ChanEstNet: A Deep Learning Based Channel Estimation for High-Speed Scenarios

Yong Liao*, Yuanxiao Hua*, Xuewu Dai†, Haimei Yao*, Xinyi Yang*
*Center of Communication and TT&C, Chongqing University, Chongqing, China.
†Faculty of Engineering and Environment, Northumbria University, Newcastle upon Tyne, United Kingdom.

***Abstract*—Aiming at the problem that the downlink channel estimation performance is limited due to the fast time-varying and non-stationary characteristics in the high-speed mobile scenarios, we propose a channel estimation network based on deep learning, called ChanEstNet. ChanEstNet uses the convolutional neural network (CNN) to extract channel response feature vectors and recurrent neural network (RNN) for channel estimation. We use a large amount of high-speed channel data to conduct offline training for the learning network, fully exploit the channel information in the training sample, make it learn the characteristics of fast time-varying and non-stationary channels, and better track the features of channels changing in high-speed environments. The simulation results show that in the high-speed mobile scenarios, compared with the traditional methods, the proposed channel estimation method has low computational complexity and significant performance improvement.**

***Index Terms*—OFDM, channel estimation, high-speed channel, deep learning, fast time-varying channel, non-stationary channel**

I. Introduction

With the rapid development of the high-speed railways, mobile communication systems in high-speed environments have become a research hotspot. For orthogonal frequency division multiplexing (OFDM) systems, downlink channel estimation has received widespread attention [1]. In high-speed environments, the wireless channels are influenced by multipath and Doppler shift, which lead to the time and frequency selective fading (double-selective fading) and non-stationary characteristics of channels [2]. The traditional channel estimation methods are not suitable in this environments.

Traditional channel estimation methods include frequency-domain and time-domain channel estimation, which have different performance and complexity [3]. For the frequency-domain channel estimation, the channel frequency response (CFR) at the pilot symbols will be estimated at first and then the CFR at the data symbols is estimated by interpolation. The methods of frequency-domain channel estimation are relative simple and frequently used, such as the least squares (LS) [4] method, the linear minimum mean square error (LMMSE) [3] method. This kind of method assumes that the change of the CFR at pilot symbols and data symbols are linear, but in high-speed environments, the change of channel is relatively complex due to the joint influence of multipath and Doppler shift, which makes the assumption of linear change not suitable for high-speed channel. Therefore, the estimation performance of traditional interpolation methods is low. For the time-domain channel estimation methods, since it can directly estimate the channel impulse response (CIR), the inter-subcarrier interference (ICI) can be eliminated [5]. However, because it needs to estimate the CIR of each path, the number of parameters to be estimated in time-domain channel estimation methods will be much more than that in frequency-domain estimation methods, so it is necessary to find an effective method to reduce the estimated parameters of the time-domain channel estimation. For the problem of too many parameters to be estimated, the traditional time-domain channel estimation methods generally used the basis expansion model (BEM) to transform the CIR into a low-dimensional space formed by the basis vector. This method can effectively reduce the number of parameters to be estimated in time-domain channel estimation methods, such as the BEM-based LS method [6], which can reduce the estimation parameters by using BEM. However, due to LS estimation algorithm features lower estimation performance, it is not suitable for high-speed scenarios. In [7], BEM-based extended kalman filter (EKF) channel estimation algorithm is proposed, which can reduce estimation parameters by using BEM, and the data symbols channel information is obtained by EKF. Although such method has a certainly estimated performance improvement compared with traditional methods, its estimated complexity is too high. Therefore, it is a challenge to find a high-performance and low-complexity channel estimation method.

Deep learning (DL), which has been developed in recent years, has shown strong ability to deal with big data. Many scholars have applied it to wireless communication systems, such as millimeter wave (mmWave) channel estimation [8], channel sate information (CSI) feedback [9], [10], and data detection [11], [12], in order to achieve excellent performance. The DL approach has not been well investigated for channel estimation, especially for channel estimation in high mobility environments. Therefore, in order to solve the weaknesses of the traditional channel estimation methods in high-speed mobile scenarios, this paper proposes a DL-based channel estimation network, called ChanEstNet. The main contributions in this paper are described as follows: Firstly, an offline training and online prediction channel estimation network with convolutional neural network (CNN) and bidirectional long short-term memory (BiLSTM) network is designed; Secondly, for the time-domain channel estimation, since there are too many parameters to be estimated, the Maxpooling network is used to reduce dimensions of the parameters and minimize estimation complexity in this paper; Finally, we verify the estimated performance and robustness

978-1-5386-8088-9/19/$31.00 ©2019 IEEE

of the proposed algorithm with different high mobility environments. The simulation results show that the proposed DL-based algorithm has significant performance improvement and better robustness compared with the traditional methods.

