Learning-aided joint time-frequency channel estimation for 5G new radio

Myers, Nitin Jonathan ; Kwon, Hyukjoon ; Ding, Yacong ; Song, Kee-Bong

DOI
10.1109/GLOBECOM46510.2021.9685651

Publication date
2021

Document Version
Final published version

Published in
Proceedings of the IEEE Global Communications Conference (GLOBECOM 2021)

Learning-aided joint time-frequency channel estimation for 5G new radio

Nitin Jonathan Myers*, Hyukjoon Kwon†, Yacong Ding†, and Kee-Bong Song†
*Delft Center for Systems and Control, Delft University of Technology, Netherlands
†Samsung Semiconductor Inc., San Diego, USA
Email: N.J.Myers@tudelft.nl, {hyukjoon.k, yacong.ding, keebong.s}@samsung.com

GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9685651

***Abstract*—In this paper, we propose a learning-aided signal processing solution for channel estimation in 5G new radio (NR). Channel estimation is an important algorithm for baseband modem design. In 5G NR, estimating the channel is challenging due to two reasons. First, the pilot signals are transmitted over a small fraction of the available time-frequency resources. Second, the real time nature of physical layer processing introduces a strict limitation on the computational complexity of channel estimation. To this end, we propose a channel estimation technique that integrates a small one hidden layer neural network between two linear minimum mean squared error (LMMSE) interpolation blocks. While the neural network leverages the advantages of offline data-driven learning, the LMMSE blocks exploit the second order online channel statistics along time and frequency dimensions. The training procedure tunes the weights of the neural network by back-propagating through the time domain LMMSE interpolation block. We derive bounds on the training loss with the proposed method and show that our approach can improve the channel estimate.**

***Index Terms*—5G NR, deep learning, channel estimation.**

I. Introduction

Modem baseband algorithms need to be carefully designed in 5G new radio (NR) to realize physical layer operations at a reduced complexity [1]. Channel estimation is an important operation, as the estimated channel is used for subsequent operations like equalization and interference whitening. Due to the use of a large bandwidth in 5G NR, the number of subcarriers is large and accordingly the frequency domain channel has a high dimension. Estimating the channel is challenging because the pilot signals, called demodulation reference signals (DMRS), are only transmitted over a fraction of subcarriers within the bandwidth [2]. In this case, the high dimensional channel must be estimated from its subsampled version in an orthogonal frequency division multiplexing (OFDM) symbol duration. The channel estimation problem is also complicated along the time dimension because DMRS are transmitted in only a few OFDM symbols within each slot.

Prior work has investigated channel estimation from a subsampled version of the time-frequency channel representation. This includes linear or spline-based interpolation using the channel estimates at the DMRS to predict the channel at the other time-frequency locations [3]. An alternative approach exploits the second order channel statistics along the frequency and the time dimensions, in order to perform 2D-linear minimum mean squared error (LMMSE) interpolation [4]. Such second order statistics can be estimated from reference signals like the tracking reference signal (TRS) or the synchronization signal block (SSB) which are transmitted periodically [5]. The 2D-LMMSE interpolation method has a high complexity, which can be circumvented under the wide sense stationary uncorrelated scattering (WSSUS) assumption on the channel [4]. Under the WSSUS assumption, 2D-LMMSE interpolation is equivalent to successively performing 1D-LMMSE interpolations along the frequency and the time dimensions, respectively [6].

Data-driven techniques based on neural networks can learn the underlying statistics of the input distribution. These methods have been applied in communications to perform joint time-frequency channel estimation. For instance, 2D convolutional neural network (CNN)-based channel refinement techniques were proposed in [7] and [8]. A learning-aided least squares technique was developed in [9]. Algorithms based on recurrent neural network and ResNet were developed in [10] and [11]. Most of the learning-based algorithms in prior work, however, do not exploit the online channel statistics available from TRS or SSB. A naive approach to solve this problem is to concatenate the noisy channel features at the input of the neural network with TRS or SSB measurements. Adding additional input features from TRS or SSB increases the dimensions of the input feature vector, that results in a higher computational complexity at inference.

In this paper, we develop a network architecture that integrates a single hidden layer neural network between the two 1D-LMMSE blocks that perform frequency and time domain interpolations. The interpolation matrices in the 1D-LMMSE blocks dynamically vary with the slot, and these matrices are determined from TRS measurements. Our training procedure accounts for this variation by back-propagating through the 1D-LMMSE block. The main contributions of this paper are:

1) We develop a low complexity network architecture that exploits both the offline channel statistics and the online channel statistics derived from TRS.
2) We derive an upper bound on the training loss with our network. The bound depends on the mean squared error (MSE) at the DMRS locations and the error due to 1D-LMMSE-based time interpolation.
3) We use simulations to show that our algorithm can achieve a gain against the baseline LMMSE interpolation-based method for a moderate increase in the computational complexity.

978-1-7281-8104-2/21/$31.00 ©2021 IEEE

II. Preliminaries

We consider an NR system in the downlink where information is transmitted over time-frequency resources called resource elements (REs). Each RE corresponds to a single subcarrier and a single OFDM symbol. A collection of N_{sym} OFDM symbols constitute a slot of which N_{ctrl} symbols are configured for the control. A group of N_{sc} successive subcarriers in a slot with the same precoding is called a resource block (RB). Strictly speaking, in NR, an RB is just defined as group of $N_{\text{sc}} = 12$ consecutive subcarriers. In this paper, we assume that each RB has $N_{\text{sc}} \times N_{\text{sym}}$ REs similar to the Long-Term Evolution (LTE) standard, as shown in Fig. 1. We consider three types of REs for physical downlink control channel (PDCCH), physical downlink shared channel (PDSCH) and DMRS. The positions of DMRS REs depend on the DMRS configuration which can be tuned in NR. We use N_{dmrs} to denote the number of DMRS symbols in a slot, and $N_{\text{re,dmrs}}$ as the number of DMRS REs at a DMRS symbol in an RB. While pilot signals are sent over the DMRS, data is sent over PDSCH REs. This data can be detected and decoded only after estimating the channel at PDSCH REs. To this end, the channel at DMRS REs is first estimated using the received channel measurements at DMRS REs. Then, an interpolation technique is used to find the channel at PDSCH REs using the channel estimates at DMRS REs.

