{c}}))\boldsymbol{Q}\boldsymbol{y}_t \tag{40}$$

i.e., $\widehat{\boldsymbol{W}}_{\star}(\widehat{\boldsymbol{C}}) = \boldsymbol{Q}^{\mathrm{H}} \operatorname{diag}(\hat{\boldsymbol{w}}_{\mathrm{SE}}(\hat{\boldsymbol{c}})\boldsymbol{Q}$.

Given $\hat{\boldsymbol{w}}_{\mathrm{SE}}(\hat{\boldsymbol{c}})$, the complexity of the estimator depends only on the number of operations required to calculate matrix-vector products with $\boldsymbol{Q}$ and $\boldsymbol{Q}^{\mathrm{H}}$. To achieve the desired complexity $\mathcal{O}(M \log M)$, the matrix $\boldsymbol{Q}$ must have some special structure that enables fast computations. If this is the case, the complexity of the *structured estimator* is dominated by the calculation of $\hat{\boldsymbol{w}}_{\mathrm{SE}}(\hat{\boldsymbol{c}})$, which is $\mathcal{O}(NK)$. In Section V-B, we show how the complexity of the calculation of $\hat{\boldsymbol{w}}_{\mathrm{SE}}(\hat{\boldsymbol{c}})$ can be reduced further.

Examples: For a uniform linear array (ULA), the channel covariance matrices, which have Toeplitz structure, are asymptotically equivalent to corresponding circulant matrices (cf. [16], Appendix B). Since all circulant matrices have the columns of the discrete Fourier transform (DFT) matrix $\boldsymbol{F}$ as eigenvectors, we have the asymptotic equivalence

$$\boldsymbol{C}_{\boldsymbol{\delta}} \asymp \boldsymbol{F}^{\mathrm{H}} \operatorname{diag}(\boldsymbol{c}_{\boldsymbol{\delta}})\boldsymbol{F} \ \ \forall \boldsymbol{\delta} \tag{41}$$

where $\boldsymbol{c}_{\boldsymbol{\delta}}$ contains the diagonal elements of $\boldsymbol{F}\boldsymbol{C}_{\boldsymbol{\delta}}\boldsymbol{F}^{\mathrm{H}}$. As a consequence, we have a corresponding asymptotic equivalence

$$\boldsymbol{W}_{\boldsymbol{\delta}} \asymp \boldsymbol{F}^{\mathrm{H}} \operatorname{diag}(\boldsymbol{w}_{\boldsymbol{\delta}})\boldsymbol{F} \ \ \forall \boldsymbol{\delta} \tag{42}$$

where $\boldsymbol{w}_{\boldsymbol{\delta}}$ contains the diagonal elements of $\boldsymbol{F}\boldsymbol{W}_{\boldsymbol{\delta}}\boldsymbol{F}^{\mathrm{H}}$. For a large-scale system, this is a very good approximation [17]. We call the structured estimator that uses Assumption 2 with $\boldsymbol{Q} = \boldsymbol{F}$ the *circulant estimator*.

To reduce the approximation error for finite numbers of antennas, we can use a more general factorization with $\boldsymbol{Q} = \boldsymbol{F}_2$, where $\boldsymbol{F}_2 \in \mathbb{C}^{2M\times M}$ contains the first M columns of a $2M \times 2M$ DFT matrix. The class of matrices that can be expressed as

$$\boldsymbol{W}_{\boldsymbol{\delta}} = \boldsymbol{F}_2^{\mathrm{H}} \operatorname{diag}(\boldsymbol{w}_{\boldsymbol{\delta}})\boldsymbol{F}_2 \tag{43}$$

are exactly the Toeplitz matrices [17]. Note that the filters $\boldsymbol{W}_{\boldsymbol{\delta}}$ do not actually have Toeplitz structure, even if the channel covariance matrices are Toeplitz matrices. The Toeplitz assumption only holds in the limit for large numbers of antennas due to the arguments given above or for low SNR when the noise covariance matrix dominates the inverse in (7). Nevertheless, the Toeplitz structure is more general than the circulant structure and, thus, yields a smaller approximation error. The estimator that uses Assumption 2 with $\boldsymbol{Q} = \boldsymbol{F}_2$ is denoted as the *Toeplitz estimator*.

An analogous result can be derived for uniform rectangular arrays (cf. Appendix C). In this case, the transformation $\boldsymbol{Q}$ is the Kronecker product of two DFT matrices, whose dimensions correspond to the number of antennas in both directions of the array.

A third example with a decomposition as in Assumption 2 is a setup with distributed antennas [18], [19]. For distributed antennas, the covariance matrices are typically modelled as diagonal matrices and, thus, the filters $\boldsymbol{W}_{\boldsymbol{\delta}}$ are diagonal as well. That is, for distributed antennas we simply have $\boldsymbol{Q} = \mathbf{I}$.

B. A Fast MMSE Estimator

The main complexity in the evaluation of $\hat{\boldsymbol{w}}_{\text{SE}}(\cdot)$ stems from the matrix-vector products in (32). The complexity can be reduced by using only matrices $\boldsymbol{A}_{\text{SE}}$ that allow for fast matrix-vector products. One possible choice are the circulant matrices.

In fact, circulant matrices naturally arise in the structured estimator for a single-path channel model with a single parameter δ for the angle of arrival. In this model, the power spectrum is shift-invariant, i.e., $g(\theta;\delta) = g(\theta-\delta)$. As a result, for $N = K$, the samples $\boldsymbol{w}_i$ of the structured estimator $\hat{\boldsymbol{w}}_{\text{SE}}$ in (32) are approximately shift invariant, i.e., their entries satisfy $[\boldsymbol{w}_i]_j = [\boldsymbol{w}_{i+n}]_{j+n}$ (the sums are modulo M) and the following assumption is satisfied (more details are given in Appendix D).

Assumption 3: The matrix $\boldsymbol{A}_{\text{SE}}$ in (33) is circulant and given by

$$\boldsymbol{A}_{\text{SE}} = \boldsymbol{F}^{\text{H}} \operatorname{diag}(\boldsymbol{F}\boldsymbol{w}_0)\boldsymbol{F} \tag{44}$$

for some $\boldsymbol{w}_0 \in \mathbb{R}^K$, where $\boldsymbol{F}$ is the K-dimensional DFT matrix.

Note that Assumption 3 is, in principle, independent of Assumption 2. We see from the examples that the structure of $\boldsymbol{W}_{\boldsymbol{\delta}}$ and, thus, the choice for $\boldsymbol{Q}$ is motivated by the array geometry, while the assumption that $\boldsymbol{A}_{\text{SE}}$ is circulant is motivated by the physical channel model. The example in Appendix D, which is based on the ULA geometry and the 3GPP channel model, suggests a circulant structure for both the filters $\boldsymbol{W}_{\boldsymbol{\delta}}$ and the matrix $\boldsymbol{A}_{\text{SE}}$ in $\hat{\boldsymbol{w}}_{\text{SE}}(\cdot)$.

However, we could think of other system setups, where the structure of $\boldsymbol{W}_{\boldsymbol{\delta}}$ in Assumption 2 is different than the structure of $\boldsymbol{A}_{\text{SE}}$ in Assumption 3. As an illustration, consider a toy example where we have an array of antennas along a long corridor, say in an airplane. Then we could reasonably assume diagonal covariance matrices, i.e., $\boldsymbol{Q} = \mathbf{I}$, but at the same time we have a shift-invariance for different positions of the users in the corridor, i.e., Assumption 3 also holds.

Given the relationship between circulant matrices and circular convolution, we can write

$$\boldsymbol{A}_{\text{SE}}\boldsymbol{x} = \boldsymbol{F}^{\text{H}} \operatorname{diag}(\boldsymbol{F}\boldsymbol{a})\boldsymbol{F}\boldsymbol{x} = \boldsymbol{a} * \boldsymbol{x} \tag{45}$$

with $\boldsymbol{a} \in \mathbb{R}^K$. Because of the FFT, the computational complexity of evaluating $\hat{\boldsymbol{w}}_{\text{SE}}(\hat{\boldsymbol{c}})$ reduces to $\mathcal{O}(M \log M)$ if $\mathcal{O}(K) = \mathcal{O}(M)$. That is, we get a *fast estimator* (FE)

$$\hat{\boldsymbol{w}}_{\text{FE}}(\hat{\boldsymbol{c}}) = \boldsymbol{w}_0 * \operatorname{softmax}(\tilde{\boldsymbol{w}}_0 * \hat{\boldsymbol{c}} + \boldsymbol{b}) \tag{46}$$

by incorporating the constraint (44) into $\hat{\boldsymbol{w}}_{\text{SE}}$. The vector $\tilde{\boldsymbol{w}}_0$ contains the entries of $\boldsymbol{w}_0$ in reversed order.

