Learning via Denoising Autoencoder on 5G NR Phase Noise Estimation

Mu-sheng Lin
Samsung Semiconductor Inc
San Diego, USA
musheng.lin@samsung.com

Hyukjoon Kwon
Samsung Semiconductor Inc
San Diego, USA
hyukjoon.k@samsung.com

***Abstract*—In this paper, on phase noise of 5G NR mmWave systems, we propose a learning-based common phase error (CPE) estimation algorithm based on the denoising autoencoder serving as a nonlinear filter on the existing CPE estimators. The proposed algorithm learns the low dimension manifold of phase noise distributions with high probability. Traditional CPE methods have limitation when the time domain pilots are few, which causes degraded system performance with CPE interpolation. Besides accurate CPE estimation, the proposed method is more robust to various FR2 channels, Doppler effects, and the numerologies. Simulation results show that block error rate (BLER) could be improved up to 1.43 dB on SNR.**

***Index Terms*—Common Phase Error, learning, denoising autoencoder, neural network, 5G NR, FR2**

I. Introduction

Cellular systems such as 4G and 5G wireless systems are built based on the orthogonal frequency division multiplexing (OFDM) technique [2]. One of the main benefits of OFDM systems is that subcarriers are orthogonal to each other and could save bandwidth due to overlapping in the frequency domain, compared to the traditional FDM systems. However, the main challenging also comes from that OFDM is sensitive to carrier frequency offsets [3] and random phase offsets [4]. These impairments equivalently affect the frequency samples deviating from the peak of the sinc function (assume rectangular window) [3] and cause inter-carrier-interference (ICI) due to non-orthogonality among subcarriers. Phase noise, the RF impairment on frequency synthesizers (more precisely on voltage control oscillators) [5], is one of the main sources to generate random phase offsets. The effect could be divided into 1) common phase error (CPE) and 2) inter-carrier-interference (ICI) [6]. With phase noise, the degraded BLER is significant, especially on higher order modulation schemes such as 64 QAM under mmWave scenarios in 5G NR systems and the impairment needs to be taken care of before channel estimation and symbol detection. The focus of this paper is on CPE. The ICI could be solved by deliberate cancellation schemes such as [7]. However, these schemes require knowledge of transmitted signal by pilot-aided methods or decision/decoding feedbacks, and will not be covered in this paper. To estimate the CPE, one simple scheme is to average over observations among subcarriers in the frequency domain to form the phase estimator similar to frequency offset estimators [3].

Deep learning techniques have brought a lot of attention in the past years [8], [9] and outperform traditional machine learning methods with significant margin [10]. Recently, successful attempts to apply these supervised learning techniques to 5G NR systems in wireless communication fields are demonstrated: channel estimation [11], multiple-input and multiple-output (MIMO) symbol detection [12], channel quality indicator (CQI) and rank indicator (RI) estimation [13] and Doppler spread estimaiton [14]. The autoencoder [15] is another type of neural networks that could learn the data distributions. In this paper, its variant, the denoising autoencoder, is suggested as a nonlinear filter from signal processing perspective such that the output of the neural network is the uncorrupted version of the target signal.

While the CPE estimation is straightforward, it has some limitation, especially when the reference signal (RS), such as the demodulation reference signal (DMRS) and the phase tracking reference signal (PTRS) of 5G NR, are not enough to obtain reliable CPE estimators in a slot. In particular, when the OFDM symbols with RS are not dense enough in the time domain while channel estimation needs one CPE per OFDM symbol to perform time domain filtering. In this case, CPE interpolation is inevitable. The simple solution is to proceed with linear interpolation in the time domain. However, the system performance could be degraded due to inaccurate channel estimation under influence of large phase noise or higher Doppler effects.