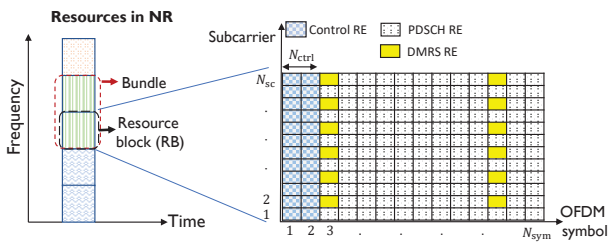

Fig. 1. The figure illustrates the concept of an RE and a RB. A bundle is a group of RBs with the same precoding. Here, $N_{\text{sym}} = 14$, $N_{\text{ctrl}} = 2$,$N_{\text{sc}} = 12$, $N_{\text{dmrs}} = 2$, $N_{\text{re,dmrs}} = 6$ and bundle size$= 2$.

We consider a narrowband precoding scenario where the precoding can vary across different bundles as shown in Fig. 1. In this paper, we focus on estimating the channel independently at every RB, i.e., the channel at PDSCH REs in an RB is estimated using DMRS REs within the same RB. This approach of channel estimation at the RB level can also be used for different bundle sizes, e.g., $1, 2$ or 4. The 2D-LMMSE-based interpolation technique is a common approach to estimate the channel at PDSCH REs [4]. Under the WSSUS assumption on the channel, 2D-LMMSE-based interpolation is equivalent to first performing frequency domain interpolation at all the DMRS locations, and then performing time domain interpolation at all the subcarriers [6]. A summary of this procedure is shown in Fig. 2.

We now discuss an overview of 1D-LMMSE interpolation along the frequency dimension (FD). We use $\mathbf{h}_{k,\text{dmrs}} \in \mathbb{C}^{N_{\text{re,dmrs}}}$ to represent the channel at the k^{th} DMRS symbol in a slot. For the DMRS configuration in Fig. 1, $k \in \{1, 2\}$. The channel measurements obtained after despreading at these DMRS REs is given by

$$\mathbf{r}_{k,\text{dmrs}} = \mathbf{h}_{k,\text{dmrs}} + \mathbf{v}_k, \tag{1}$$

where $\mathbf{v}_k$ is additive white Gaussian noise. Using the measurements in (1), FD interpolation is first performed to compute the channel $\hat{\mathbf{h}}_k$ at the k^{th} DMRS symbol. We denote the $N_{\text{sc}} \times N_{\text{re,dmrs}}$ FD interpolation matrix as $\mathbf{P}_{\text{FD}}$ to write

$$\hat{\mathbf{h}}_k = \mathbf{P}_{\text{FD}} \mathbf{r}_{k,\text{dmrs}}. \tag{2}$$

The coefficients in $\mathbf{P}_{\text{FD}}$ are derived using an LMMSE estimator that incorporates the correlation in the frequency domain channel, which is computed from the power delay profile (PDP) [12]. For the exact details on the interpolation coefficients used in this paper, we refer the interested reader to [6]. After FD interpolation is performed at every DMRS symbol, the $N_{\text{sc}} \times N_{\text{dmrs}}$ matrix

$$\hat{\mathbf{H}}_{\text{FDI}} = [\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2, \cdots, \hat{\mathbf{h}}_{N_{\text{dmrs}}}] \tag{3}$$

is constructed as the output of the FD interpolation block.

Now, $\hat{\mathbf{H}}_{\text{FDI}}$ is used to perform interpolation along the time dimension (TD), i.e., the N_{dmrs} symbols in an RB are used to estimate the channel at the $N_{\text{sym}} - N_{\text{ctrl}}$ symbols. We use $\mathbf{P}_{\text{TD}}$ to denote the $(N_{\text{sym}} - N_{\text{ctrl}}) \times N_{\text{dmrs}}$ TD interpolation matrix. The channel estimate over the RB of interest is then

$$\hat{\mathbf{H}}_{\text{intp}} = \hat{\mathbf{H}}_{\text{FDI}} \mathbf{P}_{\text{TD}}^T. \tag{4}$$

Similar to FD interpolation, the coefficients in TD interpolation are computed using an LMMSE estimator which accounts for the temporal correlation. These coefficients are obtained using the Jakes Doppler spectrum with the Doppler spread estimate from TRS [5]. In summary, LMMSE-based time-frequency interpolation exploits the second order channel statistics through the PDP and the Doppler spectrum, for channel estimation. The LMMSE-based estimator, however, is constrained to be linear by definition and it may not fully exploit the structure for channel estimation. In this paper, we develop a technique that leverages both the TD and FD interpolation blocks together with one hidden layer neural network for better channel estimation.

III. Baseline algorithms for learning-based channel estimation

In this section, we first describe the one hidden layer network which will be used within learning-based channel estimation. Then, we discuss algorithms that will be used to benchmark our technique proposed later in Section. IV.