II. System Model

For pilot-aided channel estimation methods applicable for OFDM systems, pilot pattern is the basis for subsequent research. For the block pilot pattern, the pilot symbols are inserted into all subcarriers in an OFDM symbol, namely the pilot symbols are fully inserted into the frequency domain, so it can effectively overcome frequency-selective fading [13]. The block pilot pattern has used by some mobile communication protocols, such as IEEE 802.11p [14], it indicates that the channel estimation based on the block pilot mode is applied extensively. We use block pilot channel estimation in this paper. The block pilot pattern and frame structure used in this paper is shown in Fig. 1.

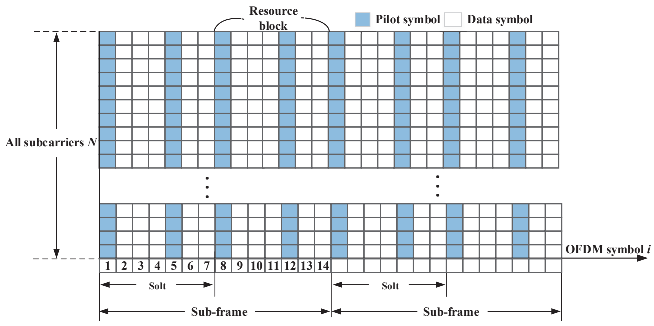

Fig. 1. Frame structure and pilot pattern.

For an OFDM system with N subcarriers and T OFDM symbols in a sub-frame, the transmitted symbols at tth OFDM symbol on the nth subcarrier are expressed as $\{\mathbf{s}_t(n)\}_{n=1}^{N}$, so the system model can be obtained

$$\mathbf{y}_t = \mathbf{F}^{\mathrm{H}}\mathbf{s}_t \otimes \mathbf{G}_t + \mathbf{z}_t \tag{1}$$

where $\mathbf{y}_t = [y_t(1), \ldots, y_t(N)]^{\mathrm{T}}$ denotes the received sequences in time domain, $\mathbf{G}_t \in \mathbb{C}^{N\times L}$ denotes the CIR matrix at tth OFDM symbol, L denotes the number of taps. $[\mathbf{F}]_{n,k} = \frac{1}{\sqrt{N}}\exp(-j\frac{2\pi}{N}kn)$ is the Fourier transforming matrix. $\mathbf{z}_t$ denotes the zero-mean additive complex Gaussian noise variable with variance is σ_z^2. $\otimes$ denotes the circular convolution. After removing the cyclic prefix (CP) and by performing discrete fourier transform (DFT), the received signal in the frequency domain can be expressed as

$$\tilde{\mathbf{y}}_t(n) = \mathbf{H}_t(n,n)\mathbf{s}_t(n) + \sum_{k=1,k\neq n}^{N}\mathbf{H}_t(n,k)\mathbf{s}_t(k) + \tilde{\mathbf{z}}_t(n) \tag{2}$$

where $\tilde{\mathbf{y}}_t \in \mathbb{C}^{N\times 1}$ and $\tilde{\mathbf{z}}_t \in \mathbb{C}^{N\times 1}$ denote the DFT result of $\mathbf{y}_t$ and $\mathbf{z}_t$, respectively. $\mathbf{H}_t \in \mathbb{C}^{N\times N}$ denotes the CFR matrix. The channel estimation aims to make the receiver estimate the channel matrix $\mathbf{H}_t$ through the known $\tilde{\mathbf{y}}_t$ and $\mathbf{s}_t$.

III. ChanEstNet Channel Estimation

In this section, the framework of ChanEstNet and the structure of proposed learning network are briefly introduced at first. Then, the form of input data and data flow in the learning network are illustrated in detail. Finally, the model training of proposed learning network is discussed.

A. The Framework of ChanEstNet

The ChanEstNet network is divided into two phases: offline training and online prediction, and its framework is shown in Fig. 2.

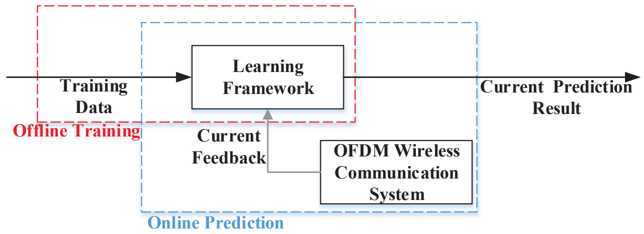

Fig. 2. The channel estimation framework of ChanEstNet.