In this paper, we propose one learning-based CPE algorithm via the denoising autoencoder to enhance the interpolated CPE results, which improves the system performance with substantial signal to noise ratio (SNR) gains. To design the learning-based CPE estimator via the denoising autoencoder, the basic question could be the behavior of the denoising autoencoder learned from the data distributions, which is described in great details in [16]. The main concept is that the denosing autoencoder learns the low dimension manifold [1] of the data distributions from the noisy data by means of denoising regularization. In our case, the neural network learns the average CPE distributions among slots and hence acts as a nonlinear filter to generate better CPE estimates closer to

[1] Here we follow the convention in [15] that the term ”manifold” in machine learning loosely means the data set could be approximately described by a small number of dimensions embedded in higher dimensional space.

correct CPEs. Also the proposed learning-based CPE algorithm shows robustness on FR2 channels, different Doppler effects, and the numerologies. The denoising capability could be explained by the vector filed behaviors learned during training [16]. More details will be presented in the following sections.

Our contributions are summarized below:

- We propose to learn the better CPE estimations with data distribution governed by phase noise via the denoising autoencoder. The idea is to learn the average CPE distributions among slots through training such that the noisy or interpolated CPEs are corrected to follow the low dimension manifold of the phase noise generating mechanism, which provides substantial BLER improvement to the 5G NR system. To the best of our knowledge, this is the first time that the denosing autoencoder is applied to the phase noise estimation problem.
- The proposed algorithm serves as a paradigm to generalize the applications of the denoising autoencoder as the nonlinear filter to enhance the traditional wireless communication algorithms. The algorithm is applicable when the embedded generating mechanism could be learned from the neural netwrok through training on higher dimension observations. This concept bridges the gap between signal processing field and the machine learning techniques on wireless communication.
- Simulation results support that the new algorithm is robust to FR2 channel conditions, Doppler effects and numerologies, which is a unified approach to the highly flexible 5G mmWave design under different scenarios.

II. System Model

A. Effects of Phase Noise

The phase noise is from the RF synthesizer, which manifests the wider power spectrum density in contrast to the impulse shape of the ideal oscillator. The phase noise results in multiplicative random phases in the time domain and hence, the received signal in the time domain could be modeled as following [17]:

$$r_m(n) = s_m(n) \otimes h_m(n) e^{j\phi_m(n)} + w_m(n) \tag{1}$$

where m is the OFDM symbol index and n is the time sample index in one FFT period. Also $s_m(n)$ is the transmitted signal in the time domain, h is the wireless channel, w denotes white Gaussian noise, and $e^{j\phi_m(n)}$ is the random phase effects caused by phase noise. $\otimes$ denotes the circular convolution. The frequency domain expression of eq. (1) could be obtained [17]

$$R_m(k) = \sum_{i=-N/2}^{\frac{N}{2}-1} J_m(i) S_m(k-i)_{modN} H_m(k-i)_{modN} + W_m(k) \tag{2}$$

where $S_m(k)$ is the transmitted data symbol at subcarrier k within the mth OFDM symbol and the channel response for the subcarrier k is denoted as H_k. The term $J_m(k)$ represents the phase noise effect, which could be obtained from

$$J_m(k) = \frac{1}{N} \sum_{i=-N/2}^{\frac{N}{2}-1} e^{j\phi_m(n)} e^{j2\pi nk/N} \tag{3}$$

and $J_m(0)$ is the CPE while $J_m(k)$'s for $k \neq 0$ are ICIs.

From eq. (3), $J_m(0)$ is just discrete fourier transform (DFT) of the vector $\{e^{j\phi_m(n)}\}$ at $k = 0$ and is the common phase factor to all the subcarriers. From eq. (2), the received signal at subcarrier k will be interfered by data symbols at the neighbor subcarriers and their corresponding channel frequency responses weighted by $\{J_m(i)\}$ from the DFT of the vector $\{e^{j\phi_m(n)}\}$ at $k = i$. Refer to [17] for more details on phase noise effects.

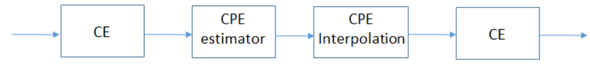


Fig. 1: Channel estimation dataflow with CPE compensation.