A. One hidden layer network

The single hidden layer network models a function that maps an $N_{\text{in}} \times 1$ real input $\mathbf{x}$ to an $N_{\text{out}} \times 1$ real output $\mathbf{y}$. We consider a hidden layer with N_{hid} nodes and assume a ReLU activation at this layer. We use $\mathbf{W}_k$ and $\mathbf{b}_k$ to denote the weights and biases of the k^{th} layer. We assume a linear activation at the output to write

$$\mathbf{y} = \mathbf{W}_2 \text{ReLU}(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2. \tag{5}$$

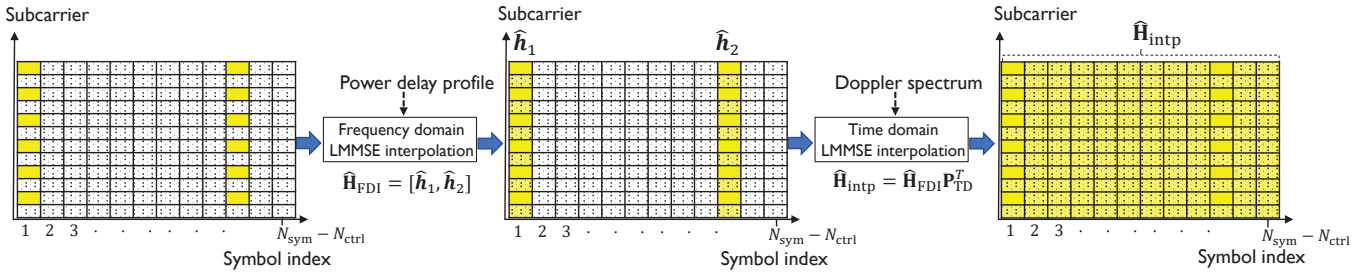

Fig. 2. In the LMMSE-based approach, frequency domain interpolation is first performed at the DMRS symbols using the PDP. Then, time domain interpolation is performed at each subcarrier using the Doppler spectrum. Here, an RE is shaded after the appropriate interpolation.

In a compact form, we rewrite (5) as $\mathbf{y} = \text{MLP}_{\boldsymbol{\xi}}(\mathbf{x})$, where $\boldsymbol{\xi}$ is a vector containing all the weights and biases. The parameters in $\boldsymbol{\xi}$ are tuned using a training dataset $\{\mathbf{x}^{(s)}, \mathbf{y}^{(s)}\}_{s=1}^{N_{\text{smp}}}$. The computational complexity of the network at inference is $\mathcal{O}(N_{\text{hid}}(N_{\text{in}} + N_{\text{out}}))$.

B. Learning-based channel estimation after interpolation

A trivial approach for learning-based channel estimation is to refine the LMMSE interpolation-based estimate in (4) using a neural network. The input feature of the network is then

$$\mathbf{x} = \begin{bmatrix} \text{vec}(\text{Re}\{\hat{\mathbf{H}}_{\text{intp}}\}) \\ \text{vec}(\text{Im}\{\hat{\mathbf{H}}_{\text{intp}}\}) \end{bmatrix}, \tag{6}$$

where $\text{Re}\{\cdot\}$ and $\text{Im}\{\cdot\}$ extract the real and imaginary parts of a matrix, and $\text{vec}(\cdot)$ reshapes the matrix into a vector. The network is trained with the ideal channel corresponding to $\hat{\mathbf{H}}_{\text{intp}}$, defined as $\mathbf{H}_{\text{ideal}}$. Similar to $\hat{\mathbf{H}}_{\text{intp}}$, $\mathbf{H}_{\text{ideal}}$ is an $N_{\text{sc}} \times (N_{\text{sym}} - N_{\text{ctrl}})$ matrix defined over an RB. The output feature vector used for training is then

$$\mathbf{y} = \begin{bmatrix} \text{vec}(\text{Re}\{\mathbf{H}_{\text{ideal}}\}) \\ \text{vec}(\text{Im}\{\mathbf{H}_{\text{ideal}}\}) \end{bmatrix}. \tag{7}$$

In this method, $N_{\text{in}} = N_{\text{out}} = 2(N_{\text{sym}} - N_{\text{ctrl}})N_{\text{sc}}$. The pairs $\{(\hat{\mathbf{H}}_{\text{intp}}^{(s)}, \mathbf{H}_{\text{ideal}}^{(s)})\}_{s=1}^{N_{\text{samp}}}$ are extracted from a channel simulator, and the network is trained by minimizing the MSE loss

$$\mathcal{L}_{\text{intp,lrn}} = \frac{\sum_{s=1}^{N_{\text{samp}}} \|\mathbf{y}^{(s)} - \text{MLP}_{\boldsymbol{\xi}}(\mathbf{x}^{(s)})\|^2}{N_{\text{samp}}} \tag{8}$$

over $\boldsymbol{\xi}$. The trained parameters are denoted by $\boldsymbol{\xi}_{\text{intp,lrn}}$.

At inference, the input feature $\mathbf{x}$ is constructed from $\hat{\mathbf{H}}_{\text{intp}}$ using (6). Then, the output of the network $\hat{\mathbf{y}} = \text{MLP}_{\boldsymbol{\xi}_{\text{intp,lrn}}}(\mathbf{x})$ is computed. Finally, the channel is estimated as

$$\hat{\mathbf{H}}_{\text{intp,lrn}} = \text{unvec}\left[\hat{\mathbf{y}}(1 : N_{\text{out}}/2) + \text{j}\hat{\mathbf{y}}(N_{\text{out}}/2 + 1 : N_{\text{out}})\right], \tag{9}$$

where $\text{unvec}[\cdot]$ is the reverse of the $\text{vec}[\cdot]$ operation and $\text{j} = \sqrt{-1}$. Although this method can exploit channel structure beyond the LMMSE estimator, the neural network operates on the interpolated channel features over an entire RB as shown in Fig. 3a. Such an approach has a higher computational complexity at inference because the dimension of the input feature vector is larger than the underlying feature $\hat{\mathbf{H}}_{\text{FDI}}$.