VI. Low-Complexity Neural Network

For most channel models, Assumptions 1, 2, and 3 only hold approximately or even not at all. That is, the estimator $\hat{\boldsymbol{w}}_{\text{FE}}$ in (46) does not yield the MMSE estimator in most practical scenarios. Nevertheless, it is still worthwhile to consider an estimator with similar structure: Calculating a channel estimate with $\widehat{\boldsymbol{W}}_{\text{GE}}$ costs $\mathcal{O}(M^2N)$ FLOPS, while using $\hat{\boldsymbol{w}}_{\text{FE}}$ only requires $\mathcal{O}(M \log M)$ operations. As we discuss in the following, another advantage is that the number of variables that have to be learned reduces from $\mathcal{O}(M^2N)$ to $\mathcal{O}(M)$, since we no longer have full matrices, but circular convolutions.

Algorithm 1: Learned fast MMSE filter.

1: Initialize variables $\boldsymbol{a}^{(\ell)}$ and $\boldsymbol{b}^{(\ell)}$ randomly
2: Generate/select a mini-batch of S channel vectors $\boldsymbol{H}_s$ and corresponding observations $\boldsymbol{Y}_s$ (and $\hat{\boldsymbol{c}}_s$) for $s = 1$, ..., S
3: Calculate the stochastic gradient

$$\boldsymbol{g} = \frac{1}{S}\sum_{s=1}^{S} \frac{\partial}{\partial[\boldsymbol{a}^{(\ell)};\boldsymbol{b}^{(\ell)}]} \left\| \boldsymbol{H}_s - \boldsymbol{Q}^{\text{H}} \operatorname{diag}(\hat{\boldsymbol{w}}(\hat{\boldsymbol{c}}_s))\boldsymbol{Q}\boldsymbol{Y}_s \right\|_F^2$$

with $\hat{\boldsymbol{w}}(\boldsymbol{x})$ as stated in (46)
4: Update variables with a gradient algorithm (e.g., [20])
5: Repeat steps 1–3 until a convergence criterion is satisfied

In Section IV we discussed how learning can be used to compensate for the approximation error that results from a finite grid size N, i.e., a violation of Assumption 1. Analogously, we can learn the variables of a convolutional neural network inspired by $\hat{\boldsymbol{w}}_{\text{FE}}$ to compensate for violations of Assumptions 2 and 3. To this end, we define the set of CNNs

$$\mathcal{W}_{\text{CNN}} = \Big\{ \boldsymbol{x} \mapsto \boldsymbol{a}^{(2)} * \phi\Big(\boldsymbol{a}^{(1)} * \boldsymbol{x} + \boldsymbol{b}^{(1)}\Big) + \boldsymbol{b}^{(2)}, \; \boldsymbol{a}^{(\ell)} \in \mathbb{R}^K, \boldsymbol{b}^{(\ell)} \in \mathbb{R}^K, \ell = 1,2 \Big\} \tag{47}$$

and the optimal *CNN estimator* is the one using

$$\hat{\boldsymbol{w}}_{\text{CNN}}(\cdot) = \underset{\hat{\boldsymbol{w}}(\cdot)\in\mathcal{W}_{\text{CNN}}}{\arg\min}\; \varepsilon(\boldsymbol{Q}^{\text{H}}\hat{\boldsymbol{w}}(\cdot)\boldsymbol{Q}). \tag{48}$$

Again, we assume that the activation function $\phi(\cdot)$ is fixed. Thus, the optimization is only with respect to the convolution kernels $\boldsymbol{a}^{(\ell)}$ and the bias vectors $\boldsymbol{b}^{(\ell)}$.

Analogously to Section IV, if we choose the softmax function as activation function, we have $\hat{\boldsymbol{w}}_{\text{FE}}(\cdot) \in \mathcal{W}_{\text{CNN}}$. Consequently, if Assumptions 1–3 are fulfilled we get

$$\varepsilon(\boldsymbol{Q}^{\text{H}}\hat{\boldsymbol{w}}_{\text{FE}}(\cdot)\boldsymbol{Q}) = \varepsilon(\boldsymbol{Q}^{\text{H}}\hat{\boldsymbol{w}}_{\text{CNN}}(\cdot)\boldsymbol{Q}) = \varepsilon(\widehat{\boldsymbol{W}}_{\star}(\cdot)). \tag{49}$$

In general, we have

$$\varepsilon(\boldsymbol{Q}^{\text{H}}\hat{\boldsymbol{w}}_{\text{FE}}(\cdot)\boldsymbol{Q}) \geq \varepsilon(\boldsymbol{Q}^{\text{H}}\hat{\boldsymbol{w}}_{\text{CNN}}(\cdot)\boldsymbol{Q}) \geq \varepsilon(\widehat{\boldsymbol{W}}_{\star}(\cdot)). \tag{50}$$

The stochastic-gradient method that learns the CNN is described in detail in Algorithm 1. We want to stress again that the learning procedure is performed off-line and does not add to the complexity of the channel estimation. During operation, the channel estimation is performed by evaluating $\hat{\boldsymbol{w}}_{\text{CNN}}(\hat{\boldsymbol{c}})$ and the transformations involving the $\boldsymbol{Q}$ matrix for given observations. If the variables are learned from simulated samples according to the 3GPP or any other channel model, this algorithm suffers from the same model-reality mismatch as does any other model-based algorithm. The fact that the proposed algorithm can also be trained on true channel realizations puts it into a significant advantage over other non-learning-based algorithms, which have to rely on models only.

Algorithm 2: Hierarchical Training.

1: Choose upsampling factor $\beta > 1$ and number of stages n
2: Set $M_0 = \lceil M/\beta^n \rceil$, $K_0 = \lceil K/\beta^n \rceil$
3: Learn optimal $\boldsymbol{a}_0^{(\ell)}, \boldsymbol{b}_0^{(\ell)} \in \mathbb{R}^{K_0}$ using Algorithm 1 assuming M_0 antennas with random initializations
4: **for** i from 1 to n **do**
5: Set $M_i = \lceil M/\beta^{n-i} \rceil$ and $K_i = \lceil K/\beta^{n-i} \rceil$
6: Interpolate $\boldsymbol{a}_i^{(\ell)}, \boldsymbol{b}_i^{(\ell)} \in \mathbb{R}^{K_i}$ from $\boldsymbol{a}_{i-1}^{(\ell)}, \boldsymbol{b}_{i-1}^{(\ell)} \in \mathbb{R}^{K_{i-1}}$
7: Normalize $\boldsymbol{a}_i^{(\ell)}$ by dividing by β
8: Learn optimal $\boldsymbol{a}_i^{(\ell)}, \boldsymbol{b}_i^{(\ell)}$ using Algorithm 1 assuming M_i antennas and using $\boldsymbol{a}_i^{(\ell)}, \boldsymbol{b}_i^{(\ell)}$ as initializations
9: **end for**

In the simulations, we compare two variants of the CNN estimator. First, we use the softmax activation function $\phi = \frac{\exp(\cdot)}{\mathbf{1}^{\mathrm{T}} \exp(\cdot)}$. The resulting softmax CNN estimator is a direct improvement over the fast estimator with $\hat{\boldsymbol{w}}_{\mathrm{FE}}$, which was derived under Assumptions 1–3. In the second variant, we use a rectified linear unit (ReLU) $\phi(x) = [x]_+$ as activation function, since ReLUs were found to be easier to train than other activation functions [21] (and they are also easier to evaluate than the softmax function).

A. Hierarchical Learning

Local optima are a major issue when learning the neural networks, i.e., when calculating a solution of the nonlinear optimization problem (48). During our experiments, we observed that, especially for a large number of antennas, the learning often gets stuck in local optima. To deal with this problem, we devise a hierarchical learning procedure that starts the learning with a small number of antennas and then increases the number of antennas step by step.

