



Research article

An improved least squares (LS) channel estimation method based on CNN for OFDM systems

Hua Yang*, Xuan Geng, Heng Xu and Yichun Shi

College of Information Engineering, Shanghai Maritime University, 1550 Haigang Avenue, Pudong District, Shanghai 201306, China

* **Correspondence:** Email: yanghua@shmtu.edu.cn.

Abstract: Least squares (LS) is a commonly used pilot-based channel estimation algorithm in orthogonal frequency division multiplexing (OFDM) systems. This algorithm is simple and easy to implement because of its low computation complexity. However, it has poor performance, especially at low signal-to-noise ratio (SNR). To solve this problem, an improved LS channel estimation method based on convolutional neural network (CNN) is proposed on the basis of analyzing the traditional LS channel estimation methods. A channel estimation compensated network is designed based on CNN, which can solve the problem of performance degradation of the mean square error (MSE) through the online and offline modules. By designing the input-output relations, training data set, and testing data set, a CNN network is iteratively trained to learn the relevant features of the channels, so that the traditional LS estimation value can be corrected to improve the accuracy. Simulation results show that the proposed method can achieve better performance in bit error rate (BER) and MSE, compared with the traditional channel estimation methods.

Keywords: convolutional neural networks; OFDM systems; least squares channel estimation; low signal-to-noise ratio

1. Introduction

Orthogonal frequency division multiplexing (OFDM) is a multi-carrier modulation method, which divides the channel into a number of orthogonal sub-channels to transmit data in parallel, thus improving the efficiency of spectrum utilization in systems. One of the keys to perform coherent detection at the OFDM receivers is the channel estimation [1], and its precision will directly affect the performance of the whole system.

Common channel estimation methods include channel estimation using reference signal (pilot-based estimation), blind channel estimation and semi-blind channel estimation [2]. Although blind

channel estimation and semi-blind channel estimation have high spectrum efficiency, their computation is too complex for practical application. Channel estimation methods based on pilot signals are commonly used in OFDM systems [3, 4], moreover, the pilot channel estimation algorithms are classified into LS channel estimation, maximum likelihood (ML) channel estimation, and minimum mean square error (MMSE) channel estimation [5]. LS channel estimation has the lowest complexity among the three types of algorithms. However, its performance will deteriorate significantly at low signal-to-noise ratio (SNR), since it ignores the influence of noise during estimation. Channel impulse response (CIR) has a great influence on ML channel estimation. In ML channel estimation, the parameters are estimated by maximizing the likelihood function to obtain the channel frequency response, however, the performance of this method will decrease significantly for a longer CIR. MMSE method has better performance, but it needs the prior statistics of the channel [6].

Considering the above analysis of traditional channel estimation algorithms, various new methods for improving traditional algorithms are currently being explored in order to further enhance channel estimation performance. Wang et al. [7] has improved the LS algorithm by applying wavelet denoising (WD) and distance decision analysis (DDA) to implement two-stage denoising based on the transform domain. The results showed that this approach could improve the detection performance of the LS channel estimation method based on the transform domain at low SNR. However, due to the relatively complex architecture of the proposed algorithm, it has high computational complexity. Especially for the denoising stage of distance decision analysis, it becomes computationally expensive as the dataset size increases. Moreover, all the computations are carried out online, which can limit its practicality for detecting real-time channel statement. Li et al. [8] proposed an improved MMSE algorithm by designing a special training sequence to convert the multi-antenna problem into a single-antenna problem, which greatly reduces the size of the inverse matrix in the MMSE algorithm. At the same time, a singular value decomposition method is designed for near-rank approximation, in order to achieve an ideal low-order estimator and further simplify the algorithm.

As a key technology of artificial intelligence, deep learning (DL) has been applied in a variety of traditional fields and has achieved significant performance improvement. DL is likewise used for channel estimations in OFDM systems. One approach is to regard the neural network as a “black box” and customize the input-output model of the network in the channel estimation algorithm, in which the parameters of the network can be learned by using training data sequences. He et al. [9] designed a learned denoising-based approximate message passing (LDAMPA) network based on learning denoising. Usually, channel estimation is performed by estimating the channel values at pilot locations and then performing some operations to estimate the channel values. This algorithm, however, learns the noise distribution in the channel and performs subtraction operations at the receiver to obtain the corresponding channel values. CNN is a kind of feedforward neural networks with deep structure and convolutional computation. Soltani et al. [10] treats the channel transmission matrix as a two-dimensional image and introduces CNN for denoising processing to obtain channel state information (CSI). Because of the CNN characteristics of local receptive field, weight sharing and spatial or temporal down-sampling, the relevant features of the channel from the received signals can be effectively extracted and the accuracy of channel estimation can be increased. From the perspective of signal detection, He et al. [11] no longer focus on estimating the channel at the pilot frequency, but concentrate on the transmitted and received information, in which its input-output relations can be learned by the neural network and the received signal can be predicted directly at the receiver. Recurrent neural