C. Learning-based channel estimation at the DMRS locations

In this method, the neural network is applied immediately after FD interpolation, but before TD interpolation. As a result the network refines the channel only at the DMRS locations. The input feature vector for the neural network is $\mathbf{x} = [\text{vec}(\text{Re}\{\hat{\mathbf{H}}_{\text{FDI}}\}); \text{vec}(\text{Im}\{\hat{\mathbf{H}}_{\text{FDI}}\})]$. We use $\mathcal{I}_{\text{dmrs}} \subset \{1, 2, \cdots N_{\text{sym}} - N_{\text{ctrl}}\}$ to denote the indices that correspond to the DMRS locations. For instance, $\mathcal{I}_{\text{dmrs}} = \{1, 10\}$ for the DMRS configuration shown in Fig. 2. We define $\mathbf{H}_{\text{dmrs}}$ as the subsampled version of the ideal channel at the DMRS locations, i.e.,

$$\mathbf{H}_{\text{dmrs}} = \mathbf{H}_{\text{ideal}}(:, \mathcal{I}_{\text{dmrs}}). \tag{10}$$

The output feature vector to train the network is constructed from the ideal channel subsampled at the DMRS locations, i.e., $\mathbf{y} = [\text{vec}(\text{Re}\{\mathbf{H}_{\text{dmrs}}\}); \text{vec}(\text{Im}\{\mathbf{H}_{\text{dmrs}}\})]$. The network used in this method has $N_{\text{in}} = N_{\text{out}} = 2N_{\text{sc}}N_{\text{dmrs}}$ and a single hidden layer.

The network is trained using pairs of FD interpolated channels and ideal channels subsampled at the DMRS locations. We use $\boldsymbol{\xi}_{\text{FDI,lrn}}$ to denote the parameters obtained by training the network with the MSE loss defined as $\mathcal{L}_{\text{lrn,dmrs}}$. The expression for $\mathcal{L}_{\text{lrn,dmrs}}$ is same as that in (8), where $\mathbf{x}$ and $\mathbf{y}$ here correspond to the channel at the DMRS symbol locations. At inference, the input feature vector $\mathbf{x}$ is constructed from $\hat{\mathbf{H}}_{\text{FDI}}$. The network output $\hat{\mathbf{y}} = \text{MLP}_{\boldsymbol{\xi}_{\text{FDI,lrn}}}(\mathbf{x})$ is then reshaped to obtain a refined version of the FD interpolated channel denoted by $\hat{\mathbf{H}}_{\text{FDI,lrn}}$. The reshape operation is same as the one in (9). Finally, TD interpolation is performed with the refined FD interpolated channel to obtain

$$\hat{\mathbf{H}}_{\text{lrn,dmrs}} = \hat{\mathbf{H}}_{\text{FDI,lrn}}\mathbf{P}_{\text{TD}}^T \tag{11}$$

as the channel estimate over the RB. We used lrn, dmrs in (11) to indicate the estimate, as neural network-based refinement is performed by only minimizing the MSE loss at the DMRS symbol locations. This method has a lower complexity at inference when compared to the method in Sec. III-B due to the reduced feature dimensions. Unfortunately, this method does not account for the MSE at the symbols other than the N_{dmrs} DMRS symbols. Minimizing the MSE loss at other symbol locations is important as they contain several PDSCH REs on which data is transmitted.

IV. Proposed learning-aided channel estimation

In this section, we discuss the proposed channel estimation technique that minimizes the MSE loss at all the $N_{\text{sym}} - N_{\text{ctrl}}$ symbols within an RB, and has a low computational complexity when compared to the method in Sec. III-B. Our technique incorporates the network in Sec. III-C through a super-network. The super-network is a cascade of a single hidden layer network and an additional linear layer derived from TD interpolation.

A. Overview of the super-network architecture

The proposed architecture includes a single hidden layer neural network that operates on the features from the FD interpolated channel, i.e., the input to the network is

$$\mathbf{x} = \begin{bmatrix} \text{vec}(\text{Re}\{\hat{\mathbf{H}}_{\text{FDI}}\}) \\ \text{vec}(\text{Im}\{\hat{\mathbf{H}}_{\text{FDI}}\}) \end{bmatrix}. \tag{12}$$

The one hidden layer network outputs a refined version of the FD interpolated channel at the DMRS symbol locations defined as $\hat{\mathbf{y}}_{\text{fd}} = \text{MLP}_{\boldsymbol{\xi}}(\mathbf{x})$. Here, $\hat{\mathbf{y}}_{\text{fd}}$ is a $2N_{\text{sc}}N_{\text{dmrs}} \times 1$ vector that is considered as an intermediate output of a super-network shown in Fig. 3c. Within this super-network, $\hat{\mathbf{y}}_{\text{fd}}$ is sent into a linear layer that implements TD interpolation. The final output of the super network, denoted by $\hat{\mathbf{y}}$, contains the $2N_{\text{sc}}(N_{\text{sym}} - N_{\text{ctrl}})$ dimension channel estimate over the RB.