As shown in Fig. 2, for offline training phase, we train the learning network by using a large amount of standard high-speed channel data values which are collected by WINNER II channel model [15]. For online prediction phase, the input of the learning network is the feedback of OFDM wireless communication system. The learning network includes the one dimension (1D) CNN network, the 1D Maxpooling network, BiLSTM network and fully connected neural network (FCNN). The structure of learning network is shown in Fig. 3.

The CNN network is mainly used to extract the pilot sequence feature values, which is composed of a series of parallel filters. These filters are connected to the input signal through a set of weights, and the convolution is calculated along the horizontal direction (time axis). Generally, a CNN network consists of multiple convolution filters, each filter processes the data on different channels and calculates the convolution summation through a sliding window. Set $\mathbf{W}$ as a convolution filter, by sliding the filter on the data to be convolved, and the convolution output is obtained by weighting the sum of the data. Therefore, the transformation formula of CNN is

$$\mathbf{x}' = f(\mathbf{W} * \mathbf{x} + \mathbf{b}) \tag{3}$$

where $\mathbf{b}$ is the offset vector, $\mathbf{x}$ is convolutional data and $f(\cdot)$ is the activation function. The factor $*$ denotes the convolution operation. In this paper, we use double tangent activation function (tanh), its expression is

$$\tanh(x) = \frac{1-\mathrm{e}^{-2x}}{1+\mathrm{e}^{-2x}} \tag{4}$$

The 1D Maxpooling network is mainly used to reduce the number of parameters to be estimated, and a pooled window is used to find the maximum value of the filter output. For the frequency-domain channel estimation, the parameters to be estimated are less, so this layer can be ignored. For the time-domain channel estimation, its expression is

$$x' = \max(\mathbf{x}') \tag{5}$$

The LSTM network is used to predict the data, which is a combination of two LSTM networks. One of the LSTM

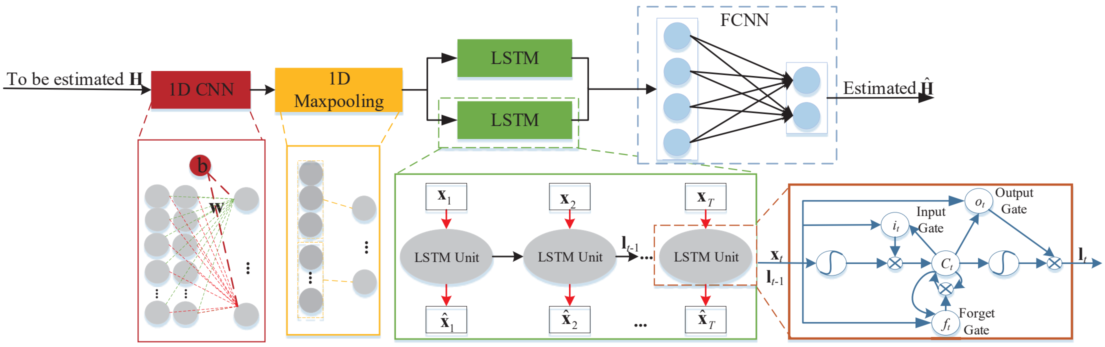

Fig. 3. The structure of proposed learning network. In this papar, the FCNN is time distributed neural network, which is independent fully connected for each time step signal.

networks is used for forward data prediction and another LSTM network is used for backward prediction. LSTM network has inherent memory cells and can keep the previously extracted information for a long period for later prediction. Each LSTM network is composed of several LSTM units. Each unit is composed of the input gate, forget gate, output gate and memory unit. The mathematic description of the LSTM structure is shown as

$$\mathbf{i}_t = \sigma(\mathbf{b}_{\mathrm{i}} + \mathbf{U}_{\mathrm{i}}\mathbf{x}_t + \mathbf{W}_{\mathrm{i}}\mathbf{l}_{t-1}) \tag{6}$$

$$\mathbf{f}_t = \sigma(\mathbf{b}_{\mathrm{f}} + \mathbf{U}_{\mathrm{f}}\mathbf{x}_t + \mathbf{W}_{\mathrm{f}}\mathbf{l}_{t-1}) \tag{7}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \sigma(\mathbf{b}_{\mathrm{c}} + \mathbf{U}_{\mathrm{c}}\mathbf{x}_t + \mathbf{W}_{\mathrm{c}}\mathbf{l}_{t-1}) \tag{8}$$

$$\mathbf{o}_t = \sigma(\mathbf{b}_{\mathrm{o}} + \mathbf{U}_{\mathrm{o}}\mathbf{x}_t + \mathbf{W}_{\mathrm{o}}\mathbf{l}_{t-1}) \tag{9}$$