network (RNN), particularly long short term memory (LSTM) can process time-series OFDM signals because of its memory ability, so that it is well adapted for time-varying channels. Mohammed et al. [12] adopts LSTM to implement the DL-based channel estimation for OFDM 5G systems. Such a “black box” approach integrates the channel information and variable parameters into the “black box”, so that it has fewer calculation parameters and is highly adaptable. However, the invisible channel information makes the interpretability of the channel state worse, and the channel estimation performance is uncontrollable. Especially in the complex channel environments, the good results for network training cannot be guaranteed when encountering unseen channel characteristics.

In addition to the approach treating neural networks as a “black box”, there is another approach of combining the traditional channel estimation methods with the neural networks, that leverages the strengths of both techniques. The combination approach takes a sequential approach where traditional estimation is initially performed to obtain initial channel estimates, which are then improved using a neural network. That is, this approach can improve the accuracy and stability of channel estimation. As shown in the Table 1, each of the approaches based on DL has its own features, which should be selected based on the specific application scenario and requirements.

Table 1. Comparison of “black box” approach and combination approach.

	“black box” approach	combination approach
Method	the neural network is regarded as a “black box” and can learn input-output relations	combine the neural network and the traditional channel estimation methods
Purpose	directly predict the received signal at the receiver	improve the estimation results based on traditional method
Request for the dataset	require a large amount of training data to achieve optimal performance	potentially offer better accuracy with less training data
Processes	process of the deep neural network	two steps (channel estimation + the deep neural network)
Strengths	high accuracy compared to the traditional methods; less parameters; time series data processing ability (RNN, LSTM); adaptable to different environments	improves the accuracy of channel estimation; robustness
Weaknesses	computationally expensive; low interpretability	high complexity of the channel estimation process
Applications	all kinds of channels but need sufficient data for training	complex and changeable channel environment

Considering the shortcomings of the traditional LS channel estimation method, we propose an improved LS channel estimation algorithm based on CNN. In our proposal a CNN network is added after the channel estimation to correct the channel estimates. Benefiting from the initial estimation provided

by LS, the reliance on a massive dataset for training the neural network model is reduced. The proposed scheme has roughly two steps. In the offline training phase, the data for simulation are generated first, including the sending data and receiving data, which are used to acquire the channel estimates through traditional LS estimation. Then, the channel estimates are input into the CNN network, and the channel data generated by Rayleigh channel model are set as targets. Using MMSE principle, the CNN network learns the distribution features of the channel by iterative training. Then, it moves to the online restoration phase. In this phase, the new channel estimates generated by the traditional LS estimation method are input into the trained CNN network. Thus the new channel estimates can be corrected and get closer to the real values. Simulation results show that this method can improve the performance of the original LS method. Its effectiveness and feasibility has also been verified by the simulation results.

2. LS channel estimation in OFDM systems

In the process of traditional OFDM communication, by performing the constellation mapping, the bit stream is mapped to the symbols at the transmitter. Then the serial symbol sequence is converted to N parallel symbol streams which are modulated to different subcarriers, and N_{fft} -point inverse fast fourier transform (IFFT) is implemented. In order to eliminate inter-symbol interference (ISI) caused by the multipath propagation, a cyclic prefix (CP) length of x_{GI} is inserted, which is greater than or equal to the maximum channel delay, τ . The serial time domain signal $x(n)$ converted from parallel format reaches the receiver through channel $h(n)$. At the receiving end, the mathematical model for receiving the signal of the multipath channel is usually expressed as:

$$y(n) = x(n) \otimes h(n) + w(n) \quad (2.1)$$

where $x(n)$, $y(n)$, $h(n)$ and $w(n)$ respectively indicates the sending signal, receiving signal, channel parameter and noise. $x(n)$ is a vector of length $N_s + x_{GI}$, and N_s is the number of subcarriers in one OFDM symbol, $0 \leq n \leq N_s + x_{GI} - 1$. All signals are complex, thus \otimes is introduced for convolution computation. The corresponding receiving signal in frequency domain can be defined as:

$$Y(k) = X(k)H(k) + W(k) \quad (2.2)$$

where $X(k)$, $Y(k)$, and $W(k)$ are derived from $x(n)$, $y(n)$ and $w(n)$, respectively, by using discrete Fourier transform (DFT), $0 \leq k \leq N_s + x_{GI} - 1$. The system is assumed to be completely synchronized.