For the single-path channel model, which motivates the circulant structure of the matrices $\boldsymbol{A}^{(\ell)}$, the convolution kernel $\boldsymbol{w}_0$ contains samples of the continuous function $w(u;0)$, i.e., $[\boldsymbol{w}_0]_k = w(2\pi(k-1)/K;0)$ (cf. Appendix D). If we assume that $w(u;0)$ is a smooth function, we can quite accurately calculate the generating vector $\boldsymbol{w}_0$ for a system with M antennas from the corresponding vector of a system with less antennas by commonly used interpolation methods.

This observation inspires the following heuristic for initializing the variables $\boldsymbol{a}^{(\ell)}$ and $\boldsymbol{b}^{(\ell)}$ of a K-dimensional CNN. We first learn the variables of a smaller CNN, e.g., we choose a CNN with dimension $K/2$. We use the resulting variables to initialize every second entry of the vectors $\boldsymbol{a}^{(\ell)}$ and $\boldsymbol{b}^{(\ell)}$. The remaining entries can be obtained by numerical interpolation.

For the filter $\hat{\boldsymbol{w}}_{\mathrm{CNN}}(\cdot)$, it is desirable to have outputs of similar magnitude, irrespective of the dimension K. By doubling the number of entries of the convolution kernels via interpolation, we approximately double the largest absolute value of $\boldsymbol{a}^{(\ell)} * \boldsymbol{x}$. To remedy this issue, we normalize the kernels of the convolution after the interpolation such that we get approximately similar values at the outputs of each layer. This heuristic leads to the hierarchical learning described in Algorithm 2.

The hierarchical learning significantly improves convergence speed and reduces the computational complexity per iteration due to the reduced number of antennas in many learning steps. In fact, for a large number of antennas the hierarchical learning is essential to obtain good performance. In Fig. 4, we show a standard box plot [22] of the MSE of the estimators obtained by applying the hierarchical and the standard learning procedure. Each data point used to generate the box plot corresponds to a randomly initialized estimator and one run of Algorithm 2 with $\beta = 2$ and $n = 3$ for the hierarchical learning and $n = 0$ for the non-hierarchical learning (and the same total number of iterations). The box plot depicts a summary of the resulting distribution, showing the median and the quartiles in a box and outliers outside of the whiskers as additional dots. The whiskers are at the position of the lowest and highest data point within a distance from the box of 1.5 times the box size. As we can see, without the hierarchical learning, the learning procedure gets stuck in local optima. With the hierarchical approach, we are less likely to be caught in local optima during the learning process.

VII. Related Work

In this section, we give a short summary of two alternative channel estimation methods with $\mathcal{O}(M \log M)$ complexity. These methods will serve as a benchmark in the numerical evaluation of our novel algorithms.

A. ML Covariance Matrix Estimation

The common approach to approximate MMSE channel estimation for unknown covariance matrices is to use a maximum likelihood (ML) estimate of the channel covariance matrix. That is, we first find an ML estimate of the channel covariance matrix $\boldsymbol{C}_{\boldsymbol{\delta}}^{\mathrm{ML}}$ based on the observations $\boldsymbol{Y}$ and then, assuming the estimate is exact, calculate the MMSE estimates of the channel vectors as in (6), (7). The disadvantage of ML estimation is that a general prior $p(\boldsymbol{\delta})$ cannot be incorporated.

The likelihood function for the channel covariance matrix given the noise covariance matrix is

$$L(\boldsymbol{C}_{\boldsymbol{\delta}}|\boldsymbol{Y}) = \exp\left(-\sum_{t=1}^{T} \boldsymbol{y}_t^{\mathrm{H}} (\boldsymbol{C}_{\boldsymbol{\delta}} + \boldsymbol{\Sigma})^{-1} \boldsymbol{y}_t - T \log |\boldsymbol{C}_{\boldsymbol{\delta}} + \boldsymbol{\Sigma}|\right) \frac{1}{\pi^{MT}} \tag{51}$$

and the ML problem reads as

$$\boldsymbol{C}_{\boldsymbol{\delta}}^{\mathrm{ML}} = \underset{\boldsymbol{C}_{\boldsymbol{\delta}} \in \mathcal{M}}{\arg\max}\, L(\boldsymbol{C}_{\boldsymbol{\delta}}|\boldsymbol{Y}) \tag{52}$$

where $\mathcal{M}$ is the set of admissible covariance matrices, which has to be included in the set of positive semi-definite matrices $\mathcal{S}_0^+$, i.e., $\mathcal{M} \subset \mathcal{S}_0^+$.

If $\mathcal{M} = \mathcal{S}_0^+$, the ML estimate is given in terms of the sample covariance matrix

$$\boldsymbol{S} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{y}_t \boldsymbol{y}_t^{\mathrm{H}} \tag{53}$$

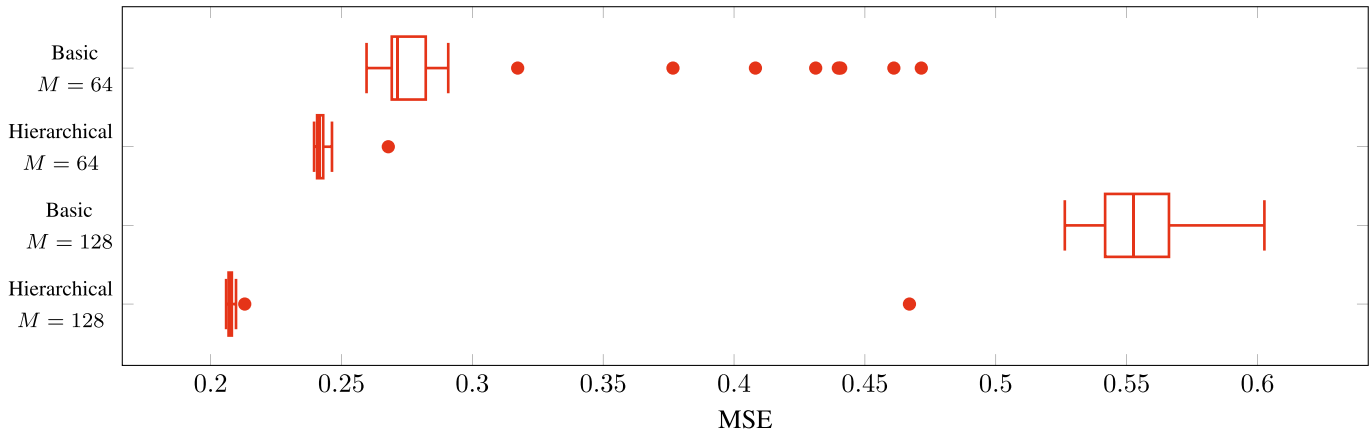

Fig. 4. Box plot with outliers (marked as dots) of the MSE after learning for 10 000 iterations for hierarchical and non-hierarchical learning. We show results for $M = 64$ and $M = 128$ antennas for 50 data points per plot and with the DFT matrix $\boldsymbol{Q} = \boldsymbol{F}$ for the transformation. Scenario with three propagation paths, $\sigma^2 = 1, T = 1$.

as

$$\boldsymbol{C}_{\boldsymbol{\delta}}^{\mathrm{ML}} = \boldsymbol{\Sigma}^{1/2} P_{\mathcal{S}_0^+} \left(\boldsymbol{\Sigma}^{-1/2} \boldsymbol{S} \boldsymbol{\Sigma}^{-1/2} - \mathbf{I} \right) \boldsymbol{\Sigma}^{1/2} \tag{54}$$

where we use the projection $P_{\mathcal{S}_0^+}(\cdot)$ onto the cone of positive semi-definite matrices [23]. The projection $P_{\mathcal{S}_0^+}(\boldsymbol{X})$ of a hermitian matrix $\boldsymbol{X}$ replaces all negative eigenvalues of $\boldsymbol{X}$ with zeros. For $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$, the estimate simplifies to

$$\boldsymbol{C}_{\boldsymbol{\delta}}^{\mathrm{ML}} = P_{\mathcal{S}_0^+} \left(\widehat{\boldsymbol{C}} - \sigma^2 \mathbf{I} \right). \tag{55}$$

Low-Complexity ML Estimation: If we have a ULA at the base station, we know that the covariance matrix has to be a Toeplitz matrix. Thus, we should choose $\mathcal{M} = \mathcal{T}_0^+$ as the set of positive semi-definite Toeplitz matrices. In this case, the ML estimate can no longer be given in closed form and iterative methods have to be used [10], [24], [25].

