Learning the MMSE Channel Estimator

David Neumann, Thomas Wiese, and Wolfgang Utschick
Professur für Methoden der Signalverarbeitung, Technische Universität München, 80290 Munich, Germany
{d.neumann,thomas.wiese,utschick}@tum.de

Abstract—We present an $O(M \log M)$ method for estimating $M$-dimensional structured channels that uses techniques from the field of machine learning. Our channel model is typical in communications: the channel vector is normal distributed given an unknown covariance matrix, which depends on random hyperparameters such as the angles of the propagation paths. If the channel model exhibits certain Toeplitz and shift-invariance structures, the complexity of the MMSE estimator can be reduced to $O(M \log M)$; otherwise, it is much higher. To obtain an $O(M \log M)$ estimator for the general case, we use the structure of this specific MMSE estimator as an informed guess for the architecture of a neural network. We discuss how this network can be efficiently trained with channel realizations to learn the MMSE estimator in the class of $O(M \log M)$ estimators.

Index Terms—channel estimation; MMSE estimation; machine learning; neural networks; spatial channel model

I. INTRODUCTION

Accurate channel estimation is a major challenge in wireless communication networks. In setups with many antennas and low signal to noise ratios, errors in the channel estimates are particularly devastating because they reduce the array gain. Since such systems play an important role in the next generations of wireless networks, e.g., in cellular massive MIMO [1]–[3] or millimeter-wave [4], [5] networks, there is a need for accurate channel estimators with low complexity. Preferably, the computational complexity scales similarly to the hardware complexity, i.e., approximately linear in the number of antennas.

The accuracy of channel estimation can be increased by exploiting structure in the spatial correlation of the channel vectors. These are commonly modelled as random (infinite) linear combinations of steering vectors [6], [7]. For typical array geometries, e.g., uniform linear or rectangular arrays, the corresponding covariance matrix has a Toeplitz structure.

Methods from the area of compressive sensing attempt to approximate the channel vector as a linear combination of $k$ steering vectors where $k$ is much smaller than the number of antennas $M$, e.g., [5], [8]. A complexity of $O(M \log M)$ can be achieved with these methods if efficient implementations are used.

The Toeplitz structure lead the authors of [9] to construct the maximum likelihood estimator of the channel covariance matrix within the class of all positive semi-definite Toeplitz matrices, which is then used to estimate the channel. However, even the low-complexity version of this estimator is given as the solution of a convex program with $M$ variables, i.e., the complexity is $O(p(M))$ where $p$ is a polynomial.

We take a different approach to include the spatial channel model in the estimation procedure: we firstly attempt to calculate the minimum mean squared error (MMSE) estimator within a given class of $O(M \log M)$ estimators. The class is chosen in a way that the $O(M \log M)$ MMSE estimator (approximately) coincides with the MMSE estimator for a certain channel model. Namely, a channel model with only a single cluster of propagation paths with a uniformly distributed cluster center. The idea is similar to the concept of the linear MMSE (LMMSE) estimator, which is the MMSE estimator within the class of all linear estimators. This estimator is optimal for Gaussian random vectors and considered a useful estimator also for distributions that deviate from Gaussian [10]. [11]. Similarly, the $O(M \log M)$ MMSE estimator performs well for channel models that deviate from the single-cluster assumption.

In summary, our main contributions are the following

- We derive the MMSE estimator for conditionally normal channel models.
- We show how the complexity of the MMSE estimator can be reduced to $O(M \log M)$ if the channel covariance matrices are Toeplitz and have a shift-invariance structure.
- We provide a principled approach to design a neural network that approximates the best $O(M \log M)$ MMSE estimator for general channel models.
- We show how the neural network can be efficiently trained in a hierarchical way so as to avoid sub-optimal estimators due to local optima.

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 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

$$y[t] = h[t] + z[t], \quad t = 1, \ldots, T$$

(1)

with the channel vectors $h[t]$ and additive Gaussian noise $z[t] \sim \mathcal{N}_C(0, C_z)$. For the major part of this work, we assume that the noise covariance is a scaled identity $C_z = \sigma^2 I$ with known $\sigma^2$. The channel vectors are assumed to be conditionally Gaussian distributed given a set of parameters $\delta$, i.e., $h[t] \| \delta \sim \mathcal{N}_C(0, C_\delta)$. In contrast to the fast fading channel vectors, the covariance matrix $C_\delta$ is assumed to be constant over the $T$ channel coherence intervals. The parameters, which describe, e.g., angles of propagation paths, are also considered as random variables, with distribution $\delta \sim p(\delta)$, which is known.

A shorter version of this paper was presented at the 21st International ITG Workshop on Smart Antennas (WSA), Berlin, Germany, 2017
In summary, we have
\[ y[t] | h[t] \sim \mathcal{N}(0, C_z) \]  
(2)
with known noise covariance matrix \( C_z \) and the hierarchical prior
\[ h[t] | \delta \sim \mathcal{N}(0, C_\delta), \ \delta \sim p(\delta). \]  
(3)
with the known prior \( p(\delta) \) of the hyperparameters.