Based on the Eq (2.2), the objective function is established according to LS Criterion as follows:

$$J(\tilde{H}) = \arg \min \left\{ (Y - X\tilde{H}_{ls})^H (Y - X\tilde{H}_{ls}) \right\} \quad (2.3)$$

where \tilde{H} is the channel estimation result based on pilot symbols. The partial derivative of Eq (2.3) can be written as:

$$\tilde{H}_{ls} = X^{-1}Y = H + X^{-1}W \quad (2.4)$$

Channel estimation value \tilde{H} is substituted into mean square error (MSE) formula. It can be calculated as:

$$MSE_{ls} = E \left\{ (H - \tilde{H}_{ls})^H (H - \tilde{H}_{ls}) \right\} = E \left\{ (X^{-1}W)^H (X^{-1}W) \right\} \quad (2.5)$$

From Eqs (2.4) and (2.5), it is found that noise has an impact on the LS channel estimation. Strong noise or low SNR results in poor performance on estimation accuracy and MSE.

3. Improved LS channel estimation based on CNN

The overall flow of the proposed LS estimation module based on CNN is depicted in Figure 1. The estimation is split into two parts: the offline and online signal processing.

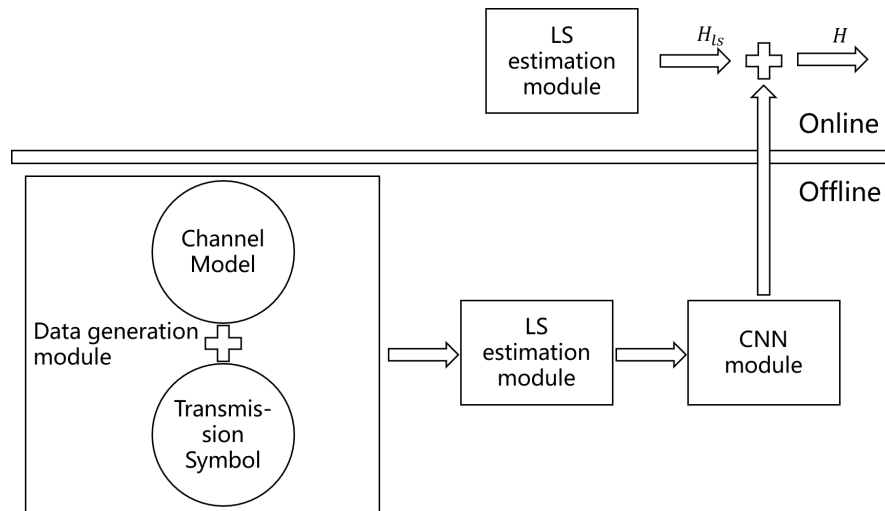


Figure 1. Channel estimation process based on CNN-LS.

3.1. Offline signal processing scheme

The offline signal processing mainly includes three modules: data generation module, LS estimation module and CNN module.

Data generation module mainly aims to generate transmission symbols $x(k)$ and true values h of the channel. In this module, a binary data string $x(k)$ is randomly generated as transmission symbols, and N_p pilots are inserted into the data string. In the simulation, the generated data is transmitted over Rayleigh fading channel and arrives at the receiver. In addition, according to Rayleigh fading channel model [13], the true value h of the channel, which is obtained through simulation, is used as the target value during the training of CNN.

In the LS estimation module, the channel estimation values $\{\tilde{h}_{p,ls}\}$ can be obtained. For OFDM systems with N_s subcarriers, we assume that there are N_p pilot symbols at the transmitter, that are uniformly inserted in comb type. At the receiver, for the m -th pilot symbol ($1 \leq k \leq N_p$), the estimation value is calculated by LS method as:

$$\tilde{h}_{p,ls}(m) = X(m)^{-1}Y(m) \quad (3.1)$$

where the pilot value at the m -th pilot and its corresponding receiving value are given as $X(m)$ and $Y(m)$, respectively. Estimation values of the pilots are processed by the interpolation calculation, and $\tilde{h}_{p,ls}$ is derived.

In the CNN module the feature distribution of the channel is simulated, and can be employed for correction of the traditional LS channel estimates. The inputs to the CNN network are the LS channel estimates $\{\tilde{h}_{p,ls}\}$, and the corresponding true values $\{h\}$ are adopted as the labels. Moreover, the network is trained in an end-to-end manner.