Since we are interested in low-complexity estimators, we approximate the solution by reducing the constraint set to positive semi-definite, circulant matrices $\mathcal{M} = \mathcal{C}^+$. This choice reduces the complexity of the ML estimator significantly [23]. The reason is that all circulant matrices have the columns of the DFT matrix $\boldsymbol{F}$ as eigenvectors. That is, we can parametrize the ML estimate as

$$\boldsymbol{C}_{\boldsymbol{\delta}}^{\mathrm{ML}} = \boldsymbol{F}^{\mathrm{H}} \operatorname{diag}(\boldsymbol{c}_{\boldsymbol{\delta}}^{\mathrm{ML}}) \boldsymbol{F} \tag{56}$$

where $\boldsymbol{c}_{\boldsymbol{\delta}}^{\mathrm{ML}} \in \mathbb{R}^M$ contains the M eigenvalues of $\boldsymbol{C}_{\boldsymbol{\delta}}^{\mathrm{ML}}$.

Incorporating (56) into the likelihood function (51), we notice that the estimate of the channel covariance matrix can be given in terms of the estimated power spectrum [23], [26]

$$\boldsymbol{s} = \frac{1}{T} \sum_{t=1}^{T} |\boldsymbol{F} \boldsymbol{y}_t|^2 \tag{57}$$

where $|\boldsymbol{x}|^2$ is the vector of absolute squared entries of $\boldsymbol{x}$. Specifically, we have the estimated eigenvalues

$$\boldsymbol{c}_{\boldsymbol{\delta}}^{\mathrm{ML}} = [\boldsymbol{s} - \sigma^2 \mathbf{1}]_+ \tag{58}$$

where the ith element of $[\boldsymbol{x}]_+$ is $\max([\boldsymbol{x}]_i, 0)$ and where $\mathbf{1}$ is the all-ones vector. The approximate MMSE estimate of the channel vector in coherence interval t is given by

$$\hat{\boldsymbol{h}}_t = \boldsymbol{F}^{\mathrm{H}} \operatorname{diag}(\boldsymbol{c}_{\boldsymbol{\delta}}^{\mathrm{ML}}) \operatorname{diag}(\boldsymbol{c}_{\boldsymbol{\delta}}^{\mathrm{ML}} + \sigma^2 \mathbf{1})^{-1} \boldsymbol{F} \boldsymbol{y}_t \tag{59}$$

and can be calculated with a complexity of $\mathcal{O}(M \log M)$ due to the FFT. The almost linear complexity makes the ML approach with the circulant approximation suitable for large-scale wireless systems.

B. Compressive Sensing Based Estimation

The ML-based channel estimation techniques exploit the Toeplitz structure of the covariance matrix, which is a result of regular array geometries and the model (5) with a continuous power density function g. In the 3GPP models, this power density function usually has a very limited angular support, i.e., $g(\theta, \boldsymbol{\delta})$ is approximately zero except for θ in the vicinity of the cluster centers $\boldsymbol{\delta}$. The resulting covariance matrices have a very low numerical rank [27]. As a consequence, under such a model, any given realization of a channel vector admits a sparse approximation

$$\boldsymbol{h} \approx \boldsymbol{D} \boldsymbol{x} \tag{60}$$

in a given dictionary $\boldsymbol{D} \in \mathbb{C}^{M \times Q}$, where all but k entries of $\boldsymbol{x}$ are zero. The vector $\boldsymbol{x}$ can be found by solving the sparse approximation problem

$$\boldsymbol{x} = \underset{\boldsymbol{x} \in \mathbb{C}^Q : |\operatorname{supp}(\boldsymbol{x})| \leq k}{\arg\min} \|\boldsymbol{y} - \boldsymbol{D} \boldsymbol{x}\|^2 \tag{61}$$

where $|\operatorname{supp}(\boldsymbol{x})|$ denotes the number of nonzero entries of $\boldsymbol{x}$. This combinatorial optimization problem can be solved efficiently with methods from the area of compressive sensing, e.g., the orthogonal matching pursuit (OMP) algorithm [28] or iterative hard thresholding (IHT) [29].

It is common to use a dictionary $\boldsymbol{D}$ of steering vectors $\boldsymbol{a}(\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ varies between $-\pi$ and π on a grid [4]. For ULAs, this grid can be chosen such that the dictionary $\boldsymbol{D}$ results in an oversampled DFT matrix, which has the advantage that matrix-vector products with this matrix can be computed efficiently. Furthermore, it was shown in [27] that this dictionary is a

reasonable choice, at least for the single-cluster 3GPP model, and if the OMP algorithm is used to find the sparse approximation.

The OMP algorithm can be extended to a multiple measurement model

$$\boldsymbol{H} \approx \boldsymbol{D}\boldsymbol{X} \tag{62}$$

with a row-sparse matrix $\boldsymbol{X}$, i.e., each channel realization is approximated as a linear combination of the same k dictionary vectors. Because the selection of the optimal sparsity level k is non-trivial, we use a genie-aided approach in our simulations. The genie-aided OMP algorithm uses the actual channel realizations $\boldsymbol{H}$ to decide about the optimal value for k that maximizes the metric of interest. The result is clearly an upper bound for the performance of the OMP algorithm.

VIII. Simulations

For the numerical evaluation of the newly introduced algorithms, we focus on the far-field model with a ULA at the base station (cf. Appendix B). We assume that the noise variance σ^2 and the correct model for the parameters, i.e., the prior $p(\boldsymbol{\delta})$ and the mapping from $\boldsymbol{\delta}$ to $\boldsymbol{C}_{\boldsymbol{\delta}}$, are known. That is, for the off-line learning procedure required by the CNN estimators, we can use the true prior to generate the necessary realizations of channel vectors and observations.

We first consider the single-path model that motivates Assumption 3 (cf. App. D). Even for this idealized model, Assumptions 1–3 only hold approximately. To compare the simple gridded estimator (GE) $\widehat{\boldsymbol{W}}_{\text{GE}}$ (Assumption 1) with the structured estimator (SE) $\hat{\boldsymbol{w}}_{\text{SE}}$ (Assumptions 1 and 2) and the fast estimator (FE) $\hat{\boldsymbol{w}}_{\text{FE}}$ (Assumptions 1–3), we first generate $N = 16M$ samples $\delta_i \in [-\pi, \pi]$ according to a uniform distribution (single-path model). We then evaluate the covariance matrices $\boldsymbol{C}_{\delta_i}$ and the MMSE filters $\boldsymbol{W}_{\delta_i}$ according to (5) and (7) with a Laplace power density (88) with an angular spread of $\sigma_{\text{AS}} = 2^\circ$.

The gridded estimator $\widehat{\boldsymbol{W}}_{\text{GE}}(\cdot)$ is then given by (20). We have chosen N sufficiently large, such that, for the single-path model, the performance of the GE is close to the performance of the (non-gridded) MMSE estimator. For the SE that uses $\hat{\boldsymbol{w}}_{\text{SE}}(\cdot)$ we consider circulant and Toeplitz structure, i.e., we use the DFT matrix $\boldsymbol{Q} = \boldsymbol{F}$ for the circulant SE and the partial DFT matrix $\boldsymbol{Q} = \boldsymbol{F}_2$ for the Toeplitz SE as explained in the examples at the end of Section V-A. The coefficients $\boldsymbol{w}_{\delta_i}$ in (29) are found by solving a least-squares problem and the respective estimators can then be evaluated as specified in Theorem 1. Since we have a finite number of antennas, we expect a performance loss compared to $\widehat{\boldsymbol{W}}_{\text{GE}}$, due to violation of Assumption 2. For the fast estimator that uses $\hat{\boldsymbol{w}}_{\text{FE}}$ in (46), we use a circulant structure $\boldsymbol{Q} = \boldsymbol{F}$ and we only need to calculate $\boldsymbol{w}_0$ for $\delta = 0$ since the matrices $\boldsymbol{A}_{\text{SE}}$ in $\hat{\boldsymbol{w}}_{\text{SE}}$ are replaced by circulant convolutions.