We will use additional structure of the channel model to reduce the computational complexity of the MMSE and ML estimators. In typical channel models for communication scenarios, e.g., those defined by the ETSI 3rd Generation Partnership Project (3GPP) for cellular networks, the covariance matrices are of the form
\[ C_\delta = \int_{-\pi}^{\pi} g(\theta; \delta) a(\theta) a(\theta)^H d\theta, \]  
(4)
where \( g(\theta; \delta) \geq 0 \) is a power density function corresponding to the parameter \( \delta \) and where \( a(\theta) \) denotes the array manifold vector of the antenna array at the base station for an angle \( \theta \). As an example, in the 3GPP urban micro and urban macro scenarios, \( g(\theta; \delta) \) is a superposition of several scaled probability density functions (pdf) of a Laplace-distributed random variable with standard deviation \( 2^\circ \) or \( 5^\circ \), respectively. Each Laplace function corresponds to one propagation path. The Laplace density models the scattering of the received power around the center of the propagation path.

For a uniform linear array (ULA) with half-wavelength spacing at the base station, the steering vector is given by
\[ a(\theta) = [1, \exp(i\pi \sin \theta), \ldots, \exp(i\pi(M-1) \sin \theta)]^H. \]  
(5)
Consequently, given a ULA at the base station, the covariance matrix has Toeplitz structure with entries
\[ [C_\delta]_{mn} = \int_{-\pi}^{\pi} g(\theta; \delta) \exp(-i\pi(m-n) \sin \theta) d\theta. \]  
(6)
If we substitute \( \omega = \pi \sin \theta \), we get
\[ [C_\delta]_{mn} = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\omega; \delta) \exp(-i(m-n) \omega) d\omega. \]  
(7)
with
\[ f(\omega; \delta) = \frac{2\pi g(\arcsin(\omega/\pi); \delta) + g(\pi - \arcsin(\omega/\pi); \delta)}{\sqrt{\pi^2 - \omega^2}}. \]  
(8)
That is, the entries of the channel covariance matrix are Fourier coefficients of the periodic spectrum \( f(\omega; \delta) \).

An interesting property of Toeplitz matrices is that, if appropriately scaled, they are asymptotically equivalent to circulant Toeplitz matrices in the sense that
\[ \|C_\delta - \hat{C}_\delta\|_F^2 / M \to 0 \]  
(9)
where \( \hat{C}_\delta \) denotes the circulant matrix with the eigenvalues \( f(2\pi(i-1)/M; \delta), i = 1, \ldots, M \) [12]. This result motivates the approximations made in the following low-complexity approaches to channel estimation.

### III. Estimation of Conditionally Normal Channels

Our goal is to estimate \( h[t] \) for each \( t \) given all observations \( Y = [y[1], \ldots, y[T]] \) and using the model \( 2, 3 \). For a fixed parameter \( \delta \), the MMSE estimator can be obtained analytically, since conditioned on \( \delta \), the observation \( y[t] \) is jointly Gaussian distributed with the channel vector \( h[t] \). Also, given the parameters \( \delta \), the observations of different coherence intervals are independent. The conditional MMSE estimate of the channel vector \( h[t] \) is
\[ \hat{h}_\delta[t] = E[h[t] | y[t], \delta] = W_\delta y[t] \]  
(10)
with
\[ W_\delta = C_\delta(C_\delta + C_z)^{-1}. \]  
(11)
Since in our model the channel covariance matrix \( C_\delta \) is unknown, we either have to find an estimate \( \hat{C}_\delta \) of the covariance matrix and replace \( C_\delta \) in \( 11 \) with the estimate \( \hat{C}_\delta \) or we incorporate the prior \( p(\delta) \) into the MMSE estimation.

#### A. ML Estimation

Before we discuss the MMSE estimation with known prior \( p(\delta) \), we take a look at the common maximum likelihood (ML) covariance matrix estimators, which we will use as a baseline in our numerical simulations. ML estimation can be applied in absence of the prior \( p(\delta) \). We first find an ML estimate of the channel covariance matrix \( C_{ML} \) based on the observations \( Y \) and then, assuming the estimate is exact, calculate the MMSE estimates of the channel vectors as in \( 10, 11 \).

The likelihood function for the channel covariance matrix given the noise covariance matrix is given by
\[ L(C_\delta | Y) = \frac{\exp \left( -\sum_{t=1}^{T} |y[t]|^2 (C_\delta + C_z)^{-1} y[t] \right)}{|C_\delta + C_z|^T} \]  
(12)
and the ML problem reads as
\[ \max_{C_\delta \in M} \log L(C_\delta | Y) \]  
(13)
where \( M \) is the set of admissible covariance matrices which has to be part of the set of positive semi-definite matrices \( S_+^M \). If \( M = S_+^M \), the ML estimate is given in terms of the sample covariance matrix
\[ \hat{C} = \frac{1}{T} \sum_{t=1}^{T} y[t] y[t]^H \]  
(14)
as
\[ C_{ML} = C_z^{1/2} P_{S_+^M} \left( C_z^{-1/2} \hat{C} C_z^{-1/2} - I \right) C_z^{1/2} \]  
(15)
where we use the projection \( P_{S_+^M} (\cdot) \) onto the cone of positive semi-definite matrices \[ 13 \]. For \( C_z = \sigma^2 I \) the estimate simplifies to
\[ C_{ML} = P_{S_+^M} \left( \hat{C} - \sigma^2 I \right). \]  
(16)
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 $\mathcal{M} = T_0^+$ 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 [9, 13, 15].

Since we are interested in low-complexity estimators we approximate the solution by reducing the constraint set to the set of positive semi-definite circulant Toeplitz matrices $\mathcal{M} = \mathcal{C}^+$. This choice reduces the complexity of the ML estimator significantly [13]. The reason is that all circulant matrices have as eigenvectors the columns of the discrete Fourier transform (DFT) matrix $F$.