Based on the above analysis, the proposed method to generate the training data may be summarized as follows: First, the channel estimate $\tilde{h}_{P,Is}^{(k)}$ is calculated from the pilot frequency. Meanwhile, the true value $h^{(k)}$ of this iteration is calculated based on Rayleigh fading channel model and stored, where k is the sample index. Then the procedure repeats until it generates the whole network training set $(\{\tilde{h}_{P,Is}^{(k)}\}, \{h^{(k)}\})_{k < 16000}$ and testing set $(\{\tilde{h}_{P,Is}^{(k)}\}, \{h^{(k)}\})_{16000 \leq k \leq 20000}$. The sample sizes of the datasets can be derived from the subscript k .

Specifically, the concrete model of the proposed CNN network is shown in Figure 2.

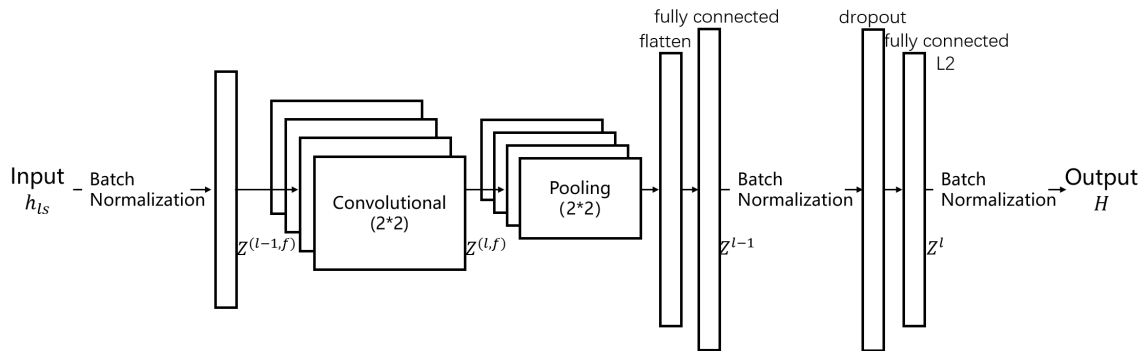


Figure 2. Structure of the proposed CNN network.

The proposed CNN network is composed of an input layer, a convolutional layer, a pooling layer, three batch normalization (BN) layers, a dropout layer and two fully connected layers, which are discussed in the following paragraphs.

1) Input layer: The neural network does not support complex numbers as inputs presently, however the channel signals are complex data. Therefore, it is necessary to process the data before being input into the network, that is, to separate the real and imaginary parts of the complex data and concatenate two parts into vectors. The channel length in this paper is 5, and the number of training data is 16,000. Consequently, after dividing the real and imaginary parts of the channel data, 16,000 sets of channel estimate values length of 10 and corresponding true values are used for training the model. The training data are fed into the neural network in batches, where the batch size is 1000 and the number of training epochs is 100.

2) Convolutional layer and pooling layer: The convolutional layer is for feature extraction of the data. Before the convolution operation, the original data are preprocessed through increasing the dimensions of data by the `tf.expand_dims` function in Tensorflow. Thus we can implement convolution of the inputs and the convolution kernels. The convolutional operation of the CNN network can be represented by:

$$z_{i,j}^{(l,f)}_{CL} = \sigma \left(\sum_{f=0}^{F_{l-1}} \sum_{s=0}^{r_w} \sum_{t=0}^{r_h} w_{s,t}^{(l,f)}_{CL} \cdot z_{i+s,j+t}^{(l-1,f)} + b^{(l,k)}_{CL} \right) \quad (3.2)$$

where $z_{i,j}^{(l,f)}_{CL}$ denotes the output of the f -th convolutional kernel in the l -th convolutional layer. $z_{i+s,j+t}^{(l-1,f)}$ is with reference to the input of the l -th convolutional layer. r_w and r_h are respectively the width and height of the convolutional kernel. The weight of the node in the convolution layer is $w_{s,t}^{(l,f)}$ and its bias is $b^{(l,k)}_{CL}$. F_{l-1} is the number of feature maps. σ is the activation function, which is ReLU. 64

convolutional kernels with size of $2 * 2$ are adopted in this paper. In addition, the convolution with a step size of 1 is performed on the input. The output of the convolutional layer is input to the pooling layer, which can be used to reduce dimension of the features and compress the data. The max pooling operation is adopted so that the maximum value within the pooling window area is set as the output value of sampling. Now assuming the output from the previous layer of size $h * w$, the maximum pooling can be expressed by:

$$Z^l(i, j)_{MP} = \max_{\substack{0 \leq m \leq r_w - 1 \\ 0 \leq n \leq r_h - 1}} (Z^{(l-1)}(i * r_w + m, j * r_h + n)) \quad (3.3)$$

where, $Z^l(i, j)_{MP}$ is the output after max pooling of the l -th pooling layer, and $0 \leq i \leq h - r_w + 1$, $0 \leq j \leq w - r_w + 1$.