As a baseline, we also show the MSE for the genie-aided MMSE estimator, which simply uses $\boldsymbol{W}_\delta$ for the correct δ. The per-antenna MSE of the channel estimation for the different approximations for a single snapshot ($T = 1$) is depicted in Fig. 5 as a function of the number of antennas M for a fixed SNR of 0 dB. We see, indeed, a gap between the gridded estimator and the two structured estimators. As expected, the Toeplitz SE outperforms the circulant SE. For this scenario, the FE yields performance close to the circulant SE. Apparently, the assumption of shift invariance is reasonably accurate. For a large number of antennas, the relative difference in performance of the algorithms diminishes and all algorithms get quite close to the genie-aided estimator.

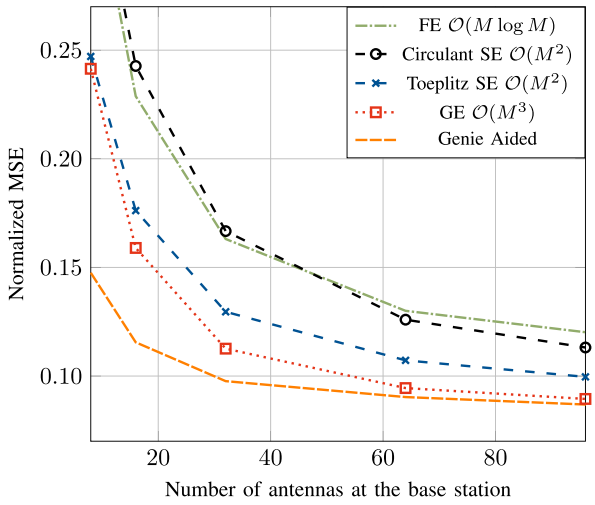

Fig. 5. MSE per antenna at an SNR of 0 dB for estimation from a single snapshot ($T = 1$). Channel model with one propagation path with uniformly distributed angle and a per path angular spread of $\sigma_{\text{AS}} = 2^\circ$.

We see that for this simple channel model, there is not much potential for our learning-based methods. However, the results change significantly for a more realistic channel model. In the following, we consider results for the 3GPP model with three propagation paths, which have different relative path gains. That is, the power density is given by

$$g_{3\text{p}}(\theta, \boldsymbol{\delta} = [\delta_1, \delta_2, \delta_3, p_1, p_2, p_3]^{\text{T}}) = \sum_{i=1}^{3} p_i g_{\text{lp}}(\theta, \delta_i) \tag{63}$$

where the angles δ_i are uniformly distributed. The path gains p_i are drawn from a uniform distribution in the interval $[0, 1]$ and then normalized such that $\sum_i p_i = 1$. The angular spread of each path is still $\sigma_{\text{AS}} = 2^\circ$.

In Figs. 6 and 7, we show the resulting normalized MSE for the numerical simulation with three propagation paths. We see that the gap between the fast estimator and the Toeplitz SE is much larger than in Fig. 5. The fast estimator does not perform well in this scenario as the shift-invariance assumption is lost when the model contains more than one propagation path.

This is where the learning-based estimators shine, since they can potentially compensate for inaccurate assumptions. We distinguish between the CNN estimator using the softmax activation function and the one with a rectified linear unit (ReLU) as activation function. In both cases, we only show results for $\boldsymbol{Q} = \boldsymbol{F}_2$, since using $\boldsymbol{Q} = \boldsymbol{F}$ lead to consistently worse results. We ran the hierarchical learning procedure described in Section VI-A for 10 000 iterations with mini-batches of 20 samples generated from the channel model. We also include

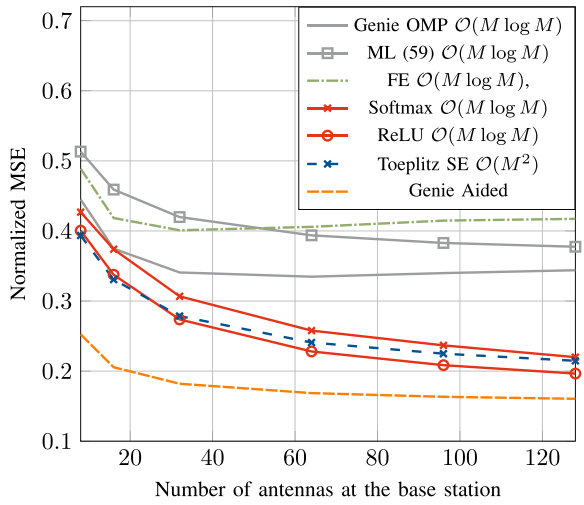

Fig. 6. MSE per antenna at an SNR of 0 dB for estimation from a single snapshot ($T = 1$). Channel model with three propagation paths (cf. (63)).

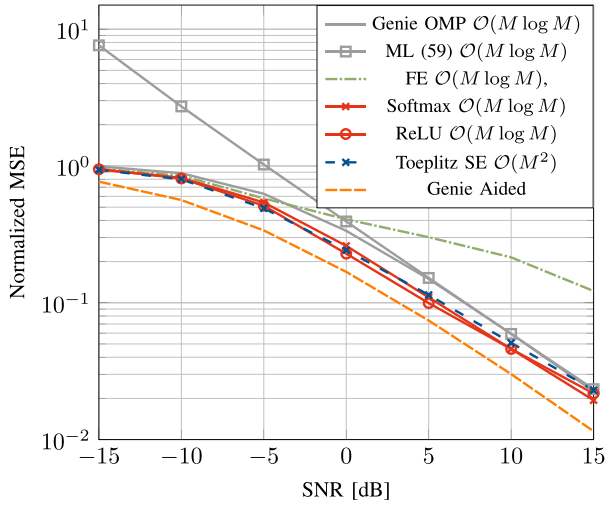

Fig. 7. MSE per antenna for M = 64 antennas and for estimation from a single snapshot ($T = 1$). Channel model with three propagation paths (cf. (63)).

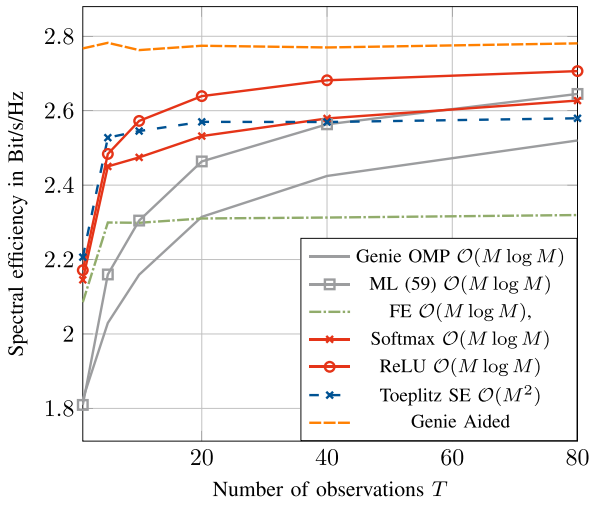

Fig. 8. Spectral efficiency for $M = 64$ antennas and an SNR of -10 dB at the cell edge. The urban macro channel model specified in [8] is used to generate the channels.

TABLE I
SIMULATION PARAMETERS FOR FIG. 8

Path-loss coefficient	3.5
Log-normal shadow fading	0 dB
Min. distance	1000 m
Max. distance	1500 m
SNR at max. distance	-10 dB

results for the ML estimator and the genie-aided OMP algorithm discussed in Sections VII-A and VII-B, respectively. For the OMP algorithm we use a four-times oversampled DFT matrix as dictionary $\boldsymbol{D}$.

The performance of the softmax-CNN estimator shows that it is, indeed, a good idea to use optimized variables instead of plug-in values that were derived under assumptions that fail to hold. It is astonishing that the ReLU-CNN estimator, which has the same computational complexity as the softmax-CNN estimator, significantly outperforms all other estimators of comparable complexity. In fact, the ReLU-CNN estimator even outperforms the more complex Toeplitz SE estimator. This can be explained by the fact that, compared to the single-path model, the number of parameters $\boldsymbol{\delta}$ is increased and the choice of $N = 16M$ samples no longer guarantees a small gridding error.