Incorporating (17) into the likelihood function for (12), we notice that the estimate of the channel covariance matrix can be given in terms of the estimated power spectrum as

$$C_{\delta}^{ML} = F^{H} \text{diag}(c_{\delta}^{ML}) F$$

where $c_{\delta}^{ML} \in \mathbb{R}^M$ contains the $M$ eigenvalues of $C_{\delta}^{ML}$.

The estimate of the channel covariance matrix can be given in terms of the estimated power spectrum as

$$\hat{c} = \frac{1}{T} \sum_{t=1}^{T} |Fy[t]|^2$$

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

$$c_{\delta}^{ML} = [\hat{c} - \sigma^2 1]_+$$

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

$$\hat{h}[t] = F^{H} \text{diag}(c_{\delta}^{ML}) \text{diag}(\hat{c}_{\delta}^{ML} + \sigma^2 1)^{-1} F y[t]$$

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

B. MMSE Estimation

In the following, we derive the MMSE estimator for a known prior $p(\delta)$ and a known mapping $\delta \mapsto C_{\delta}$ of the hyperparameters to the covariance matrices. The MMSE estimate of $\hat{h}[t]$ given $Y = [y[1], \ldots, y[T]]$ can be reformulated to

$$\hat{h}[t] = E[h[t] | Y]$$

where $E$ stands for the expected value with respect to the prior $p(\delta)$. In summary, we can write the MMSE filter as

$$\hat{W} = \frac{E_{\delta} \left[ p(Y | \delta) W_{\delta} \right]}{E_{\delta} \left[ p(Y | \delta) \right]}$$

where $E_{\delta}$ stands for the expected value with respect to the prior $p(\delta)$. The MMSE estimation in (25) can be interpreted as follows. We first calculate the MMSE filter $\hat{W}$ as a convex combination of filters for known covariance matrices and then apply the resulting filter to the observation. Clearly, the challenging part of the MMSE estimation is the calculation of the MMSE filter from the observations $Y$.

Since the observations $y[t]$ are normal distributed given the parameters $\delta$, we get the following result for the MMSE filter.

**Lemma 1. With the noise covariance matrix $C_{\delta} = \sigma^2 I$ and the sample covariance matrix $\hat{C} = \frac{1}{T} \sum_{t=1}^{T} y[t] y^T[t]$, the MMSE filter $\hat{W}$ from (27) can be calculated as

$$\hat{W} = \frac{E_{\delta} \left[ \exp \left( \frac{1}{2} \text{tr}(W_{\delta} \hat{C}) + T \log |I - W_{\delta}| \right) W_{\delta} \right]}{E_{\delta} \left[ \exp \left( \frac{1}{2} \text{tr}(W_{\delta} \hat{C}) + T \log |I - W_{\delta}| \right) \right]}$$

with $W_{\delta} = C_{\delta}(C_{\delta} + \sigma^2 I)^{-1}$.**

Proof. See Appendix A.

Note that the sample covariance matrix $\hat{C}$ is a sufficient statistic to calculate the MMSE filter $\hat{W}$. That is, we have the basic structure

$$h[t] = \hat{W}(\hat{C}) g[t]$$

which will be important later on.

C. Computing the MMSE Filter

If we assume that, as functions of $\delta$, the prior pdf $p(\delta)$ and the entries of the matrix $W_{\delta}$ are Riemann-integrable, we can evaluate the expectations with arbitrary precision by numerical integration. The matrices $W_{\delta}$ and the log-det expressions

$$b_{\delta} = T \log |I - W_{\delta}|$$

can be computed offline on a grid for $\delta$. That is, we choose grid points $\delta_i$, for $i = 1, \ldots, N$, by sampling from the prior $p(\delta)$ and then evaluate

$$\hat{W} \approx \frac{1}{N} \sum_{i=1}^{N} \exp \left( \frac{1}{2} \text{tr}(W_{\delta_i} \hat{C}) + b_{\delta_i} \right) W_{\delta_i}.$$
IV. LOW-COMPLEXITY APPROXIMATION OF THE MMSE FILTER

Our approach to reduce the complexity required to calculate the MMSE filter can be broken down into two steps. We first exploit a common structure of the covariance matrices which follows from the geometry of the antenna array. For our channel model and if a ULA is used at the base station, the covariance matrices have Toeplitz structure. But the approach can also be applied to other array geometries, e.g., uniform rectangular arrays. The common structure allows us to find a low-dimensional parametrization of the MMSE estimator, i.e., a structured MMSE estimator, which can be implemented efficiently. 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. As shift invariance leads to circulant matrices, we can use the DFT to calculate the parameters of the structured MMSE estimator with low complexity.

A. Step One: A Structured MMSE Estimator

In the first step, we replace the filters \( W_\delta \) with structured approximations that use only \( \mathcal{O}(M) \) parameters. Specifically, we make the following assumption.

**Assumption 1.** The filters \( W_\delta \) can be decomposed as

\[
W_\delta = Q^H \text{diag}(w_\delta)Q
\]

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

Note that the requirement \( K = \mathcal{O}(M) \) ensures the desired dimensionality reduction. Without this requirement we can clearly represent arbitrary matrices with the structure in (32) for \( K = M^2 \).