B. CPE Estimator and Interpolation

The phase noise effect could be divided into 1) CPE and 2) ICI. Assume that the ICI effect is not significant. Then eq. (2) could be approximated as

$$R_m(k) \approx J_m(0) H_m(k) S_m(k) + W_m(k) \tag{4}$$

which means that the estimate of the CPE, $J_m(0)$ is obtained from the observations among subcarriers in the frequency domain at the OFDM symbol m. The CPEs to channel estimation could be obtained from the channel estimation results for DMRS symbols or from the despreading outputs of PTRS symbols such as the proposed in [17]. The CPEs are sent to the channel estimation (CE) module as shown in Fig. 1. The interpolation actually has an extra role on generating CPEs at OFDM symbols without RS as corrupted input features from the denoising autoenoder perspective.

III. Learning-based CPE Algorithm

In this section, we propose a learning based CPE algorithm via the denoising autoencoder that utilize a neural network to “refine” the existing CPE estimates. By “refine” we mean the algorithm serves as a nonlinear filter such that the outputs of the neural netwrok are uncorrupted version of the targeting signal.

A. Denoising Autoencoder

In [15], the autoencoder is defined as one neural network that tries to ”copy” the input, i, to the outputs, o through the hidden layer h (by the internal representation or a code) but only keeps certain resemblance structure of inputs to the training data with restrictions to avoid learning the identity map. Precisely, the process could be described with a pair of encoder and decoder functions (e, f) such that a code (at the hidden layer) of $h = e(x)$ is constructed with restrictions and the loss is minimized on the Loss function L with respect to certain criteria,

$$loss = L(x, d(e(x))) \tag{5}$$

where $o = x$ and $i = x$ express that inputs and outputs are the same for the autoencoder. Common choices of L are cross-entropy or mean square error (MSE).

Typically based on the number of nodes in the hidden layer, there are two types of autoencoders: 1) undercomplete autoencoders and 2) overcomplete autoencoders. When the number of hidden layer nodes is smaller than the number of nodes at the input layer, the autoencoder is called undercomplete. The purpose is to learn the latent variables to represent the label data distributions. On the other hand, for the case that hidden layer nodes are larger or equal to the input features, it's named as an overcomplete autoencoder. The goal of overcomplete auto-encoder is usually to learn/distinguish the samples from the label data distributions by means of the extra capacity of larger hidden layer nodes. Among overcomplete autoencoders, particular interest is on the denoising autoencoder for our study, first mentioned by Vincent [18]. With the concept of denoising autoencoder, eq. (5) could be rewritten as

$$loss = L(x, d(e(\tilde{x}))) \tag{6}$$

where $o = x$ are the data and $i = \tilde{x}$ denotes the inputs are corrupted version of data for the denoising autoencoder.

B. Motivation

From the manifold learning perspective, the denoising autoencoder model could learn the low dimension manifold from the noisy data to the correct labels [18]. In wireless communication, this property could be useful to enhance or refine the traditional existing algorithms by comparing the noisy/corrupted inputs and the output labels to learn the intrinsic property of the target parameters that dominates the data distributions. A case in point of our study is as follows: CPEs generated from the phase noise are correlated among OFDM symbols and could be viewed as data distributions following certain low dimension manifold: The phase noise is only one dimension and the observations are complex numbers with the dimension of the size of FFT, referred to eq. (4). Then the observations are mapped to CPEs in Euclidean space of R^2 by the CPE estimator. As shown in Fig. 2, the possible trajectories of CPEs are controlled by phase noise with correlation constrained by the generating process. Here we assume that the phase noise gereating process is one Wiener process and could be modeled by Lorentzian power spectrum density (PSD) [4]. In Fig. 2, each time instant t_m represents one OFDM symbol and the CPE at the consecutive time instant t_{m+1} depends on the previous CPE at time instant t_m. The blue circle represents the bound of noisy data distributions such that noisy data are within Euclidean distance of σ from the uncorrupted data. With the correlation constrained by the Lorentzian PSD, the CPEs follow the phase noise distributions in a certain range with high probability. Notice that since the CPEs are continuous random variables, the paths are actually uncountable. We want to learn a high probability path such that within one slot consisting of fourteen OFDM symbols, the CPEs are concentrated on the low dimension manifold of phase noise distributions.