$$\mathbf{l}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{10}$$

where $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$, $\mathbf{c}_t$ and $\mathbf{l}_t$ are the input gate, forget gate, output gate, memory unit and hidden layer vector, respectively. $\mathbf{U}_{\mathrm{i}}$, $\mathbf{W}_{\mathrm{i}}$, $\mathbf{U}_{\mathrm{f}}$, $\mathbf{W}_{\mathrm{f}}$, $\mathbf{U}_{\mathrm{c}}$, $\mathbf{W}_{\mathrm{c}}$, $\mathbf{U}_{\mathrm{o}}$ and $\mathbf{W}_{\mathrm{o}} \in \mathbb{R}^{d\times d}$ are the weights matrix of LSTM network, $\mathbf{b}_{\mathrm{i}}$, $\mathbf{b}_{\mathrm{f}}$, $\mathbf{b}_{\mathrm{c}}$ and $\mathbf{b}_{\mathrm{o}} \in \mathbb{R}^{d}$ are the offsets of LSTM. The weights and offsets are learned by training, σ is the sigmoid function, $\odot$ is the element multiply and d is the input sequence dimension, which equal the number of OFDM subcarriers in this paper, t is the number of input sequences, which equal the number of OFDM symbols, in another word, it is the number of LSTM units. The update equation and output of LSTM network at each time step t can be simplified as (11) and (12)

$$\mathbf{h}_t = \mathrm{LSTM}(\mathbf{h}_{t-1}, \mathbf{x}_t, \Theta) \tag{11}$$

$$\hat{\mathbf{x}}_t = \tanh(\mathbf{W}_{\mathrm{l2o}}\mathbf{h}_t + \mathbf{b}_{\mathrm{l2o}}) \tag{12}$$

where $\mathrm{LSTM}(\cdot)$ is the combination of (6)-(10), Θ is all the parameters of the the LSTM network. $\mathbf{W}_{\mathrm{l2o}}$ and $\mathbf{b}_{\mathrm{l2o}}$ denote hidden-to-output weight and offset. BiLSTM network is a combination of two LSTM networks, so we can get the output transformation formula of the BiLSTM network as

$$\mathbf{p}_t = \mathrm{Concat}(\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_t^{\dagger}) \tag{13}$$

where $\hat{\mathbf{x}}_t^{\dagger}$ denotes the backward output of BiLSTM at time step t, $\mathbf{p}_t$ denotes the output of BiLSTM at time step t. $\mathrm{Concat}(\cdot)$ denotes the function that concatenate two tensors in a specified dimension. So the output of BiLSTM at each time step can be expressed as

$$\mathbf{p}_t = \mathrm{BiLSTM}(\mathbf{l}_{t-1}, \mathbf{l}_{t-1}^{\dagger}, \hat{\mathbf{x}}_t, \hat{\mathbf{x}}_t^{\dagger}, \Theta_{\mathrm{bi}}) \tag{14}$$

where $\mathbf{l}_{t-1}^{\dagger}$ denotes the backward hidden vector of the BiLSTM network at time step $t-1$, $\hat{\mathbf{x}}_t^{\dagger}$ denotes the backward input of the BiLSTM network at time step t. Θ_{bi} and $\mathrm{BiLSTM}(\cdot)$ denote all the parameter and transformation function of BiLSTM network, respectively. The output of BiLSTM network at each time step is transformed by using FCNN in the last layer of learning network. In contrast to CNN, each input elements of FCNN are connected to different weights, and the output is the weighted sum of all input elements.

B. Data Flow In ChanEstNet

Input data passes through the proposed learning network that generates the predicted CSI, and the data flow is shown in Fig. 4.

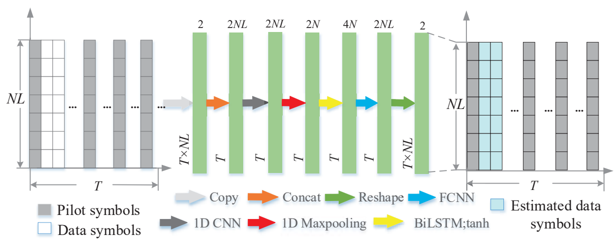

Fig. 4. Data flow of the proposed learning network.