If we replace the filters \( W_\delta \) in (28) with the parametrization in (32), we can simplify the trace expressions according to

\[
\text{tr}(W_\delta \hat{C}) = \text{tr}((\text{diag}(w_\delta)Q\hat{C}Q^H)) = w_\delta^T \hat{c}
\]

where \( \hat{c} \) contains the diagonal elements of the matrix \( Q\hat{C}Q^H \). If we plug in the expression for the sample covariance matrix \( \hat{C} \), we can write \( \hat{c} \) as

\[
\hat{c} = \frac{1}{T} \sum_{t=1}^{T} |Qy[t]|^2.
\]

Consequently, since the sample-covariance matrix \( \hat{C} \) appears in (28) only in terms of (32), the MMSE estimator simplifies to

\[
\hat{W} = Q^H \text{diag}(\hat{w}(\hat{c}))Q
\]

with the optimal filter

\[
\hat{w}(\hat{c}) = \frac{\exp \left( \frac{1}{T^2} w_\delta^T \hat{c} + b_\delta \right) w_\delta}{\mathbb{E}_\delta \left[ \exp \left( \frac{1}{T^2} w_\delta^T \hat{c} + b_\delta \right) \right]},
\]

where \( b_\delta = T \log(1 - \text{diag}(w_\delta)) \). Note that \( \hat{w} \) is a function from \( \mathbb{R}^K \) to \( \mathbb{R}^K \).

For the numerical evaluation of the expectations in (36) we construct a matrix

\[
A = [w_{\delta_1}, \ldots, w_{\delta_N}] \in \mathbb{R}^{K \times N}
\]

and a vector

\[
b = [b_{\delta_1}, \ldots, b_{\delta_N}]^T \in \mathbb{R}^N
\]

that consist of all vectors \( w_{\delta_i} \) and scalars \( b_{\delta_i} \), respectively, for all sampled grid points \( \delta_i \). The numerical approximation of the filter (36) evaluates to

\[
\hat{w}(\hat{c}) = A \frac{\exp \left( \frac{T}{\pi^2} A^T \hat{c} + b \right)}{1^T \exp \left( \frac{T}{\pi^2} A^T \hat{c} + b \right)}
\]

where \( \exp \) is applied element-wise.

The MMSE estimates of the channel vectors using the structured MMSE estimator can be calculated as

\[
\hat{h}[t] = Q^H \text{diag}(\hat{w}(\hat{c}))Qy[t].
\]

Given \( \hat{w}(\hat{c}) \), the complexity of the estimator depends only on the number of operations required to calculate matrix vector products with \( Q \) and \( Q^H \). To achieve the desired complexity of \( \mathcal{O}(M \log M) \) operations, the matrix \( Q \) must have some special structure that enables fast computations. If this is the case, the complexity of the structured MMSE estimator is dominated by the complexity to calculate \( \hat{w}(\hat{c}) \), which is \( \mathcal{O}(NK) \). Reducing the complexity of the calculation of \( \hat{w}(\hat{c}) \) is the topic of the next subsection.

**Examples.** As mentioned before, for a ULA, the channel covariance matrices, which have Toeplitz structure, are asymptotically equivalent to corresponding circulant matrices [12]. That is, if \( F \) denotes the unitary DFT matrix, we have the asymptotic equivalence

\[
C_\delta \sim A F^H \text{diag}(c_\delta)F \quad \forall \delta
\]

where \( c_\delta \) contains the diagonal elements of \( FC_\delta F^H \). As a consequence, we have a corresponding asymptotic equivalence for the MMSE filters

\[
W_\delta \sim A F^H \text{diag}(w_\delta)F \quad \forall \delta
\]

where \( w_\delta \) contains the diagonal elements of \( FW_\delta F^H \). For a large-scale system, we can assume that the circulant approximation is exact and use Assumption [11] with \( Q = F \). We call the estimator that uses Assumption [11] with \( Q = F \) the circulant MMSE estimator. Given the element-wise MMSE filter \( \hat{w}(\hat{c}) \), we can, thus, calculate the MMSE estimates of the channel vectors with a complexity of \( \mathcal{O}(M \log M) \).

To reduce the approximation error for finite numbers of antennas we can use a more general factorization with \( Q = F_2 \), where \( F_2 \) contains the first \( M \) columns of an \( 2M \times 2M \) DFT matrix. The class of matrices that can be expressed as

\[
W_\delta = F_2^H \text{diag}(w_\delta)F_2
\]

are exactly the Toeplitz matrices. The estimator that uses Assumption [11] with \( Q = F_2 \) is denoted as the Toeplitz MMSE estimator. The complexity for given \( \hat{w}(\hat{c}) \) is still \( \mathcal{O}(M \log M) \).

An analogous result can be derived for uniform rectangular arrays. In this case the transformation \( Q = F^T \otimes F \) is the...
Algorithm 1 Reduced-Complexity MMSE Filter

1: Calculate the covariance matrix $C_{\delta_0}$ based on the hyperparameter $\delta_0$
2: Calculate the filter $W_{\delta_0} = C_{\delta_0}(\sigma^2 I + C_{\delta_0})^{-1}$
3: Calculate the corresponding low-dimensional parameter vector $u_{\delta_0}$ such that $W_{\delta_0} = Q^H \text{diag}(u_{\delta_0})Q$
4: Given $c$ from (43) use $A = F^H \text{diag}(F u_{\delta_0})F$ to get the approximately optimal filter
   \[ \hat{w}(c) = A \exp \left( \frac{T}{\sigma} A^T c + b \right) \]
   which is then used to calculate the channel estimates
   \[ \hat{h}[t] = Q^H \text{diag}(\hat{w}(c))Qy[t] \]

Kronecker product of two DFT matrices. The sizes of the DFT matrices correspond to the number of antennas in both directions of the array.