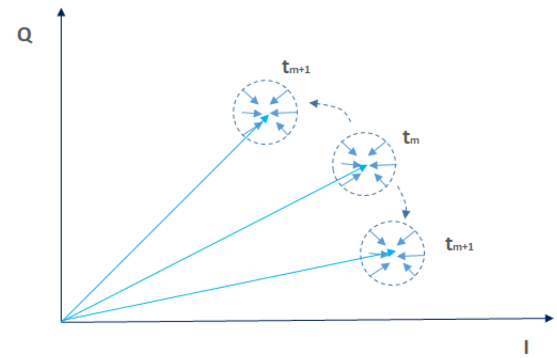


Fig. 2: Possible CPE trajectories controlled by the phase noise with time.

Proposition 1. *Given that phase noise generating process follows the Wiener process with Lorentzian PSD, the CPEs are concentrated near the low dimension manifold of phase nosie distributions with high probability embedded in* R^2

Proof. Sketch of the proof is as follows: The phase noise is generated by (IIR) filtering of white Gaussian random variables with Lorentzian PSD. Hence, the CPEs are a correlated Gaussian random vector. Within one slot, if we connect all the mean values of each element in the Gaussian random vector, the CPEs are concentrated near the constructed path with high probability. □

To understand the learning behavior of denoising autoencoder, the authors in [16] maintain that when the noise variance is small, with Taylor expansion, the optimal reconstruction function, $r^*_\sigma(x)$ ($d(e(x))$ in our context) behaves asymptotically as $\sigma \to 0$ as (Theorem 1 in [16])

$$r^*_\sigma = x + \sigma \frac{\partial log(p(x))}{\partial x} + o(\sigma^2) \tag{7}$$

where x is one data observation without noise, r^*_σ describes the denoising autoencoder and $p(x)$ is the data distribution without noise. The eq. (7) means that the corrupted inputs could be corrected by the vector field near the data distribution of the low dimension manifold $p(x)$. For our case of study, the high probability sample path of CPEs in one slot constructed from proposition 1 is just one possible $p(x)$ for a particular slot and among slots there are many of these high probability paths. Hence we have proposition 2 to learn the average path among slots with high probability:

Proposition 2. *Denoising autoencoder could approximate the CPEs concentrated near the low dimension manifold of phase noise distributions with high probability*

Proof. Sketch of the proof is as follows: From proposition 1, within one slot, the CPEs concentrates near the constructed path in low dimension manifold with high probability. Then we take average of the constructed paths with large numbers of slots. □

With the help of proposition 2, we could now build our learning-based CPE algorithm based on the denoising autoencoder. As shown in Fig. 2, the noisy data distributed within the open ball of radius σ could be corrected by the vector field described in eq. (7). Some technical details related to this approach are as follow: Firstly, in previous section, the interpolation process fills the gap between the RS's in the time domain. Those inaccurate interpolated CPEs could be viewed as corrupted input features from the aspect of the denoising autoencoder. That is, if the interpolation error is not large (limited by the σ in eq. (7)), then the learning-based CPE algorithm is able to correct the CPEs following the average phase noise path, even though the observations of these CPEs on the non-RS symbols are not accessible. Notice that the interpolation error could be controlled by the interpolation method and the density of RS symbol in the slot. Secondly, the original CPE estimator outputs could be modeled as correct CPEs plus Gaussian noise and the estimation error depends on the number of observations in the frequency domain and the receiver SNR level. Although proposition 2 could only approximate the average phase noise path with high probability, not the true paths, simulation results support that the BLER could be greatly improved with the proposed learning-based CPE algorithm.