Specifically, the original data will be pre-processed at first. The original CSI data is complex, so we extract the real and imaginary parts of the original data and then concatenate them into a dimension, the transformation process is the orange arrow in Fig. 4. Because LSTM network needs time sequence as input, the number of the OFDM symbols is taken as time sequence number. After preprocessing, the data is used as the input of 1D CNN network and feature vectors were extracted through CNN. In particular, for time-domain channel estimation, since the CIR is directly estimated, the original data has added delay dimension compared to the frequency-domain channel estimation. Unlike the frequency domain channel estimation, a 1D Maxpooling layer is used for compressing the parameters to be estimated. After feature extraction or parameters dimensional reduction, the data is input to the BiLSTM network, and the CSI at the data symbols is obtained via the BiLSTM network. Finally, the output dimension of the BiLSTM network is reduced through the FCNN network. The following part introduces the input data, extraction of the frequency feature value and channel estimation in details, respectively.

1) Input data: In this paper, the input data of the learning network is the CSI matrix in a sub-frame. For frequency-domain channel estimation, the CSI matrix can be expressed as $\mathbf{H} \in \mathbb{C}^{T\times N}$, and for time-domain channel estimation, the CSI matrix can be expressed as $\mathbf{G} \in \mathbb{C}^{T\times NL}$. The CSI at pilot symbols are estimated by LS, and the CSI at data symbols are set as 0. Therefore, the input of the learning network can be expressed as $\mathbf{H} \in \mathbb{C}^{T\times NL}$, L is 1 for frequency-domain channel estimation. The LSTM network needs time sequence as input. Therefore, the CSI is transformed into a form of sequence, its input data is expressed as

$$\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_t, \ldots, \mathbf{h}_T] \tag{15}$$

where $\mathbf{h}_t \in \mathbb{C}^{1\times NL}$ denotes the CSI at the tth OFDM symbols. Since the channel data is a complex signal, it is necessary to pre-process the data before inputted the proposed learning network. By extracting the real part and the imaginary part of the input data and concatenate the real part and the imaginary part, the input data becomes $\mathbf{H}' \in \mathbb{R}^{T\times 2NL}$ and $\mathbf{h}'_t \in \mathbb{R}^{1\times 2NL}$.

2) Extraction of frequency feature vector: The input data is pre-processed and sent to the CNN network. The main task of CNN is to extract and select the frequency feature vector. In order to extract the feature vector, the CNN network performs a convolution operation on $\mathbf{H}'$, and the number of filters is $2NL$. Based on (3), the output of the CNN network is

$$\mathbf{H}'' = f(\mathbf{W} * \mathbf{H}' + \mathbf{b}) \tag{16}$$

After data is processed by the CNN network, the output dimensions do not change, namely $\mathbf{H}'' \in \mathbb{R}^{T\times 2NL}$. In particular, for time-domain channel estimation, the output of the CNN will be reduced by Maxplooing network. Setting the pooling window size of the Maxpooling network is $1 \times L$, so the data dimension after pooling is $\mathbf{H}'' \in \mathbb{R}^{T\times 2N}$, $\mathbf{h}''_t \in \mathbb{R}^{1\times 2N}$.

3) Channel estimation: Our proposed learning network aims to predict the current CSI based on the past and current feedback and future data. Considering the LSTM network is excellent in learning of the sequence task, the LSTM network is used to predict the current CSI in this paper. For forward prediction, the CSI at the latter moment is predicted by the CSI at the previous moment. For the backward prediction, the CSI at the previous moment is predicted by the CSI at latter moment, the forward and backward pilot information is fully utilized to further improve the accuracy of the channel estimation. For channel estimation, each time step of the BiLSTM network has an output. Based on (14), we can get the predicted CSI at each time step as

$$\mathbf{h}'''_t = \mathrm{BiLSTM}(\mathbf{l}_{t-1}, \mathbf{l}^{\dagger}_{t-1}, \mathbf{h}''_t, \mathbf{h}''^{\dagger}_t, \Theta_{\mathrm{bi}}) \tag{17}$$

The output dimension of the BiLSTM network is twice of the LSTM network, i.e., $\mathbf{h}'''_t \in \mathbb{R}^{4N}$. Finally, the output of each time step of the BiLSTM network is dimensionally transformed through the FCNN, so that the final output dimension is consistent with the input dimension, that is, the number of fully connected neurons is $2NL$, its transformation expression is

$$\mathbf{h}''''_t = \tanh(\mathbf{W}_{1,t}\mathbf{h}'''_t + \mathbf{b}_{1,t}) \tag{18}$$

where $\mathbf{W}_{1,t}$ and $\mathbf{b}_{1,t}$ are the weight and offset of the FCNN at time step t, respectively. The final outputs are then reshaped into two $T \times NL$ tensors as the final estimated real and imaginary parts, and then add the real and imaginary parts together to get the final output as

$$\hat{\mathbf{H}} = [\hat{\mathbf{h}}_1, \ldots, \hat{\mathbf{h}}_t, \ldots, \hat{\mathbf{h}}_T] \tag{19}$$