3) BN layer: In deep neural networks, with the increase of the network depth, the input distribution changes after multiple linear and nonlinear transformations. However, their corresponding labels remain consistent, which leads to many problems such as decrease of learning speed and gradient vanishing. Therefore, in this paper we add a BN layer after the fully connected layer [14], which is mainly used to accelerate the training speed and improve the normalization ability of the network. The BN layer is mainly to cast the input distribution of any number of neurons to a standard normal distribution with mean 0 and variance 1, in each layer of the neural network. Due to the problem that some data cannot be activated after enforcing normalization, the transformation and reconstruction method is employed. Furthermore, two parameters γ, β are introduced and trained, so that the network can recover the feature distribution that the original network needed to learn. The computation flow of the BN layer is shown in Algorithm 1.

Algorithm 1 BN algorithm

Input: Each mini-batch: $B = x_{(1...m)}$

Input: Parameters to be learned: γ, β

Output: $\{y_i = BN_{\gamma, \beta}(x_i)\}$

- 1: $\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^m x_i$ // mini-batch mean
 - 2: $\varepsilon_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$ // mini-batch variance
 - 3: $\widehat{x}_l \leftarrow \frac{x_i - \mu_B}{\sqrt{\varepsilon_B^2 - \epsilon}}$ // normalization
 - 4: $y_i \leftarrow \beta \widehat{x}_l + \gamma \equiv BN_{\gamma, \beta}(x_i)$ // quantization and offset
-

4) Dropout layer: Since the proposed network is trained for the offline condition using the data generated by the channel model, not only training accuracy but also testing accuracy of the model needs attention. Therefore, all these requests put forward higher demand for normalization of the model. During the training of neural networks, over-fitting often occurs, that is, the model has high prediction accuracy on the training set, but low accuracy on the testing set. Accordingly, we adopt the dropout layer to avoid over-fitting, and the way it works is: Half of the feature detectors are ignored to reduce their interaction. Using this method, the model will not rely on some local features, so as to improve the generalization of the model. The network model for the dropout layer is shown in Figure 3.

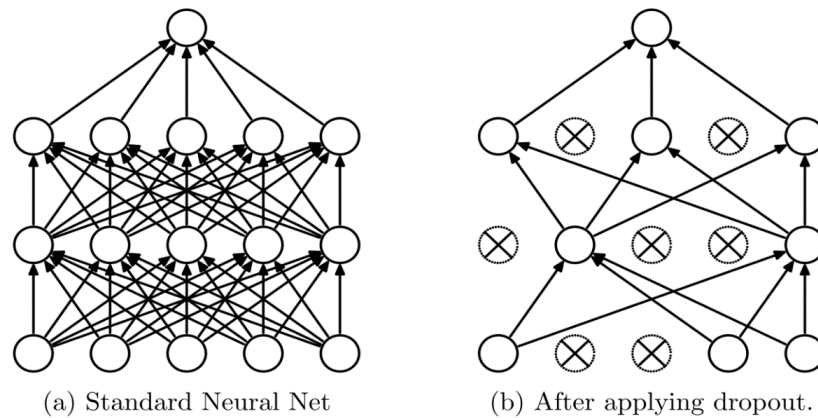


Figure 3. Network model after applying dropout.

5) Fully connected layer: The data coming from the max pooling layer are flattened, and then used as the inputs of the fully connected layer. The role of the fully connected layer is to combine the extracted features to get the output. Because only one fully connected layer will lead to nonlinear problems, two fully connected layers are used in this paper. The equation in fully connected layers can be described as:

$$Z^l_{FC} = \sigma(W^l_{FC}Z^{l-1} + b_{FC}) \quad (3.4)$$

where, Z^l denotes the output of the l -th layer of fully connected layer, W_{FC} and b_{FC} are the weights and biases of the fully connected layer nodes, respectively. The output of the last fully connected layer is a vector of length 10, which represents the channel data in the form of concatenation of its real and imaginary parts.