We would like to highlight that the network in Fig. 3b and the proposed super-network in Fig. 3c only differ by a single layer at the output. This layer realizes the LMMSE-based TD interpolation operation using the Doppler spectrum, and the weights of this layer are derived from (4) as

$$\mathbf{W}_3 = \begin{bmatrix} \text{Re}\{\mathbf{P}_{\text{TD}}\} \otimes \mathbf{I}_{N_{\text{sc}}} & -\text{Im}\{\mathbf{P}_{\text{TD}}\} \otimes \mathbf{I}_{N_{\text{sc}}} \\ \text{Im}\{\mathbf{P}_{\text{TD}}\} \otimes \mathbf{I}_{N_{\text{sc}}} & \text{Re}\{\mathbf{P}_{\text{TD}}\} \otimes \mathbf{I}_{N_{\text{sc}}} \end{bmatrix}, \tag{13}$$

where $\mathbf{I}_{N_{\text{sc}}}$ is an identity matrix of size $N_{\text{sc}} \times N_{\text{sc}}$, and $\otimes$ denotes the kronecker product. The bias of the output layer in the super network is set to zero to exactly mimic LMMSE-based TD interpolation. As a result, the output of the super-network is

$$\hat{\mathbf{y}} = \mathbf{W}_3 \hat{\mathbf{y}}_{\text{fd}}. \tag{14}$$

It is important to note that the weights of the output layer in the super-network are derived from the LMMSE estimator and these weights are not trained during back-propagation.

B. Training and inference with the super-network

We now explain the training procedure for the proposed super-network. Unlike the approach in Sec. III-C that minimizes the MSE loss only at the DMRS symbol locations for network training, the proposed approach minimizes the MSE loss at all the $N_{\text{sym}} - N_{\text{ctrl}}$ symbols in an RB. Such a minimization is possible due to the final layer in the super network that performs TD interpolation. The network is trained with the output feature $\mathbf{y}$ in (7) which contains the ideal channel over the RB. The loss function used for training is

$$\mathcal{L}_{\text{lrn,RB}} = \frac{\sum_{s=1}^{N_{\text{samp}}} \|\mathbf{y}^{(s)} - \mathbf{W}_3^{(s)} \text{MLP}_{\boldsymbol{\xi}}(\mathbf{x}^{(s)})\|^2}{N_{\text{samp}}}. \tag{15}$$

We would like to mention two important aspects of the loss function in (15). First, the TD interpolation weights depend on the Doppler spread estimate which varies over time. As a result, the matrix $\mathbf{W}_3^{(s)}$ that contains these weights changes with the training sample. Second, we note that although the final layer in the super-network containing $\mathbf{W}_3$ is not trained, the parameters of the preceeding layers in $\boldsymbol{\xi}$ are trained. In this case, the gradients with $\boldsymbol{\xi}$ are computed by back-propagating through the final layer in the super-network.

In the proposed method, the FD interpolated channel $\hat{\mathbf{H}}_{\text{FDI}}$, the ideal channel $\mathbf{H}_{\text{ideal}}$, and the TD interpolation matrix $\mathbf{P}_{\text{TD}}$ are dumped from our channel simulator. Then, this information is used to construct the training feature set $\{\mathbf{x}^{(s)}, \mathbf{y}^{(s)}, \mathbf{W}_3^{(s)}\}_{s=1}^{N_{\text{samp}}}$. Training the super-network is different from conventional training due to the fact that the weights of the final layer vary with the sample index. To this end, we use a batch size of 1 and load the non-trainable weights of the final layer for each sample. Such an approach is also used for validation and inference as $\mathbf{W}_3$ changes according to the Doppler spread estimate. At inference, the output of the super-network $\hat{\mathbf{y}}$ is reshaped into the channel estimate

$$\hat{\mathbf{H}}_{\text{lrn,RB}} = \text{unvec}\left[\hat{\mathbf{y}}(1 : N_{\text{out}}^{\text{sup}}/2) + \text{j}\hat{\mathbf{y}}(N_{\text{out}}/2 + 1 : N_{\text{out}}^{\text{sup}})\right], \tag{16}$$

where $N_{\text{out}}^{\text{sup}} = 2N_{\text{sc}}(N_{\text{sym}} - N_{\text{ctrl}})$ is the size of the output in the super-network.

C. Analysis on the training loss

A good initialization for the trainable weights in the super-network is the trained weights of the network in Sec. III-C, i.e., $\boldsymbol{\xi}^{(0)} = \boldsymbol{\xi}_{\text{FDI,lrn}}$. This is because $\boldsymbol{\xi}_{\text{FDI,lrn}}$ achieves good refinement of the FD interpolated channels. Under this initialization, we analyse the training loss of the super-network $\mathcal{L}_{\text{lrn,RB}}$ in terms of $\mathcal{L}_{\text{lrn,dmrs}}$.

To aid our analysis, we consider the super-network in Fig. 3c and simply use a matrix representation instead of the vector versions. We define $\hat{\mathbf{H}}_{\text{fd}}$ as the $N_{\text{sc}} \times N_{\text{dmrs}}$ complex matrix whose vector version is $\hat{\mathbf{y}}_{\text{fd}}$. Note that $\mathbf{H}_{\text{ideal}}$ is the matrix representation of $\mathbf{y}$. For a particular training sample, the error at the output of the super-network can be expressed as

$$\delta = \|\hat{\mathbf{H}}_{\text{fd}} \mathbf{P}_{\text{TD}}^T - \mathbf{H}_{\text{ideal}}\|_{\text{F}}. \tag{17}$$

Now, we define $\mathbf{E}$ as the error due to TD interpolation, i.e.,

$$\mathbf{E} = \mathbf{H}_{\text{ideal}} - \mathbf{H}_{\text{dmrs}} \mathbf{P}_{\text{TD}}^T. \tag{18}$$