C. Model Training

In order to train the ChanEstNet network, we use the end-to-end approach to obtain all the weights and biases in the ChanEstNet network. We set the transformation formula and all parameters of the ChanEstNet network as $f_{\mathrm{est}}(\cdot)$ and Θ_{est}, respectively, and the estimated channel matrix can be denoted by $\hat{\mathbf{H}} = f_{\mathrm{est}}(\mathbf{H}, \Theta_{\mathrm{est}})$. We use adaptive moment estimation (ADAM) algorithm to update the set of parameters for the ChanEstNet network. The ADAM algorithm is different from the traditional gradient descent algorithm with fixed learning rate. It can update the learning rate adaptively by training. The loss function of the network is mean squared error (MSE), so the predicted loss of our model is

$$L(\Theta_{\mathrm{est}}) = \frac{1}{T}\sum_{i=1}^{M}(f_{\mathrm{est}}(\mathbf{H}, \Theta_{\mathrm{est}}) - \mathbf{H}^*_i)^2 \tag{20}$$

where the $\mathbf{H}^*_i$ is the supervision message, and M is the total number of samples in the training set. In this papar, the WINNER II [15] wireless channel model are used to simulate fast time-varying and non-stationary channels and generate training samples. The training, validation, and testing sets have 10000, 2000, and 500 samples, respectively. The epochs, initial learning rate and batch size are set as 100, 0.01 and 200, respectively.

IV. Analysis On Simulation Results

In this section, we will evaluate the performance of time-domain channel estimation and frequency-domain channel estimation of the proposed methods in different environments. The MATLAB and Python simulation platform are used for simulation analysis of the proposed methods. The parameters of simulation system are shown in Table I.

A. Normalized Mean Square Error

The Fig. 5 compares normalized MSE (NMSE) performance of the frequency-domain channel estimation LS [4] method, LMMSE [3] method and the proposed frequency-domain channel estimation F-ChanEstNet method. The Fig. 6 compares the time-domain channel estimation BEM-based LS method [6], BEM-based EKF [7] method and the proposed time-domain channel estimation T-ChanEstNet method in different speed environments.

The simulation results show that the NMSE curve of three frequency-domain methods are much different. When the

TABLE I. Parameters of simulation system.

Parameters	Values
Frequency of carrier	2.8 GHz
Bandwidth	1.8 MHz
Number of subcarriers	72
Length of FFT	72
Length of CP	9
Modulation	QPSK
Number of taps	12
Non-stationary channel	WINNER II [15]

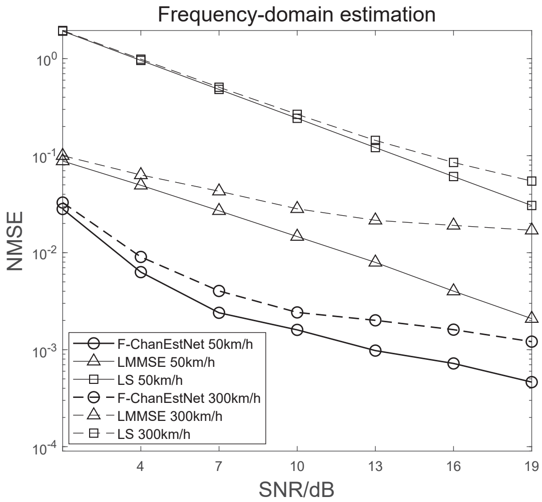

Fig. 5. NMSE comparison of frequency-domain channel estimation methods.

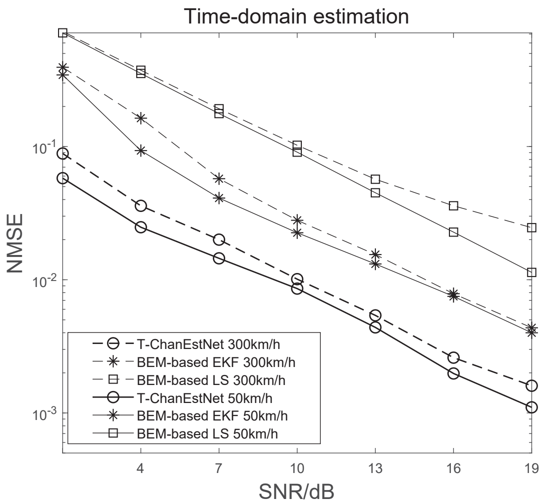

Fig. 6. NMSE comparison of time-domain channel estimation methods.