4. Online signal processing

Forward propagation is adopted in the online signal processing part. To begin with, a new LS estimate h_{pre} arrives at the receiver. Then it is fed into the trained neural network, in order to correct the LS estimate and obtain the corresponding new channel estimate \hat{h}_{pre} based on the CNN-LS algorithm. Assuming that $f_{est}()$ is the transformation formula for the network and θ_{est} is the parameter of the network, the corrected output can be expressed as:

$$\hat{h}_{pre} = f_{est}(h_{pre}, \theta_{est}) \quad (4.1)$$

5. Neural network training and testing

As described in Section 3.1, the CNN network is trained using the training data set $(\tilde{h}_{P.Ls}^{(k)}, h^{(k)})_{k < 16000}$ in an end-to-end manner [9], in order to optimize the weights and biases of the network. The loss function used for training is given as $L = \frac{1}{N} \sum_{n=1}^N (\tilde{h}_{P.Ls} - h)^2$, and N is related to the size of each mini-batch. The Adam optimization algorithm is employed to minimize the loss function, combining the optimal performance of AdaGrad and RMSProp algorithms. Consequently, independent adaptive learning rates

are designed for different parameters by computing the first and second moment estimations of the gradient. The size and the number of the convolution kernel are respectively set to 2 and 64, moreover, it uses the ReLU activation function. For further improving the network performance, we adopt the BN layer and dropout layer. The network training stops at 100 epochs when the loss value does not decrease, and this network is regarded as a well-trained artificial neural network. The testing data are sent to the well-trained network to validate the proposed method. The final value of the loss function in the simulation is $5.2051\text{e-}05$, which cannot be reduced any further.

In the testing phase, the feature distribution of the channel is first generated the same as the training process. For the testing set $(\tilde{h}_{P_LS}^{(k)}, h^{(k)})_{16000 \leq k \leq 20000}$, $\tilde{h}_{P_LS}^{(k)}$ are input into the trained CNN network. The predictive values $\{\tilde{h}_{P_LS}^{(k)}\}_{pre}$ are calculated, and compared with the true values $h^{(k)}$. By comparing the bit error rate (BER) of the traditional estimation methods and the improved LS estimation method based on CNN, we can demonstrate that the proposed LS estimation method has better performance.

6. Simulation and evaluation

The CNN model used for training in this paper contains one convolutional layer, one pooling layer, and two fully connected layers. The sample sizes in the training and testing sets are 16,000 and 4000, respectively. Rayleigh channel model is used in the simulation. The number of subcarriers in OFDM system is 1024, and quadrature phase shift keying (QPSK) is adopted in the modulation process. The proposed CNN method is implemented by using Python 3.5.2 and Tensorflow 1.13.1, moreover, the CPU is AMD R5 3600 and GPU is Nvidia GeForce RTX 2060.

First, we evaluate the performance of the improved LS estimation algorithm based on CNN in terms of BER over a range of SNR values, comparing to the traditional estimation algorithms. The traditional channel estimation algorithms include the LS algorithm, DFT channel estimation algorithm and LMMSE channel estimation algorithm. Here, assuming the channel estimation algorithms for the OFDM systems are based on the comb-type pilot arrangement, the number of pilot symbols is 8, and they are uniformly inserted into the subcarriers. Figure 4 shows the BER performance for the different channel estimation algorithms.

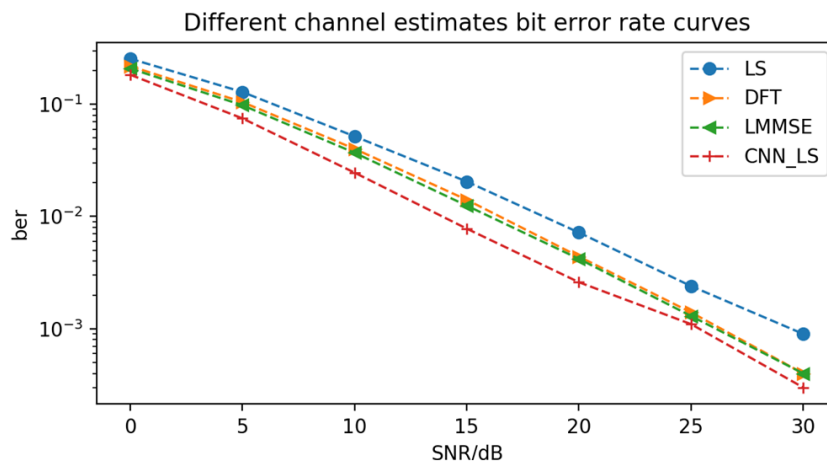


Figure 4. BER vs. SNR for different channel estimation methods.

The results in Figure 4 illustrate that compared to the traditional channel estimation algorithms, the BER performance has significantly improved in the proposed CNN_LS algorithm. Through the iterative training of CNN, the model of CNN_LS has learned the feature distribution of the channel, which can correct the existing LS estimation results to improve the system performance. Furthermore, comparing with the DFT and LMMSE channel estimation algorithms, CNN_LS also has better performance.