The error $\mathbf{E}$ is inherent to TD interpolation and is independent of learning algorithm. We define σ_{TD} as the maximum singular value of $\mathbf{P}_{\text{TD}}$. Substituting $\mathbf{H}_{\text{ideal}}$ from (18) in (17), we get

$$\delta = \|(\hat{\mathbf{H}}_{\text{fd}} - \mathbf{H}_{\text{dmrs}}) \mathbf{P}_{\text{TD}}^T - \mathbf{E}\|_{\text{F}} \tag{19}$$

$$\overset{(a)}{\leq} \|(\hat{\mathbf{H}}_{\text{fd}} - \mathbf{H}_{\text{dmrs}}) \mathbf{P}_{\text{TD}}^T\|_{\text{F}} + \|\mathbf{E}\|_{\text{F}} \tag{20}$$

$$\overset{(b)}{\leq} \sigma_{\text{TD}} \|\hat{\mathbf{H}}_{\text{fd}} - \mathbf{H}_{\text{dmrs}}\|_{\text{F}} + \|\mathbf{E}\|_{\text{F}}, \tag{21}$$

where (a) follows from the triangle inequality and (b) from the fact that $\|\mathbf{AB}\|_{\text{F}} \leq \|\mathbf{A}\|_{\text{F}} \sigma_{\max}(\mathbf{B})$ [13].

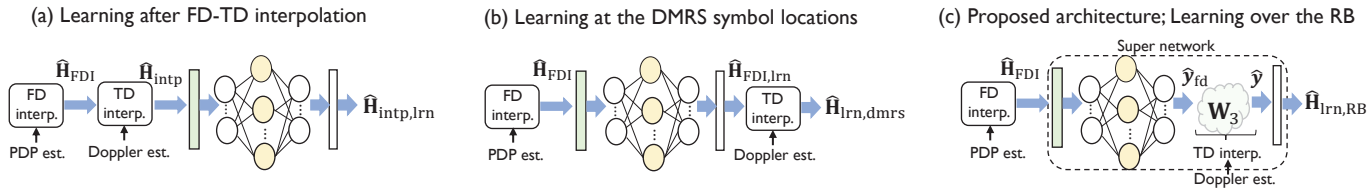

Fig. 3. Learning-aided channel estimation algorithms discussed in this paper. The network in (a) results in a higher complexity than those in (b) and (c). The proposed super network in (c) contains the TD interpolation block through which back-propagation is performed in training. The rectangles before the input to the network reshape complex matrices to real vectors, and those at the network output invert this operation.

We now consider the MSE loss over all the samples, i.e., $\mathcal{L}_{\mathrm{lrn,RB}} = \mathbb{E}[\delta^2]$, where the expectation is over the input distribution. We define $\epsilon = \mathbb{E}[\|\mathbf{E}\|_{\mathrm{F}}^2]$. Putting these definitions together with (21), we have

$$\mathcal{L}_{\mathrm{lrn,RB}} \leq \sigma_{\mathrm{TD}}^2 \mathbb{E}\left[\|\hat{\mathbf{H}}_{\mathrm{fd}} - \mathbf{H}_{\mathrm{dmrs}}\|_{\mathrm{F}}^2\right] + 2\mathbb{E}\left[\|\hat{\mathbf{H}}_{\mathrm{fd}} - \mathbf{H}_{\mathrm{dmrs}}\|_{\mathrm{F}}\|\mathbf{E}\|_{\mathrm{F}}\right] + \epsilon^2. \quad (22)$$

We note that the error $\mathbf{E}$ is independent of the estimation error at the DMRS locations, i.e., $\mathbf{H}_{\mathrm{fd}} - \mathbf{H}_{\mathrm{dmrs}}$. As a result, $\mathbb{E}[\|\hat{\mathbf{H}}_{\mathrm{fd}} - \mathbf{H}_{\mathrm{dmrs}}\|_{\mathrm{F}}\|\mathbf{E}\|_{\mathrm{F}}] = \mathbb{E}[\|\hat{\mathbf{H}}_{\mathrm{fd}} - \mathbf{H}_{\mathrm{dmrs}}\|_{\mathrm{F}}]\mathbb{E}[\|\mathbf{E}\|_{\mathrm{F}}]$. Now, we use the fact that $\mathbb{E}[|x|] \leq \sqrt{\mathbb{E}[|x|^2]}$ in (22) to write

$$\mathcal{L}_{\mathrm{lrn,RB}} \leq \sigma_{\mathrm{TD}}^2 \mathbb{E}\left[\|\hat{\mathbf{H}}_{\mathrm{fd}} - \mathbf{H}_{\mathrm{dmrs}}\|_{\mathrm{F}}^2\right] \quad (23)$$

$$+ 2\epsilon\sqrt{\mathbb{E}[\|\hat{\mathbf{H}}_{\mathrm{fd}} - \mathbf{H}_{\mathrm{dmrs}}\|_{\mathrm{F}}^2]} + \epsilon^2. \quad (24)$$

When the trainable weights of the super-network are initialized to $\boldsymbol{\xi}_{\mathrm{FDI,lrn}}$, we have $\mathbb{E}[\|\hat{\mathbf{H}}_{\mathrm{fd}} - \mathbf{H}_{\mathrm{dmrs}}\|_{\mathrm{F}}^2] = \mathcal{L}_{\mathrm{lrn,dmrs}}$. The training loss of the super-network under this initialization is defined as $\mathcal{L}_{\mathrm{lrn,RB}}^{(0)}$. From (23), we have

$$\mathcal{L}_{\mathrm{lrn,RB}}^{(0)} \leq \sigma_{\mathrm{TD}}^2 \mathcal{L}_{\mathrm{lrn,dmrs}} + 2\epsilon\sqrt{\mathcal{L}_{\mathrm{lrn,dmrs}}} + \epsilon^2. \quad (25)$$

After training the super-network, the loss $\mathcal{L}_{\mathrm{lrn,RB}} \leq \mathcal{L}_{\mathrm{lrn,RB}}^{(0)}$. Therefore, $\mathcal{L}_{\mathrm{lrn,RB}} \leq \sigma_{\mathrm{TD}}^2 \mathcal{L}_{\mathrm{lrn,dmrs}} + 2\epsilon\sqrt{\mathcal{L}_{\mathrm{lrn,dmrs}}} + \epsilon^2$. This bound depends on $\mathcal{L}_{\mathrm{lrn,dmrs}}$, i.e., the loss when training only at the DMRS locations, and the TD interpolation error ϵ.