speed is 50 km/h, the signal-to-noise ratio (SNR) gain of the LMMSE algorithm is about 16 dB higher than that of LS algorithm and the SNR gain of the F-ChanEstNet is about 9 dB higher than that of LMMSE algorithm. This is because the LS algorithm is simpler than LMMSE algorithm, the estimation error is higher due to ignore the effects of noise, and the LMMSE can improve estimation precision by using the channel statistics information. At a speed of 300 km/h, the NMSE of the three methods is on an upward trend compared with 50 km/h, and the SNR gain of LMMSE method to LS method is 5 dB averagely. At this time, the difference of SNR gain between F-ChanEstNet and LMMSE is significant, the main reason is that the hypothesis of linear interpolation that the change of CIR is linear is not applicable for high speed channel. The proposed method first learns the characteristics of channel variation through training, and then estimates the response at the data symbols through nonlinear mapping, so it is more suitable for high-speed scenario and is also excellent in a low-speed environments.

For time-domain estimation, the NMSE performance of various methods is very close at different speeds. Since the time-domain estimation estimates the channel gain for each path, the ICI caused by the Doppler shift can be estimated. From Fig. 6 , we can see that the SNR gain of BEM-based EKF algorithm is about 3.5 dB higher than that of BEM-based LS algorithm and the SNR gain of T-ChanEstNet is 4 dB higher than that of BEM-based LS, the main reason is that the CIR linear change theory is not suitable for high-speed channel. Although the BEM-based EKF methods are similar to T-ChanEstNet at this time, the estimation time is too long due to higher complexity. Therefore, the proposed method can also have higher performance in time-domain estimation.

In a word, the ChanEstNet channel estimation method shows higher NMSE performance both in time-domain estimation and frequency-domain estimation. For the low-speed environments, the SNR gain of the proposed method will gradually reduce with the increase of SNR. For a high-speed environments, the proposed method features excellent NMSE performance.

B. Bit Error Rate

The bit error rate (BER) performance is the macro index to measure the influence of the channel estimation method on the whole system performance. The Fig. 7 and Fig. 8 compare the BER performance of different algorithms at speeds of 50 km/h and 300 km/h.

For the frequency-domain channel estimation, it can be seen from Fig. 7 that at 50 km/h, the LMMSE algorithm has about 4.5 dB SNR gain to the LS algorithm, the BER performance of F-ChanEstNet algorithm and LMMSE algorithm is equivalent at a low SNR (SNR$\leq$10dB) and F-ChanEstNet algorithm has about 6 dB SNR gain compared with LS algorithm. However, in the case of high SNR (SNR$>$10dB), the BER of F-ChanEstNet method decreases rapidly and its BER performance is significantly superior to that of LMMSE algorithm. At 300 km/h, the LMMSE algorithm has an SNR gain of about 4 dB to the LS algorithm. Similarly, at a low SNR, The F-ChanEstNet method has the same BER performance as the LMMSE algorithm, and has an SNR gain about 5 dB to LS. In the case of high SNR, the F-ChanEstNet method has about 4 dB SNR gain to LMMSE algorithm.

For the time-domain estimation, as shown in Fig. 8, the BER performance of T-ChanEstNet is equivalent to that of BEM-based EKF method at 50 km/h, its BER performance is significantly superior to that of BEM-based LS method, because the CIR change tends to be stationary in a low-speed environments and BEM-based EKF algorithms is also

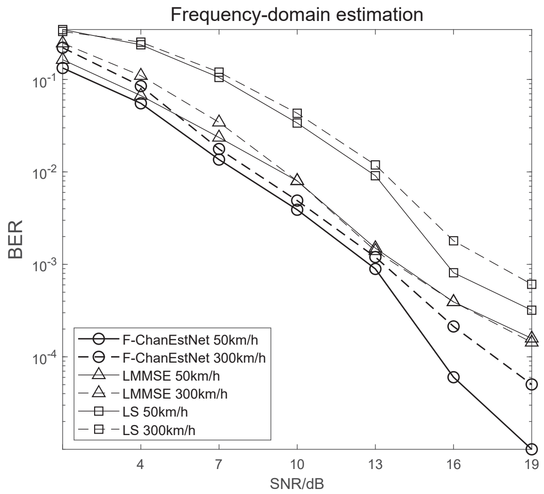

Fig. 7. BER comparison of frequency-domain channel methods.