A third example with a decomposition as in Assumption 1 are distributed antennas [16, 17]. For distributed antennas, the covariance matrices are typically modelled as diagonal matrices and, thus, the filters $W_{\delta}$ are also diagonal. That is, for distributed antennas we simply have $Q = I$.

B. Step Two: A Fast MMSE Estimator

The main complexity in the calculation of $\hat{w}$ are the matrix vector-products with $A$ and $A^T$. For a dense matrix $A$, the computation of $\hat{w}(c)$ needs $O(KN)$ operations. To achieve a complexity of the same order as the ML estimator in (20), we must find a representation for $A$ that allows for fast matrix-vector products. One assumption that allows us to find such a representation is the following.

Assumption 2. We have a grid $\delta_i$, $i = 0, \ldots, K - 1$, which accurately represents the prior $p(\delta)$ and for which the diagonal filters
   \[ u_{\delta_i} = S_i u_{\delta_0} \] (44)
can be expressed in terms of a generating vector $u_{\delta_0}$ and the cyclic shift operator $S_i$ with shift $i$.

Given Assumption 2, the matrix $A$ in (43) is also a circulant matrix generated by the vector $u_{\delta_0}$. Thus, $A$ has the eigenvalues $F u_{\delta_0}$ and can be represented as
   \[ A = F^H \text{diag}(F u_{\delta_0})F \in \mathbb{R}^{K \times K} \] (45)
where $F$ is the $K$-dimensional DFT matrix. Therefore, the computational complexity of evaluating $\hat{w}(c)$ reduces to $O(K \log K)$, that is, $O(M \log M)$ as of Assumption 1. The complete low-complexity algorithm, which exploits both assumptions, is described in Alg. 1 and we will refer to this algorithm as fast MMSE filter.

Example. One example that approximately fulfills the shift invariance in Assumption 2 is the 3GPP spatial channel model with only a single propagation path. In this case, we only have one parameter for the covariance matrix: the angle of the path center $\delta$ which is uniformly distributed. The power density function of the angle of arrival is given by
   \[ g_\theta(\theta; \delta) = \exp(-d_{2\pi}(\delta, \theta)/\sigma_{AS}) \] (46)
where $d_{2\pi}(\theta, \delta)$ is the wrap-around distance between $\theta$ and $\delta$ and can be thought of as $|\theta - \delta|$ for most $(\theta, \delta)$ pairs. In other words, for different $\delta$, the function $g_\theta(\theta; \delta)$ is simply a shifted version of $g_\theta(\theta; 0)$, i.e., $g_\theta(\theta; \delta) = g_\theta(\theta - \delta; 0)$.

For small spatial frequencies $\omega$, the spectrum $f_p(\omega; \delta)$ is approximately proportional to the power density $g_\theta(\omega; \delta)$. Thus, we have
   \[ f_p(\omega; \delta) \approx f_p(\omega - \delta; 0) \] (47)
and the approximation error increases with $\delta$. Since the vectors $c_\delta$ of the circulant approximations basically contain uniform samples of the spectrum $f_p(\omega; \delta)$, the vectors $c_\delta$ are also approximately shifted versions of $c_0$. The actual vectors $u_{\delta}$ for different $\delta$ are depicted in Fig. 4. We note that the shift invariance is lost for large angles, which correspond to peaks in the middle of the graph, due to the mildly non-linear transformation from $g$ to $f$. Nevertheless, for this simple channel model we can apply Assumption 2 with reasonable accuracy.

Note again that this approach only works because of the approximate shift invariance of the covariance matrices for the specific channel model with only a single propagation path. For more involved channel models, it is not straightforward to find a good circulant approximation of the matrix $A$. In the next section, we discuss how to improve the performance of low-complexity estimators with similar structure using a learning method.

V. LEARNING THE MMSE ESTIMATOR

In general, the fast MMSE estimator does not yield the MMSE filter since the covariance matrices are not exactly circular for finite $M$, and because the shift invariance does not hold for typical channel models. However, it may still be worthwhile to consider the class of low-complexity estimators that are of the same form, i.e., the set of all estimators that are nonlinear functions of the form
   \[ \hat{h}[t] = Q^H \text{diag}(\hat{w}(c))Qy[t] \] (48)
Algorithm 2 Learned fast MMSE filter

1: Generate/select a mini-batch of $S$ channel vectors $H_s$ and corresponding observations $Y_s$, for $s = 1, \ldots, S$
2: Calculate the stochastic gradient with respect to the model parameters (in this case $w_0$ and $b$)
\[
g = \frac{1}{S} \sum_{s=1}^{S} \frac{\partial}{\partial w_0} b || Q^H \text{diag}(\hat{w}(\hat{c}(Y_s)))QY_s - H_s ||^2_F
\]
with $\hat{w}(x)$ as stated in Alg. 1
3: Use your favorite gradient based algorithm to update the parameters (e.g. [19])

with $\hat{c} = T^{-1} \sum_{t=1}^{T} ||Qy||^2$ and where $\hat{w}$ is a member of the class of functions
\[
\mathcal{W} = \left\{ x \mapsto \begin{pmatrix} \exp(x^T A^T x + b) \\ T^T \exp(x^T A^T x + b) \end{pmatrix}, A \in \mathcal{C}, b \in \mathbb{R}^M \right\}
\]
with $\mathcal{C} \in \mathbb{R}^{K \times K}$ denoting the set of all circulant Toeplitz matrices.