The performances of different channel estimation algorithms in term of normalized MSE (NMSE) varied by SNR are shown in Figure 5. We introduce the NMSE to express the error vector magnitude, in order to quantify the performance gap between the different channel estimation methods. From Figure 5 we can observe a huge performance benefit of the CNN_LS algorithm with low SNR, compared with the traditional estimation algorithms. Conversely, with high SNR, all the estimation algorithms have great improvement on NMSE performance.

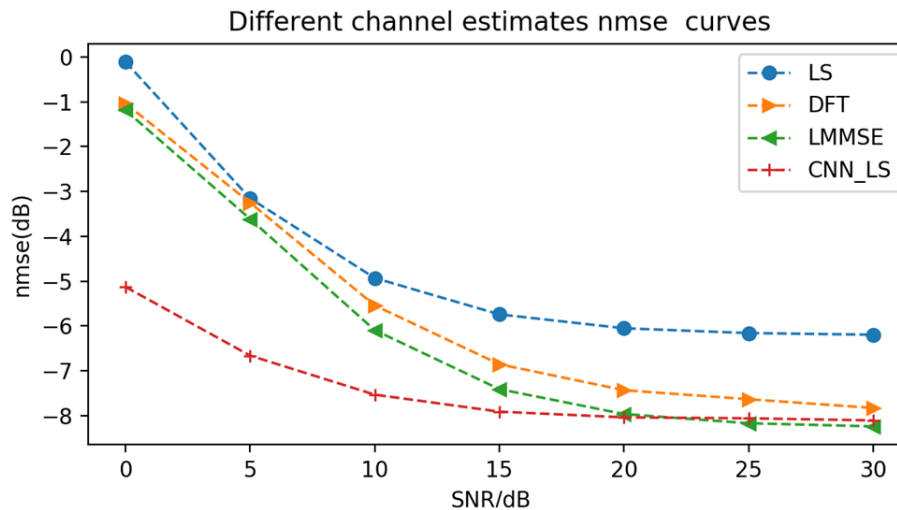


Figure 5. NMSE vs. SNR for different channel estimation methods.

As the number of pilots has an impact on channel estimation, the quality of channel estimation can be improved when the number of pilot symbols is large, however it also affects the transmission efficiency. Therefore, in this paper we investigate the effect of the number of pilot symbols for different channel estimation algorithms. With the increase of the number of pilot symbols, i.e., 8, 16, 32, and 64, the corresponding BER performance of different algorithms with an SNR value of 20 dB can be found in Figure 6.

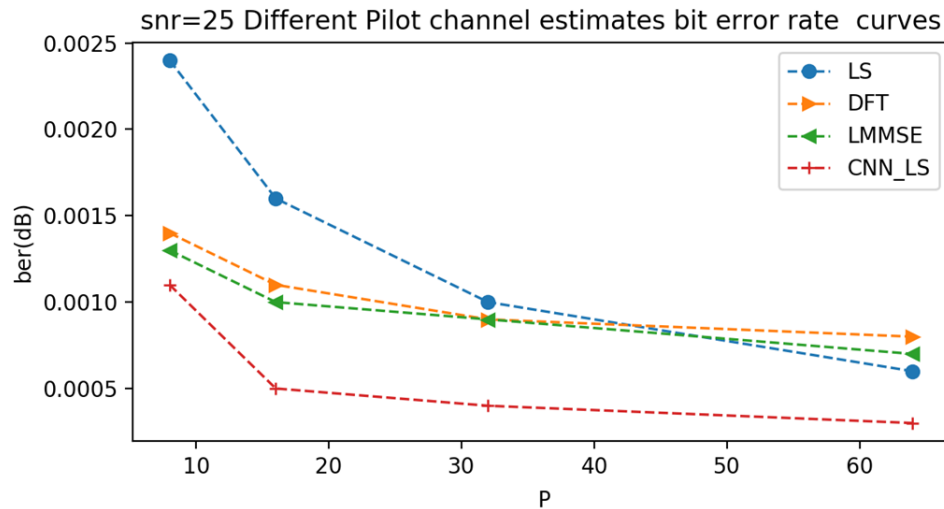


Figure 6. BER vs. No. of pilots for different channel estimation methods.