Finally, we use the 3GPP urban-macro channel model as specified in [8] with a single user placed at different positions in the cell. The parameters used in the simulation are given in Table I. In Fig. 8, we depict the performance in terms of spectral efficiency with respect to the number of available observations. Specifically, we use a matched filter in the uplink and evaluate the rate expression

$$r = \mathrm{E}\left[\log_2\left(1 + \frac{|\hat{\boldsymbol{h}}^{\mathrm{H}}\boldsymbol{h}|^2}{\sigma^2\|\hat{\boldsymbol{h}}\|^2}\right)\right] \tag{64}$$

with Monte Carlo simulations. We assume that the SNR is the same during training and data transmission. Note that this rate expression, which assumes perfect channel state information (CSI) at the decoder, yields an upper bound on the achievable rate, which is a simple measure for the accuracy of the estimated subspace. Commonly used lower bounds on the achievable rate that take the imperfect CSI into account are not straightforward to apply to our system model and are, therefore, not shown.

The performance of both the ML estimator and the ReLU-CNN estimator converge towards the genie aided estimator for large T. For a small to moderate number of observations, the CNN-based approach is clearly superior. The upper bound on the OMP performance is still lower than the performance of the circulant ML estimator.

We also see that the ReLU-CNN estimator outperforms the Toeplitz SE for a high number of observations. The reason is that we use a fixed number of samples $N = 16M$, which leads to an error floor with respect to the number of observations. In other words, for higher numbers of observations the complexity of the gridded estimator has to be increased to improve the estimation

accuracy. In contrast, the accuracy of the ReLU-CNN estimator improves just like that.

The simulation code is available online [30].

IX. Conclusion

We presented a novel approach to learn a low-complexity channel estimator, which is motivated by the structure of the MMSE estimator. In contrast to other approaches, there are no model parameters which have to be fine-tuned for different channel models. These parameters are learned from channel realizations that could be generated from a model or measured. Despite this lack of explicit fine tuning, the proposed method outperforms state-of-the-art approaches at a very low computational cost. Although we could consider more general NNs, e.g., by replacing convolution matrices with arbitrary matrices, our simulation results suggest that this is not worthwhile, at least as long as the 3GPP models are used.

It will be interesting to establish whether the NN estimators perform equally well for channel models in which the Toeplitz assumption is not satisfied. In fact, recent work [31] suggests that the model in (5) based on the far-field assumption does not provide a perfect fit when using large arrays with lots of antennas. However, the structure of the neural network is not required to perfectly fit the channel model, since the optimized variables can compensate for an inappropriate structure, at least partially. The only requirement is that suitable training data for the learning procedure is available.

Appendix

A. Proof of Lemma 1

We show Lemma 1 for the slightly more general case with arbitrary full-rank noise covariance matrices $\boldsymbol{\Sigma}$. Let $\boldsymbol{S} = T^{-1}\sum_{t=1}^{T}\boldsymbol{y}\boldsymbol{y}^{\mathrm{H}}$ denote the sample covariance matrix. The likelihood of $\boldsymbol{Y}$ in

$$\hat{\boldsymbol{h}}_{\mathrm{MMSE}} = \frac{\int p(\boldsymbol{Y}|\boldsymbol{\delta})\boldsymbol{W}_{\boldsymbol{\delta}}\, p(\boldsymbol{\delta})d\boldsymbol{\delta}}{\int p(\boldsymbol{Y}|\boldsymbol{\delta})\, p(\boldsymbol{\delta})d\boldsymbol{\delta}}\boldsymbol{y} \tag{65}$$

is proportional to (we only need to consider factors with $\boldsymbol{\delta}$, because other terms cancel out)

$$p(\boldsymbol{Y}|\boldsymbol{\delta}) \propto \prod_{t=1}^{T}\frac{\exp\big(-\boldsymbol{y}_t^{\mathrm{H}}(\boldsymbol{C}_{\boldsymbol{\delta}}+\boldsymbol{\Sigma})^{-1}\boldsymbol{y}_t\big)}{|\boldsymbol{C}_{\boldsymbol{\delta}}+\boldsymbol{\Sigma}|} \tag{66}$$

$$= \exp\big(-T\mathrm{tr}((\boldsymbol{C}_{\boldsymbol{\delta}}+\boldsymbol{\Sigma})^{-1}\boldsymbol{S})\big)\prod_{t=1}^{T}|(\boldsymbol{C}_{\boldsymbol{\delta}}+\boldsymbol{\Sigma})^{-1}|. \tag{67}$$

We express $(\boldsymbol{C}_{\boldsymbol{\delta}}+\boldsymbol{\Sigma})^{-1}$ in terms of $\boldsymbol{W}_{\boldsymbol{\delta}}$ as follows: We have

$$\boldsymbol{C}_{\boldsymbol{\delta}} = \boldsymbol{W}_{\boldsymbol{\delta}}(\boldsymbol{C}_{\boldsymbol{\delta}}+\boldsymbol{\Sigma}) \tag{68}$$

$$\Leftrightarrow\ \boldsymbol{C}_{\boldsymbol{\delta}}+\boldsymbol{\Sigma} = \boldsymbol{W}_{\boldsymbol{\delta}}(\boldsymbol{C}_{\boldsymbol{\delta}}+\boldsymbol{\Sigma})+\boldsymbol{\Sigma} \tag{69}$$

$$\Leftrightarrow\ \mathbf{I} = \boldsymbol{W}_{\boldsymbol{\delta}}+\boldsymbol{\Sigma}(\boldsymbol{C}_{\boldsymbol{\delta}}+\boldsymbol{\Sigma})^{-1} \tag{70}$$

$$\Leftrightarrow\ \boldsymbol{\Sigma}^{-1}(\mathbf{I}-\boldsymbol{W}_{\boldsymbol{\delta}}) = (\boldsymbol{C}_{\boldsymbol{\delta}}+\boldsymbol{\Sigma})^{-1}. \tag{71}$$

If we plug this expression into the likelihood (67), we obtain

$$p(\boldsymbol{Y}|\boldsymbol{\delta}) \propto \exp\big(-T\,\mathrm{tr}(\boldsymbol{\Sigma}^{-1}(\mathbf{I}-\boldsymbol{W}_{\boldsymbol{\delta}})\boldsymbol{S})\big)\prod_{t=1}^{T}|\boldsymbol{\Sigma}^{-1}(\mathbf{I}-\boldsymbol{W}_{\boldsymbol{\delta}})| \tag{72}$$

$$\propto \exp\big(T\,\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{W}_{\boldsymbol{\delta}}\boldsymbol{S})\big)\prod_{t=1}^{T}|\,\mathbf{I}-\boldsymbol{W}_{\boldsymbol{\delta}}| \tag{73}$$

$$= \exp\big(T\,\mathrm{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{W}_{\boldsymbol{\delta}}\boldsymbol{S}) + T\log|\,\mathbf{I}-\boldsymbol{W}_{\boldsymbol{\delta}}|\big) \tag{74}$$

since $\boldsymbol{\Sigma}$ does depend on $\boldsymbol{\delta}$. If we substitute $\boldsymbol{\Sigma} = \sigma^2\,\mathbf{I}$ and $\widehat{\boldsymbol{C}} = T/\sigma^2\boldsymbol{S}$, Lemma 1 follows.

B. Uniform Linear Array

For a uniform linear array (ULA) with half-wavelength spacing at the base station, the steering vector is given by

$$\boldsymbol{a}(\theta) = \big[1,\ \exp(i\pi\sin\theta),\ \ldots,\ \exp(i\pi(M-1)\sin\theta)\big]^{\mathrm{H}}. \tag{75}$$

Consequently, the covariance matrix has Toeplitz structure with entries

$$[\boldsymbol{C}_{\boldsymbol{\delta}}]_{mn} = \int_{-\pi}^{\pi} g(\theta;\boldsymbol{\delta})\exp(-i\pi(m-n)\sin\theta)d\theta. \tag{76}$$

If we substitute $u = \pi\sin\theta$, we get

$$[\boldsymbol{C}_{\boldsymbol{\delta}}]_{mn} = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(u;\boldsymbol{\delta})\exp(-i(m-n)u)du \tag{77}$$

with

$$f(u;\boldsymbol{\delta}) = 2\pi\frac{g(\arcsin(u/\pi);\boldsymbol{\delta}) + g(\pi-\arcsin(u/\pi);\boldsymbol{\delta})}{\sqrt{\pi^2-u^2}} \tag{78}$$

where we extended g periodically beyond the interval $[-\pi,\pi]$. That is, the entries of the channel covariance matrix are Fourier coefficients of the periodic spectrum $f(u;\boldsymbol{\delta})$.