We want to find the function $w \in \mathcal{W}$ that minimizes the $\text{MSE}$. That is, we want to solve
\[
\min_{w \in \mathcal{W}} \mathbb{E} \left[ ||Q^H \text{diag}(\hat{w}(\hat{c}))QY - H||^2_F \right]
\]
where $H = [h[1], \ldots, h[T]]$. We know from the previous section that the class of functions $\mathcal{W}$ contains the MMSE estimator if Assumptions 1 and 2 hold, but in general, we will not get the MMSE estimator.

Now, we replace the expectation operator by a Monte Carlo estimate, i.e., we solve
\[
\min_{w \in \mathcal{W}} \frac{1}{P} \sum_{p=1}^{P} ||Q^H \text{diag}(\hat{w}(\hat{c}(Y_p)))QY_p - H_p||^2_F
\]
which has the advantage that the true distribution of $h$ does not need to be known; we only need samples $(H_p, Y_p)$. In fact, we could also take a sample of channel vectors and observations from a measurement campaign to learn the MMSE estimator for the “true” channel model. The resulting algorithm, which we will refer to as learned fast MMSE filter, learns the structured MMSE estimator with $\hat{w}$ in the class of functions $\mathcal{W}$ and is described in Alg. 2. The stochastic gradient algorithm is applied to mini-batches of the cost function (50), i.e., at each step, the sum is only over a subset of all indices. This is the typical approach to train neural networks [18]. The size of the mini-batches is usually in the order of tens up to a few hundreds of samples.

A. Neural Network Architecture

For realistic channel models, the learned fast MMSE filter is clearly sub-optimal. In simulations, we observe a significant gap to the Toeplitz MMSE filter when the number of grid points $N$ is chosen large enough. Our goal is to find a more general class of functions with the same order of complexity and in which we can find a better low-complexity MMSE filter. We still use Assumption 1, i.e., we still have a structured estimator, but we use a class of functions that has better generalization properties than $\mathcal{W}$ to find $\hat{w}$.

Functions in the class $\mathcal{W}$ can be visualized as the graph shown in Fig. 2. We note that the functions can be represented by a typical neural network architecture. In the context of neural networks, we identify two convolutional layers represented by the circulant matrices $A$ and $A^T$, and the non-linear, element-wise softmax activation function. In contrast to a typical neural network, the convolutions use the same parameters. The softmax function is also not commonly used as an activation function.

With a general activation function and separate parameters for the different convolutional layers, we obtain the network depicted in Fig. 3. This architecture can be generalized to a higher number of layers and a higher number of convolutions per layer. We refer to the resulting estimator as a structured neural network estimator. Since we focus on the Toeplitz structure in this work we show results for the Toeplitz neural network estimator in our simulations.

For each layer $\ell = 1, \ldots, L$ of the neural network, the input vector $x^{(\ell)}$ is mapped to
\[
x^{(\ell+1)} = \phi \left( A^{(\ell)} x^{(\ell)} + b^{(\ell)} \right)
\]
where the $A^{(\ell)}$ are (block-)circulant and $\phi$ is an activation function. For the first layer we have the input $x^{(1)} = \hat{c}$. The other layers use the output of the previous layer as input. The output of the last layer is used as the spatial filter $\hat{w} = x^{(L+1)}$.

In our simulations we use rectified linear units $\phi(x) = [x]_+$ as activation functions, since they were found to be easier to train than other activation functions [20]. As we use only convolution matrices and simple activation functions, the overall complexity of the algorithm is still $O(M \log M)$, but the higher flexibility compared to the more restrictive class of functions in $\mathcal{W}$ allows us to reach the performance of the more complex Toeplitz MMSE filter.

B. Hierarchical Training

Local optima are a major issue when training the neural networks, i.e., when calculating a solution of the nonlinear optimization problem (51). During our experiments, we observed
of the function assumption (cf. (47)), the entries of the vector $w$ of antennas and then obtain a starting value for the parameters. W e start with a small number of antennas and then increases the number of antennas step-by-step. For the low-complexity estimator with the shift-invariance assumption (cf. (47)), the entries $w^0$ of the generating vector $w$ are obtained by interpolation from the corresponding vector of a system with less antennas by typical interpolation methods.

Now, we assume that this interpolation property also holds if we use learning to adapt $w^0$. That is, we learn the generating vector $w^0$ of the convolution matrix $A$ for a smaller number of antennas and then obtain a starting value for $w^0$ for the desired number of antennas by interpolation. We then add a few additional training steps to refine the parameters for the larger number of antennas.

We apply the same approach to the neural networks introduced in Section V-A. We start with a small number of antennas which is increased iteratively until we reach the desired number. After each increase of $M$, the parameters complexity per iteration due to the reduced number of antennas.

Fig. 4. Box-plot of the MSE after training for 4000 iterations for hierarchical and non-hierarchical training. We show results for $M = 64$ and $M = 128$ antennas. Scenario with three propagation paths, $\sigma^2 = 1$, $T = 1$.

**Algorithm 3 Hierarchical Training**

1. Choose $\beta \in (0, 1)$, e.g., $\beta = 0.5$
2. for $i$ from 1 to $n$ do
   3. set $M^{(i)} = \lceil M^{\beta^{i-1}} \rceil$
   4. use some interpolation method (e.g. linear interpolation) to increase the number of entries of the interpolation kernels of all $A^{(i)}$ and the bias vectors $b^{(i)}$ from $K^{(i-1)}$ to $K^{(i)}$
   5. Normalize the $A^{(i)}$ by multiplying with $\beta$
   6. Train the network for a fixed number iterations with a gradient based algorithm (cf. Alg. 2)
3. end for

that local optimal can become an issue when the number of antennas is large. To deal with this problem, we devise a hierarchical training procedure that starts the training with a small number of antennas and then increases the number of antennas step-by-step.