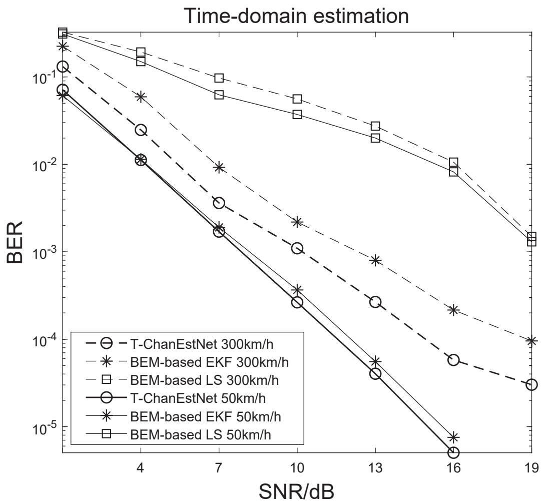

Fig. 8. BER comparison of time-domain channel estimation methods.

suitable well. Since the the LS estimation algorithm is too simple, the BER performance is not better. At 300 km/h, the BER performance of different channel estimation methods will tend to converge with the growth of SNR, due to the influence of the channel environment. The SNR gain of BEM-based EKF is about 6.5 dB to BEM-based LS, at this time, the SNR gain of the T-ChanEstNet algorithm is about 2 dB compared with BEM-based EKF algorithm and its SNR gain reaches about 9.5 dB to BEM-based LS algorithm.

Overall, the BER performance of the proposed method is superior to that of other algorithms both in the frequency-domain estimation and time-domain estimation at a high-speed scenarios, which reflects the overall performance that is more adaptive to high-speed scenarios.

V. Conclusions

In the paper, for the weakness of the traditional channel methods in a high-speed scenarios, a channel estimation method based on deep learning is proposed. The nonlinear mapping characteristics of deep learning can better adapt to the changing characteristics of high-speed channels, and the channel information in the offline training sample can be effectively used to improve the accuracy of channel estimation. Finally the performance of the ChanEstNet method is analyzed in a high-speed scenarios through simulation comparison of the time-domain estimation and frequency-domain estimation. The simulation results show that in the case of low estimation complexity, the channel estimateion precision and whole system performance of the ChanEstNet method is both superior to that of the traditional methods.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61501066), the Chongqing Frontier and Applied Basic Research Project (No. cstc2015jcyjA40003), the graduate research and innovation foundation of Chongqing, China (No. CYS18061), and the Fundamental Research Funds for the Central Universities (No.106112017CDJXY500001).

References

[1] L. Yang, G. Ren, B. Yang and Z. Qiu, "Fast time-varying channel estimation technique for LTE uplink in HST environment," *IEEE Trans. Veh. Technol.*, vol. 61, no. 9, pp. 4009-4019, Nov. 2012.

[2] A. Ghazal et al, "A non-stationary IMT-advanced MIMO channel model for high-mobility wireless communication systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 4, pp. 2057-2068, April 2017.

[3] C. S. Yong, J. Kim and W. Yang. *MIMO-OFDM wireless communications with MATLAB*. KOR: Wiley Publishing, 2010.

[4] H. D. Zarrinkoub. *Understanding LTE with MATLAB: from mathematical modeling to simulation and prototyping*. USA: Wiley Publishing, 2014

[5] Hlawatsch, Franz, and G. Matz. *Wireless communications over rapidly time-varying channels*. Academic Press, 2013.

[6] J. l. Zhang et al, "Using basis expansion model for physical layer authentication in time-variant system," 2016 *IEEE Conf. Commun. Netw. Secu. (CNS)*, Philadelphia, PA, 2016, pp. 348-349.

[7] X. Shen, Y. Liao and X. Dai, "BEM-based EKF-RTSS channel estimation for non-stationary double-selective channel," *IEEE/CIC Int. Conf. Commun. China (ICCC)*, pp. 1-6, 2018

[8] H. He, C. Wen, S. Jin and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852-855, Oct. 2018.

[9] C. Wen, W. Shih and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748-751, Oct. 2018.

[10] T. Wang, C. Wen, S. Jin and G. Y. Li, "Deep learning-based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wireless Commun. Lett.*, to appear.

[11] H. Ye, G. Y. Li and B. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114-117, Feb. 2018.

[12] X. Gao, S. Jin, C. Wen and G. Y. Li, "ComNet: combination of deep learning and expert knowledge in OFDM receivers," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2627-2630, Dec. 2018.

[13] Y. Liu, Z. Tan, H. Hu, L. J. Cimini and G. Y. Li, "Channel estimation for OFDM," *IEEE Commun. Surv. Tuts.*, vol. 16, no. 4, pp. 1891-1908, 2014.

[14] J. Gozalvez, M. Sepulcre and R. Bauza, "IEEE 802.11p vehicle to infrastructure communications in urban environments," *IEEE Commun. Mag.*, vol. 50, no. 5, pp. 176-183, May 2012.

[15] J. Meinila et al. *WINNER II channel models*. USA: John Wiley & Sons, Ltd. Press, 2008.