V. Simulations

We consider an NR system operating at a carrier frequency of 3 GHz. The number of subcarriers within an RB is $N_{\mathrm{sc}} = 12$. A subcarrier spacing of 15 KHz is used such that every RB spans 180 KHz of bandwidth. Here, the RB occupies one slot or subframe of duration 1 ms. Each slot contains $N_{\mathrm{sym}} = 14$ OFDM symbols of which $N_{\mathrm{ctrl}} = 2$ symbols are used for the control. A total of 106 RBs are used over a bandwidth of 20 MHz. We consider two radio frequency chains at the transmitter and the receiver. The same neural network is applied to each of the four multiple-input multiple-output layers. The transmitter applies the same precoding over a group of 2 contiguous RBs, i.e., the bundle size is 2. It is important to note that our paper models and estimates the effective channel that already includes precoding.

The training dataset used in this paper comprises three types of channels: Extended Pedestrian A (EPA 5), Extended Vehicular A (EVA 30) and Extended Typical Urban (ETU 70). Here, the 5 in EPA 5 corresponds to a Doppler spread of 5 Hz in the channel. The modulation and coding scheme (MCS) used for transmission over EPA 5, EVA 30 and ETU 70 are $25, 25$ and 16. Now, we explain how the channel is estimated with the algorithms discussed in this paper. First, the PDP and the Doppler spread estimate are obtained using TRS. The TRS is sent using the channel state information-reference signal (CSI-RS), which is periodically transmitted every 20 ms. With the PDP and Doppler spread estimates from TRS, the LMMSE weights are computed for FD and TD interpolation blocks. To train the neural networks in this paper, we need $\hat{\mathbf{H}}_{\mathrm{intp}}$, $\hat{\mathbf{H}}_{\mathrm{FDI}}$ and $\mathbf{H}_{\mathrm{ideal}}$. We periodically dump this information over four consecutive slots at a periodicity of 10 ms, for an SNR range $[32, 38]$ dB. For the proposed method, the TD interpolation matrix $\mathbf{P}_{\mathrm{TD}}$ is also dumped for every pair of $\hat{\mathbf{H}}_{\mathrm{FDI}}$ and $\mathbf{H}_{\mathrm{ideal}}$. In this work, the matrix $\mathbf{P}_{\mathrm{TD}}$ contains the Bessel function coefficients which are purely real.

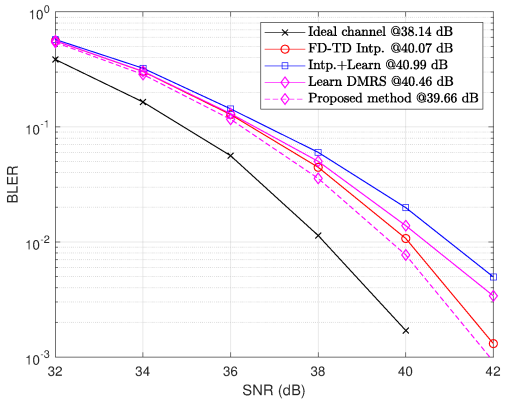

Fig. 4. BLER with SNR for EPA 5 channels. The proposed method outperforms the LMMSE technique, unlike the baseline algorithms.

We use the front-loaded DMRS configuration in Fig. 1 where $N_{\mathrm{dmrs}} = 2$. We randomly choose $300,000$ samples from the dumped channel dataset where every sample corresponds to the channel over a single RB. Then, this dataset is split at random into 90% training samples and 10% test samples. We set $N_{\mathrm{hid}} = 600$ for all the methods and use the Adam optimizer for training. For the parameters in this paper, we observe that the inference complexity of the networks in Fig. 3b and Fig. 3c is $6\times$ lower than the network in Fig. 3a.

We evaluate the performance of the algorithms in terms of

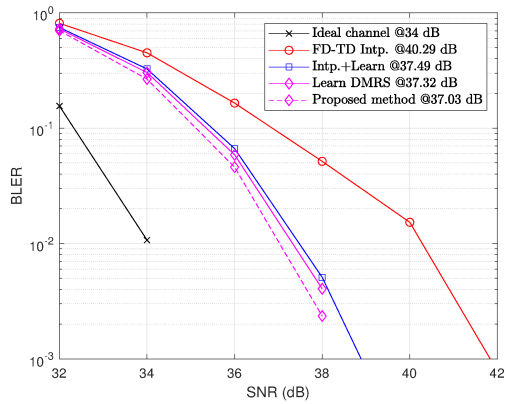

Fig. 5. BLER with SNR for EVA 30 channels. Learning-aided methods achieve similar BLER as the LMMSE at a much lower SNR.

the block error rate (BLER). The BLER is a better metric than the MSE as it captures the ultimate performance of the channel estimation algorithm. The legend for each curve indicates the SNR at which BLER=1%. Algorithms that achieve 1% BLER at a lower SNR are preferred for operation.