For the low-complexity estimator with the shift-invariance assumption (cf. (47)), the entries of the vector $w^0$ are samples of the function

$$w(\omega) = \frac{f(\omega)}{f(\omega) + \sigma^2}. \tag{53}$$

If we assume that $f(\omega)$ is a smooth function, we can quite accurately calculate the generating vector $w^0$ for a system with $M$ antennas from the corresponding vector of a system with less antennas by typical interpolation methods.

Now, we assume that this interpolation property also holds if we use learning to adapt $w^0$. That is, we learn the generating vector $w^0$ of the convolution matrix $A$ for a smaller number of antennas and then obtain a starting value for $w^0$ for the desired number of antennas by interpolation. We then add a few additional training steps to refine the parameters for the larger number of antennas.

We apply the same approach to the neural networks introduced in Section V-A. We start with a small number of antennas which is increased iteratively until we reach the desired number. After each increase of $M$, the parameters

of the network, the kernels of the convolutions $A^{(i)}$ and the bias vectors $b^{(i)}$, are obtained by interpolation from the previously trained smaller network. Since we use rectified linear units as activation functions, we normalize the kernels of the convolution such that we get approximately similar values at the outputs of each layer. The method is described step-by-step in Alg. 3. This approach significantly improves convergence speed of the training while also lowering computational complexity per iteration due to the reduced number of antennas in most training steps.

In fact, for a larger number of antennas the hierarchical training is essential to obtain good performance. In Fig. 4 we show a box-plot of the MSE after training with 4000 iterations. As we can see, without the hierarchical training, the estimators get stuck in local optima and only the occasional outlier converges to an estimator with close to optimal performance. With the hierarchical approach, we are less likely to be caught in local optima during the training process. At the same time the complexity is reduced.

**VI. SIMULATIONS**

We first analyze the accuracy of the approximations that lead to the low-complexity estimators. We begin with the single snapshot, single-path channel model that motivated the shift invariance, i.e., Assumption 2. The per-antenna MSE of the channel estimation for the different approximations is depicted in Fig. 3 as a function of the number of antennas $M$ for a fixed SNR of 0 dB. We use the power density in (46) with an angular spread of $\sigma_{AS} = 2^\circ$.

As a baseline, we show the MSE for the genie-aided MMSE estimator, i.e., the MMSE estimator for known parameters $\delta$. We see a significant gap in performance of the circulant MMSE estimator compared to the MMSE estimator, which, however, closes for larger numbers of antennas. As expected, the Toeplitz MMSE estimator is closer to the MMSE estimator in terms of performance, which is why we focus on the Toeplitz structure in the following.

For this scenario, the fast MMSE estimator that uses Assumptions 1 and 2 yields performance close to the circulant MMSE estimator. This is
because the assumption of shift invariance of the power density function is reasonably accurate.

In the following simulations, we compare the neural network estimator with the MMSE estimators and the ML estimator introduced in Sec. III-A. The neural network estimator is implemented as shown in Fig. 3 with two square Toeplitz matrices $A^{(1)}$ and $A^{(2)}$ and two real vectors $b^{(1)}$ and $b^{(2)}$ and the rectified linear activation function $\phi(x) = [x]_+$. For comparison purposes, we also show results for the orthogonal matching pursuit (OMP) algorithm [21], which uses the multiple measurement model $H = AX$ with a four-times oversampled DFT matrix $A$ and a row-sparse matrix $X$, i.e., each channel realization is approximated as a linear combination of $k$ DFT vectors. Because the selection of the optimal sparsity level $k$ is non-trivial, we use a genie-aided approach in which the OMP algorithm uses the actual channel realizations $H$ to decide about the optimal value for $k$ that maximizes the metric of interest. The result is an upper bound for the performance of the OMP algorithm.

In Figs. 6 and 7 we show results for a channel model with more underlying parameters. We use the 3GPP model with three propagation paths which have different relative path gains. That is, the power density is given by

$$g_p(\theta, \delta) = [\delta_1, \delta_2, \delta_3, p_1, p_2, p_3]^T = \sum_{i=1}^{3} p_i g_p(\theta, \delta_i)$$  \hspace{1cm} (54)$$

where the angles $\delta_i$ are uniformly distributed. The path gains are also drawn from a uniform distribution, but then normalized such that $\sum_i p_i = 1$. The per path angular spread is still $\sigma_{AS} = 2^\circ$.

This increases the number of free parameters from one to five. As a consequence, the MMSE estimator becomes intractable due to the large number of required grid points. The Toeplitz MMSE estimator, which exploits the structure of the covariance matrices, still needs to sample the vectors $w_\delta$ for different parameters and is, thus, also not suitable for this scenario. Nevertheless, we depict the performance of the Toeplitz MMSE estimator with $N = M^2$ grid points. That is, the computational complexity is $O(M^3)$. We further show results for the fast MMSE filter described in Alg. 1 with $w_0$ from the single path model as well as for the learned fast MMSE filter in Alg. 2.

In the multiple-path scenario the Toeplitz neural network (NN) estimator introduced in Section V-A significantly outperforms the other $O(M \log M)$ methods. Specifically, we note the gap to the learned fast MMSE filter which is trained with the same amount of data. We ran the training procedure for 8000 iterations with mini-batches of 20 samples generated from the channel model. For high SNR the fast MMSE estimator can be worse than the least squares estimators while the more learning based methods always outperform the least squares estimator.