As shown in Figure 6, at high SNR, the CNN_LS method also has better performance than the traditional estimation methods. Moreover, the performance of the CNN_LS method is robust to the variable number of pilot symbols. When the number of pilot symbols is greater than 30, with the increase of pilot symbols, the BER performance of the proposed algorithm decreases slowly. In summary, the proposed algorithm can meet the requirement of BER by using less pilot symbols, saving the pilot overhead of the system.

7. Conclusions

Considering the poor performance of the traditional LS channel estimation method, we propose an improved LS channel estimation method based on CNN in this paper. This method learns the distribution features of wireless channels by designing the input-output relations, training set and testing set of the CNN network, so that the traditional LS estimate can be corrected. The simulation results show that compared with the traditional LS channel estimation algorithms, the performance of BER and NMSE was better improved by the proposed CNN_LS algorithm, and the overhead of pilot frequencies was reduced. This observation leads us to conclude that the proposed CNN_LS algorithm has better performance on BER, MSE and robustness than the traditional channel estimation methods in OFDM.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by Innovation Program of Shanghai Municipal Education Commission (No.: 2101070010E00121).

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. S. Coleri, M. Ergen, A. Puri, A. Bahai, Channel estimation techniques based on pilot arrangement in OFDM systems, *IEEE Trans. Broadcast.*, **48** (2002), 223–229. <https://doi.org/10.1109/TBC.2002.804034>
2. Y. Gong, K. B. Letaief, Low rank channel estimation for space-time coded wideband OFDM systems, in *IEEE 54th Vehicular Technology Conference. VTC Fall 2001. Proceedings (Cat. No.01CH37211)*, **2** (2001), 772–776. <https://doi.org/10.1109/VTC.2001.956875>
3. M. Morelli, U. Mengali, A comparison of pilot-aided channel estimation methods for OFDM systems, *IEEE Trans. Signal Process.*, **49** (2001), 3065–3073. <https://doi.org/10.1109/78.969514>
4. Y. Li, L. J. Cimini, N. R. Sollenberger, Robust channel estimation for OFDM systems with rapid dispersive fading channels, *IEEE Trans. Commun.*, **46** (1998), 902–915. <https://doi.org/10.1109/26.701317>
5. Y. S. Cho, J. Kim, W. Y. Yang, C. G. Kang, *MIMO-OFDM Wireless Communications with MATLAB*, Wiley Publishing, 2010. <https://doi.org/10.1002/9780470825631.ch4>
6. J. Long, Study and simulation on channel estimate algorithm in OFDM system, *Commun. Technol.*, **41** (2008), 7–8. <https://doi.org/10.3969/j.issn.1002-0802.2008.10.003>
7. D. Wang, Z. Mei, J. Liang, J. Liu, An improved channel estimation algorithm based on WD-DDA in OFDM system, *Mobile Inf. Syst.*, **2021** (2021), 6540923. <https://doi.org/10.1155/2021/6540923>
8. Y. Li, C. Tao, G. Secogranados, A. Mezghani, A. L. Swindlehurst, L. Liu, Channel estimation and performance analysis of One-Bit massive mimo systems, *IEEE Trans. Signal Process.*, **65** (2017), 4075–4089. <https://doi.org/10.1109/TSP.2017.2706179>
9. H. He, C. K. Wen, J. Shi, G. Y. Li, Deep learning-based channel estimation for beamspace mmwave massive mimo systems, *IEEE Wireless Commun. Lett.*, **7** (2018), 852–855. <https://doi.org/10.1109/LWC.2018.2832128>
10. M. Soltani, V. Pourahmadi, A. Mirzaei, H. Sheikhzadeh, Deep learning-based channel estimation, *IEEE Commun. Lett.*, **23** (2019), 652–655. <https://doi.org/10.1109/LCOMM.2019.2898944>
11. H. Ye, G. Y. Li, B. H. Juang, Power of deep learning for channel estimation and signal detection in OFDM systems, *IEEE Wireless Commun. Lett.*, **7** (2017), 114–117. <https://doi.org/10.1109/LWC.2017.2757490>
12. A. S. M. Mohammed, A. I. A. Taman, A. M. Hassan, A. Zekry, Deep learning channel estimation for OFDM 5G systems with different channel models, *Wireless Pers. Commun.*, **128** (2023), 2891–2912. <https://doi.org/10.1007/s11277-022-10077-6>
13. S. A. Fechtel, A novel approach to modeling and efficient simulation of frequency-selective fading radio channels, *IEEE J. Sel. Areas Commun.*, **11** (1993), 422–431. <https://doi.org/10.1109/49.219555>

-
14. S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in *Proceedings of the 32nd International Conference on Machine Learning*, **37** (2015), 448–456. Available from: <http://proceedings.mlr.press/v37/ioffe15.html>.



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)