An interesting property of the Toeplitz covariance matrices is that we can define a circulant matrix $\widetilde{\boldsymbol{C}}_{\boldsymbol{\delta}}$ with the eigenvalues $f(2\pi k/M;\boldsymbol{\delta})$, $k = 0,\ldots,M-1$, such that $\widetilde{\boldsymbol{C}}_{\boldsymbol{\delta}} \asymp \boldsymbol{C}_{\boldsymbol{\delta}}$ [16]. That is, to get the elements of the circulant matrices, we approximate the integral in (77) by the summation

$$[\widetilde{\boldsymbol{C}}_{\boldsymbol{\delta}}]_{mn} = \frac{1}{M}\sum_{k=0}^{M-1} f(2\pi k/M;\boldsymbol{\delta})e^{-i(m-n)2\pi k/M}. \tag{79}$$

C. Uniform Rectangular Array

To work with a two-dimensional array, we need a three-dimensional channel model. That is, in addition to the azimuth angle θ, we also need an elevation angle ϕ to describe a direction of arrival. Under the far-field assumption, the covariance matrix is given by

$$\boldsymbol{C}_{\boldsymbol{\delta}} = \int_{-\pi/2}^{\pi/2}\int_{-\pi}^{\pi} g(\theta,\phi;\boldsymbol{\delta})\boldsymbol{a}(\theta,\phi)\boldsymbol{a}(\theta,\phi)^{\mathrm{H}}d\theta d\phi. \tag{80}$$

For a uniform rectangular array (URA) with half-wavelength spacing at the base station, we have $M = M_H M_V$ antenna elements, where M_H is the number of antennas in the horizontal

direction and M_V the number of antennas in the vertical direction. The correlation between the antenna element at position (m,p) and the one at (n,q), given the parameters $\boldsymbol{\delta}$, is given by

$$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}}\int_{-\pi}^{\pi} g(\theta,\phi;\boldsymbol{\delta})e^{i\pi((n-m)\sin\theta+(q-p)\cos\theta\sin\phi)}d\theta d\phi \quad (81)$$

$$=\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \tilde{g}(\theta,\phi;\boldsymbol{\delta})e^{i\pi((n-m)\sin\theta+(q-p)\cos\theta\sin\phi)}d\theta d\phi \quad (82)$$

where

$$\tilde{g}(\theta,\phi;\boldsymbol{\delta}) = g(\theta,\phi;\boldsymbol{\delta}) + g(\pi-\theta,\phi;\boldsymbol{\delta}). \quad (83)$$

We can map the square $[-\pi/2,\pi/2]^2$ bijectively onto the circle with radius π with the substitution $u=\pi\sin\theta$ and $\nu=\pi\cos\theta\sin\phi$. The transformed integral can be written as

$$\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} f(u,\nu;\boldsymbol{\delta})e^{-i\pi((m-n)u+(p-q)\nu)}du\,d\nu \quad (84)$$

with

$$f(u,\nu;\boldsymbol{\delta}) = \begin{cases} \tilde{f}(u,\nu;\boldsymbol{\delta}), & \text{for } u^2+\nu^2\le\pi^2, \\ 0, & \text{otherwise.} \end{cases} \quad (85)$$

The nonzero entries of the two dimensional spectrum are given by

$$\tilde{f}(u,\nu;\boldsymbol{\delta}) = \frac{\tilde{g}(\arcsin(u/\pi),\arcsin(\nu/(\pi\sqrt{1-u^2}))}{\sqrt{(\pi^2-u^2)(\pi^2-u^2-\nu^2)}}. \quad (86)$$

That is, for a URA, the entries of the channel covariance matrix are two-dimensional Fourier coefficients of the periodic spectrum $f(u,\nu;\boldsymbol{\delta})$.

We can use the results for the ULA case to show that the URA covariance matrix is asymptotically equivalent to a nested circulant matrix with the eigenvalues $f(2\pi m/M_H, 2\pi p/M_V;\boldsymbol{\delta})$ where $m=0,\dots,M_H-1$ and $p=0,\dots,M_V-1$. The eigenvectors of the nested circulant matrix are given by $\boldsymbol{F}_{M_H}\otimes\boldsymbol{F}_{M_V}$ where $\boldsymbol{F}_M$ denotes the M-dimensional DFT matrix.

To show the asymptotic equivalence, we first replace the Toeplitz structure along the horizontal direction by a circulant structure. This yields an asymptotically equivalent matrix due to the results from [16]. Second, we replace the Toeplitz structure along the vertical direction by a circulant structure to get the desired result. Clearly, the asymptotic equivalence only holds if M_H and M_V both go to infinity.

D. Shift Invariance

To get circulant matrices $\boldsymbol{A}_{\text{SE}}$ in the structured estimator $\hat{\boldsymbol{w}}_{\text{SE}}$ in (32), we need several assumptions. First, we assume that the circulant approximation in (79) holds exactly, i.e., the columns $\boldsymbol{w}_i$ of $\boldsymbol{A}_{\text{SE}}$ contain uniform samples of the continuous filter

$$w(u;\boldsymbol{\delta}_i) = \frac{f(u;\boldsymbol{\delta}_i)}{f(u;\boldsymbol{\delta}_i)+\sigma^2}. \quad (87)$$

Next, we assume a single parameter δ and shift invariance of the spectrum, i.e., $f(u;\delta)=f(u-\delta)$ from which $w(u;\delta)=w(u-\delta)$ follows. Finally, the prior of δ has to be uniform on the same grid that generates the samples of the $\boldsymbol{w}_i$.

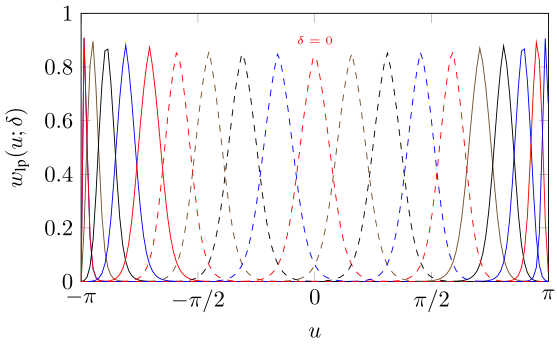

Fig. 9. Functions $w_{\text{lp}}(u;\delta)$ for different $\delta\in[-\pi/2,\pi/2]$ sampled on a uniform grid. The peaks of the graphs are at $\pi\sin\delta$. The graphs for $\delta\in[-\pi/4,\pi/4]$ are depicted with a dashed line-style.

Example: An example that approximately fulfills these assumptions is the 3GPP spatial channel model for a ULA with only a single propagation path. In this case, we only have one parameter for the covariance matrix: the angle of the path center δ, which is uniformly distributed. The power density function of the angle of arrival (cf. (5)) is given by the Laplace density

$$g_{\text{lp}}(\theta;\delta) = \exp\left(-d_{2\pi}(\delta,\theta)/\sigma_{\text{AS}}\right) \quad (88)$$

where $d_{2\pi}(\theta,\delta)$ is the wrap-around distance between θ and δ and can be thought of as $|\theta-\delta|$ for most (θ,δ) pairs. In other words, for different δ, the function $g_{\text{lp}}(\theta;\delta)$ is simply a shifted version of $g_{\text{lp}}(\theta;0)$, i.e., $g_{\text{lp}}(\theta;\delta)=g_{\text{lp}}(\theta-\delta;0)$.