Finally, we use the urban-macro channel model as specified
is given as
\[
p(Y|\delta) = \prod_{t=1}^{T} \exp \left( \frac{-y[t]^H(C_\delta + C_z)^{-1}y[t]}{\pi^M|C_\delta + C_z|} \right)
\]
\[
= \exp \left( -T \text{tr}([C_\delta + C_z]^{-1}\hat{C}) \right)
\]
\[
= \exp \left( -T \text{tr}([C_\delta + C_z]^{-1}\hat{C}) - T \log |C_\delta + C_z| \right)
\]
(58)

We express \(C_\delta + C_z\) in terms of \(W_\delta\) as follows: First, we have
\[
C_\delta = W_\delta(C_\delta + C_z) \quad \Leftrightarrow \quad C_\delta = (I-W_\delta)^{-1}W_\delta C_z.
\]
(59)

We can then write
\[
(C_\delta + C_z)^{-1} = ((I-W_\delta)^{-1}(W_\delta C_z + (I-W_\delta) C_z))^{-1}
\]
\[
= ((I-W_\delta)^{-1}C_z)^{-1}
\]
\[
= C_z^{-1}(I-W_\delta)
\]
(60)
(61)
(62)

so that
\[
\frac{p(Y|\delta)}{E_\delta[p(Y|\delta)]} = \exp \left( -T \text{tr}([C_z^{-1}(I-W_\delta)\hat{C}] - T \log |(I-W_\delta)^{-1}C_z|) \right)
\]
\[
E_\delta \left[ \exp \left( -T \text{tr}([C_z^{-1}(I-W_\delta)\hat{C}] - T \log |(I-W_\delta)^{-1}C_z|) \right) \right]
\]
\[
= \exp \left( T \text{tr}([C_z^{-1}W_\delta\hat{C}] + T \log |I-W_\delta|) \right)
\]
\[
E_\delta \left[ \exp \left( T \text{tr}([C_z^{-1}W_\delta\hat{C}] + T \log |I-W_\delta|) \right) \right]
\]
(63)
(64)
(65)

If we substitute \(C_z = \sigma^2 I\), Lemma [1] follows.

\section*{REFERENCES}

[1] F. Rusek, D. Persson, B. K. Lau, E. Larsson, T. Marzetta, O. Edfors, and F. Tufvesson, “Scaling Up MIMO: Opportunities and Challenges with Very Large Arrays,” IEEE Signal Process. Mag., vol. 30, pp. 40–60, Jan. 2013.

[2] T. Marzetta, “Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas,” IEEE Trans. Wireless Commun., vol. 9, pp. 3590–3600, Nov. 2010.

[3] 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, pp. 264–273, Feb. 2013.

[4] D. C. Araujo, A. L. F. de Almeida, J. A. Xasnas, and J. C. M. Mota, “Channel estimation for millimeter-wave very-large MIMO systems,” in Proceedings of the 22nd European Signal Processing Conference (EUSIPCO), Sep. 2014, pp. 81–85.

[5] 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.

[6] 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.

[7] T. S. Rappaport, F. Gutierrez, E. Ben-Dor, J. N. Murdock, Y. Qiao, and J. I. Tamir, “Broadband millimeter-wave propagation measurements and models using adaptive-beam antennas for outdoor urban cellular communications,” IEEE Trans. Antennas Propag., vol. 61, no. 4, pp. 1850–1859, Apr. 2013.

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

\section*{APPENDIX}

\subsection*{A. Proof of Lemma [1]}

We show Lemma [1] for the slightly more general case with arbitrary noise covariance matrices \(C_z\). The posterior distribution of \(y\) in
\[
h_{\text{MMSE}} = \frac{E_\delta[p(Y|\delta)W_\delta]}{E_\delta[p(Y|\delta)]} y
\]
(55)
[9] 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.

[10] V. Solov’ev, “A minimax-Bayes estimate on classes of distributions with bounded second moments,” Russian Mathematical Surveys, vol. 50, p. 832, 1995.

[11] ——, “On the problem of minimax-Bayesian estimation,” Russian Mathematical Surveys, vol. 53, p. 1104, 1998.

[12] R. M. Gray, Toeplitz and circulant matrices: A review. now publishers inc, 2006.

[13] D. Neumann, M. Joham, L. Weiland, and W. Utschick, “Low-Complexity Computation of LMMSE Channel Estimates in Massive MIMO,” in Workshop on Smart Antennas, Mar. 2015, pp. 1–6.

[14] J. P. Burg, D. G. Luenberger, and D. L. Wenger, “Estimation of structured covariance matrices,” Proceedings of the IEEE, vol. 70, pp. 963–974, Sep. 1982.

[15] T. W. Anderson, “Asymptotically Efficient Estimation of Covariance Matrices with Linear Structure,” The Annals of Statistics, vol. 1, pp. 135–141, 1973.

[16] H. Q. Ngo, A. Ashikhmin, H. Yang, E. Larsson, and T. Marzetta, “Cell-Free Massive MIMO: Uniformly great service for everyone,” in Int. Workshop Signal Process. Adv. Wireless Commun., Jun. 2015, pp. 201–205.

[17] 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.

[18] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016, http://www.deeplearningbook.org.

[19] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” CoRR, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[20] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in Proceedings of the 14 th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 2011, pp. 315–323.

[21] M. Gharavi-Alkhansari and T. S. Huang, “A fast orthogonal matching pursuit algorithm,” in Acoustics, Speech and Signal Processing. Proceedings of the 1998 IEEE International Conference on, vol. 3, May 1998, pp. 1389–1392.