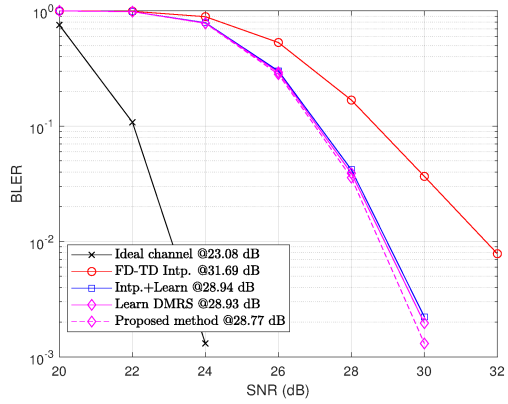

Fig. 6. BLER as a function of SNR for ETU 70 channels. The proposed method outperforms LMMSE and other baseline algorithms.

We observe from Fig. 4, Fig. 5 and Fig. 6 that the proposed algorithm achieves significant performance improvement than the LMMSE-based interpolation method ("FD − TDIntp.") for all the channel types. We notice that the baseline learning algorithms marked "Intp. + Learn" and "Learn DMRS" perform better for EVA and ETU channels. Note that "Intp. + Learn" is discussed in Sec. III-B and "Learn DMRS" is discussed in Sec. III-C. The baseline algorithms, however, perform poor for the EPA channels. We noticed that the performance improvement with the proposed method is due to two reasons. First, the super-network architecture in Sec. IV trains the MLP network while accounting for the online variation in the TD interpolation matrix. The training here is done in a end-to-end fashion to estimate channel over an RB. Such an end-to-end trained network is expected to outperform the MLP network in Fig. 3 trained using the surrogate function $\mathcal{L}_{\mathrm{lrn,dmrs}}$ [14]. Second, we found initialization of the trainable super network weights with $\boldsymbol{\xi}_{\mathrm{FDI,lrn}}$ is better than using random weights. The performance gap in the BLER relative to ideal channel is due to noise in the channel measurements. In summary, our results indicate the benefit of augmenting the LMMSE-based interpolation technique, that uses online statistics, with data-driven learning using one hidden-layer network.

VI. Conclusions

In this paper, we proposed a learning-aided channel estimation algorithm for 5G NR. Our algorithms exploit the joint time-frequency structure in the channels using a neural network, together with the online channel statistics such as the power delay profile and the Doppler spectrum. These statistics are computed using the tracking reference signal that is transmitted periodically. We showed using simulations that our approach performs better than LMMSE-based time-frequency interpolation technique. We would like to highlight that the proposed algorithm exploits channel structure at the level of resource blocks. Extending our algorithm to estimate channels over a bundle is an interesting research direction.

References

[1] R. Wittig, A. Goens, C. Menard, E. Matus, G. P. Fettweis, and J. Castrillon, "Modem design in the era of 5G and beyond: The need for a formal approach," in *Proc. of the 27th IEEE Intl. Conf. on Telecommun. (ICT)*, 2020, pp. 1–5.

[2] G. Noh, B. Hui, J. Kim, H. S. Chung, and I. Kim, "DMRS design and evaluation for 3GPP 5G new radio in a high speed train scenario," in *Proc. of the IEEE Global Commun. Conf. (GLOBECOM)*, 2017, pp. 1–6.

[3] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Trans. on broadcasting*, vol. 48, no. 3, pp. 223–229, 2002.

[4] P. Hoeher, S. Kaiser, and P. Robertson, "Two-dimensional pilot-symbol-aided channel estimation by Wiener filtering," in *IEEE Intl. Conf. on Acoustics, Speech, and Signal Process. (ICASSP)*, vol. 3, 1997, pp. 1845–1848.

[5] Y. Ding and H. Kwon, "Doppler spread estimation for 5G NR with supervised learning," in *IEEE Global Commun. Conf. (GLOBECOM)*, 2020, pp. 1–7.

[6] X. Dong, W.-S. Lu, and A. C. Soong, "Linear interpolation in pilot symbol assisted channel estimation for OFDM," *IEEE Trans. on Wireless Commun.*, vol. 6, no. 5, pp. 1910–1920, 2007.

[7] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 652–655, 2019.

[8] Y. Liao, Y. Hua, and Y. Cai, "Deep learning based channel estimation algorithm for fast time-varying MIMO-OFDM systems," *IEEE Commun. Lett.*, vol. 24, no. 3, pp. 572–576, 2019.

[9] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, "Enhancing least square channel estimation using deep learning," in *Proc. of the 91st IEEE Vehicular Technology Conference (VTC-Spring)*, 2020, pp. 1–5.

[10] M. B. Fischer, S. Dörner, S. Cammerer, T. Shimizu, B. Cheng, H. Lu, and S. t. Brink, "Wiener filter versus recurrent neural network-based 2d-channel estimation for V2X communications," *arXiv preprint arXiv:2102.03163*, 2021.

[11] D. Maruyama, K. Kanai, and J. Katto, "Performance evaluations of channel estimation using deep-learning based super-resolution," in *Proc. of the 18th IEEE Annual Consumer Commun. & Networking Conf. (CCNC)*, 2021, pp. 1–6.

[12] T. Cui and C. Tellambura, "Power delay profile and noise variance estimation for OFDM," *IEEE Commun. Lett.*, vol. 10, no. 1, pp. 25–27, 2006.

[13] C. D. Meyer, *Matrix analysis and applied linear algebra*. Siam, 2000, vol. 71.

[14] F. Sohrabi, K. M. Attiah, and W. Yu, "Deep learning for distributed channel feedback and multiuser precoding in FDD massive MIMO," *IEEE Trans. on Wireless Commun.*, 2021.