Due to the symmetry of the ULA, we can restrict the parameter δ to the interval $[-\pi/2,\pi/2]$ without loss of generality. For angles $\delta\in[-\pi/4,\pi/4]$, i.e., if the cluster center is located at the broadside of the array, the arcsin-transform is approximately linear. As a consequence, the correspondence (88) is approximately true also for the transformed spectrum $f(u;\boldsymbol{\delta})$ (cf. (78)) and, by virtue of (87), also the continuous filter is approximately shift-invariant. This discussion is illustrated by Fig. 9, which shows the continuous filter $w_{\text{lp}}(\cdot;\delta)$ for different δ (the peaks are at $\pi\sin\delta$). For $\delta\in[-\pi/4,\pi/4]$, the different filters are approximately shifted versions of the central filter, i.e.,

$$w_{\text{lp}}(u;\delta)\approx w_{\text{lp}}(u-\delta;0). \quad (89)$$

For large M, the approximation error from using (79) is reduced and we can approximate the matrix $\boldsymbol{A}_{\text{SE}}$ by a circular convolution with uniform samples $\boldsymbol{w}_0$ of $w_{\text{lp}}(u;0)$ as convolution kernel. We get a *fast estimator* $\hat{\boldsymbol{w}}_{\text{FE}}$ by setting $\boldsymbol{A}_{\text{SE}}$ in the structured estimator $\hat{\boldsymbol{w}}_{\text{SE}}$ to

$$\boldsymbol{A}_{\text{SE}} = \boldsymbol{F}^{\text{H}}\,\text{diag}\,(\boldsymbol{F}\boldsymbol{w}_0)\boldsymbol{F} \quad (90)$$

where $[\boldsymbol{w}_0]_k = w_{\text{lp}}(2\pi(k-1)/K;0)$.

An analogous shift invariance can be derived for a uniform rectangular array. In this case, we have a two-dimensional shift-invariance and, thus, two-dimensional convolutions. For the case of distributed antennas that appeared in the examples in Section V-A, we do not see a straightforward way to make a similar simplification.

REFERENCES

[1] F. Rusek *et al.*, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

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

[3] D. C. Araujo, A. L. F. de Almeida, J. Axnas, and J. C. M. Mota, "Channel estimation for millimeter-wave very-large MIMO systems," in *Proc. Eur. Signal Process. Conf.*, Sep. 2014, pp. 81–85.

[4] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 831–846, Oct. 2014.

[5] M. K. Samimi and T. S. Rappaport, "3-D millimeter-wave statistical channel model for 5G wireless system design," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 7, pp. 2207–2225, Jul. 2016.

[6] S. Sun *et al.*, "Investigation of prediction accuracy, sensitivity, and parameter stability of large-scale propagation path loss models for 5G wireless communications," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 2843–2860, May 2016.

[7] H. Yin, D. Gesbert, M. Filippou, and Y. Liu, "A coordinated approach to channel estimation in large-scale multiple-antenna systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 264–273, Feb. 2013.

[8] 3GPP, "Spatial channel model for multiple input multiple output (MIMO) simulations (release 12)," *3rd Generation Partnership Project (3GPP), TR 25.996 V12.0.0*, 2014.

[9] X. Rao and V. K. N. Lau, "Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3261–3271, Jun. 2014.

[10] S. Haghighatshoar and G. Caire, "Massive MIMO channel subspace estimation from low-dimensional projections," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 303–318, Jan. 2017.

[11] A. Omri, R. Bouallegue, R. Hamila, and M. Hasna, "Channel estimation for LTE uplink system by perceptron neural network," *Int. J. Wireless Mobile Netw.*, vol. 2, no. 3, pp. 155–165, 2010.

[12] L. Zhang and X. Zhang, "MIMO channel estimation and equalization using three-layer neural networks with feedback," *Tsinghua Sci. Technol.*, vol. 12, no. 6, pp. 658–662, 2007.

[13] R. Prasad, C. R. Murthy, and B. D. Rao, "Joint approximately sparse channel estimation and data detection in OFDM systems using sparse Bayesian learning," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3591–3603, Jul. 2014.

[14] X. Zhou and X. Wang, "Channel estimation for OFDM systems using adaptive radial basis function networks," *IEEE Trans. Veh. Technol.*, vol. 52, no. 1, pp. 48–59, Jan. 2003.

[15] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1980.

[16] R. M. Gray, "Toeplitz and circulant matrices: A review," *Found. Trends Commun. Inf. Theory*, vol. 2, no. 3, pp. 155–239, 2006.

[17] C. L. Epstein, " How well does the finite Fourier transform approximate the Fourier transform?" *Commun. Pure Appl. Math.*, vol. 58, pp. 1421–1435, Oct. 2005.

[18] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO: Uniformly great service for everyone," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun.*, Jun. 2015, pp. 201–205.

[19] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, pp. 1834–1850, Mar. 2017.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[21] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, Fort Lauderdale, FL, USA, 2011, pp. 315–323.

[22] M. Frigge, D. C. Hoaglin, and B. Iglewicz, "Some implementations of the boxplot," *Amer. Statistician*, vol. 43, pp. 50–54, 1989.

[23] D. Neumann, M. Joham, L. Weiland, and W. Utschick, "Low-complexity computation of LMMSE channel estimates in massive MIMO," in *Proc. 19th Int. ITG Workshop Smart Antennas*, Mar. 2015, pp. 1–6.

[24] J. P. Burg, D. G. Luenberger, and D. L. Wenger, "Estimation of structured covariance matrices," *Proc. IEEE*, vol. 70, no. 9, pp. 963–974, Sep. 1982.

[25] T. W. Anderson, "Asymptotically efficient estimation of covariance matrices with linear structure," *Ann. Statist.*, vol. 1, pp. 135–141, 1973.

[26] A. Dembo, "The relation between maximum likelihood estimation of structured covariance matrices and periodograms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 6, pp. 1661–1662, Dec. 1986.

[27] T. Wiese, L. Weiland, and W. Utschick, "Low-rank approximations for spatial channel models," in *Proc. 20th Int. ITG Workshop Smart Antennas*, Münich, Germany, Mar. 2016, pp. 1–5.

[28] M. Gharavi-Alkhansari and T. S. Huang, "A fast orthogonal matching pursuit algorithm," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 3, May 1998, pp. 1389–1392.

[29] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, Nov. 2009.

[30] D. Neumann and T. Wiese, "Simulation code," 2017. [Online]. Available: https://github.com/tum-msv/learning-mmse-est

[31] X. Gao, O. Edfors, F. Tufvesson, and E. G. Larsson, "Massive MIMO in real propagation environments: Do all antennas contribute equally?" *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 3917–3928, Nov. 2015.

David Neumann received the Dipl.-Ing. degree in electrical engineering from Technische Universität München (TUM), München, Germany, in 2011. He is currently working toward the doctoral degree at the Professorship for Signal Processing, TUM. His research interests include transceiver design for large-scale communication systems and estimation theory.

Thomas Wiese (S'15) received the Dipl.-Ing. degree in electrical engineering and the Dipl.-Math. degree in mathematics from Technische Universität München (TUM), München, Germany, in 2011 and 2012, respectively. He is currently working toward the doctoral degree at the Professorship for Signal Processing, TUM. His research interests include compressive sensing, sensor array processing, and convex optimization.

Wolfgang Utschick (SM'06) completed several years of industrial training programs before he received the Diploma degree in 1993 and the doctoral degree (both with Hons.) in 1998 in electrical engineering with a dissertation on machine learning, from Technische Universität München (TUM), München, Germany. Since 2002, he is a Professor at TUM where he is chairing the Professorship of Signal Processing. He teaches courses on signal processing, stochastic processes, and optimization theory in the field of wireless communications, various application areas of signal processing, and power transmission systems. Since 2011, he has been a regular Guest Professor at Singapore's new autonomous university, Singapore Institute of Technology. He holds several patents in the field of multiantenna signal processing and has authored and coauthored a large number of technical articles in international journals and conference proceedings and has been awarded with a couple of best paper awards. He edited several books and is a Founder and the Editor of the Springer book series Foundations in Signal Processing, Communications and Networking. He has been a Principal Investigator in multiple research projects funded by the German Research Fund and a Coordinator of the German DFG priority program Communications Over Interference Limited Networks. He is a member of the VDE and therein a member of the Expert Group 5.1 for Information and System Theory of the German Information Technology Society. He is currently chairing the German Signal Processing Section. He has also been serving as an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING and has been member of the IEEE Signal Processing Society Technical Committee on Signal Processing for Communications and Networking. Since 2017, he serves as the Dean of the Department for Electrical and Computer Engineering, TUM.