C. Structure of the learning-based CPE Algorithm

The data flow of the proposed learning-based CPE algorithm is shown in Fig. 3. The CPEs on RS OFDM symbols (either on DMRSs or PTRSs) are estimated and then CPEs on non-RS OFDM symbols are interpolated. Notice that the interpolation methods could be any algorithm with bounded interpolation errors satisfying the eq. (7), which also depends on RS density in the time domain. The following block is a typical denoising autoencoder built by the neural network. The outputs of the neural network are used in channel estimation (CE). Fig. 4 illustrates the structure of neural network consisting of one input layer, one hidden layer and one output layer. At each hidden layer node, the sigmoid function is applied as the activation function, which provides the nonlinearity to the network. During the training process, the interpolated CPEs for all OFDM symbols serve as the input features to the neural network while the uncorrupted CPEs are obtained from the phase noise models. The mean square error (MSE) is the optimization criterion. The neural network is an overcomplete autoencoder so the number of hidden layer nodes is larger than the input features. The natural choice for regression problem is to have the output activation function set as the linear function. In such a case, no output normalization is needed.

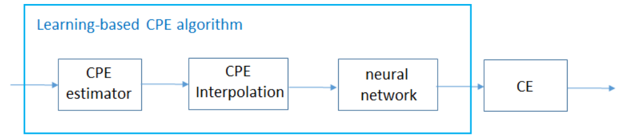


Fig. 3: Data flow of learning-based CPE algorithm and CE.

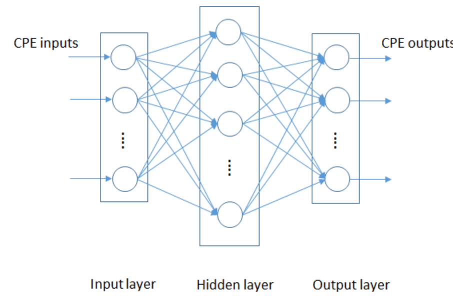


Fig. 4: Structure of neural network inside learning-based CPE algorithm.

IV. Simulation Results

In this section, the simulation results are presented. We consider FR2 scenarios with different channels, Doppler effects (the UE with different speeds), and the 5G NR numerologies [1].

A. Parameters of the Neural Network

The neural network in our implementation has only one hidden layer with sixteen nodes. The activation function at the hidden layer is sigmoid function. The loss function is mean square error (MSE). Since there are fourteen OFDM symbols for normal cyclic prefix (NCP) cases in one slot for 5G NR systems, the number of nodes for the inputs and outputs are both fourteen.

TABLE I: Neural Network Settings

Parameter	Settings
input features	14
hidden layer nodes	16
hidden layer activation	sigmoid
output nodes	14
output layer activation	linear
loss function	MSE

B. Parameters of Simulation Environment

The considered channels are in FR2 with SCS = 60kHz or 120 kHz at carrier frequency of 28 GHz. The bandwidth is set for 200 MHz (132 RBs). The PTRS time density is 4 to show the superiority of proposed learning-based CPE algorithm over plain linear interpolation. The phase noise level of integrated phase noise (IPN) is -30 dB at 28 GHz. Particularly, the phase noise is generated by IIR filtering of white Gaussian vectors. To show significant impact of phase noise, 64 QAM is selected as the modulation order. The number of MIMO stream layers is two, which is the maximum allowable choice for FR2 in 5G NR systems.

TABLE II: Simulation Settings

Parameter	Settings
Carrier Frequency	28GHz
Channel	TDL/CDL/EPA
UE Speed	3 km/hr
SCS	60kHz/120 kHz
Phase noise level	IPN = -30 dB
Layer	2
Modulation	64 QAM
PTRS time density	4
PTRS frequency density	2

C. Training

The proposed neural network is robust with respect to different channel conditions. Specifically, the network is trained by samples from TDL-A channel only but the same network could work for all TDL/CDL/EPA channels. However, the number of dmrs symbols in one slot does have some impact on the network. Denote as dmrs [1,3,n] for DMRS configuraiton 1, 1 front-loaded dmrs symbol at symbol 3 and n representing the number of add-on dmrs symbols as described in [1]. In the training process, we collect one-third samples from dmrs [1,3,0], dmrs [1,3,1], and dmrs [1,3,2] each. On SNR range, the samples are from SNR = 20 dB up to SNR = 40 dB with 2 dB step size.

D. BLER Performance

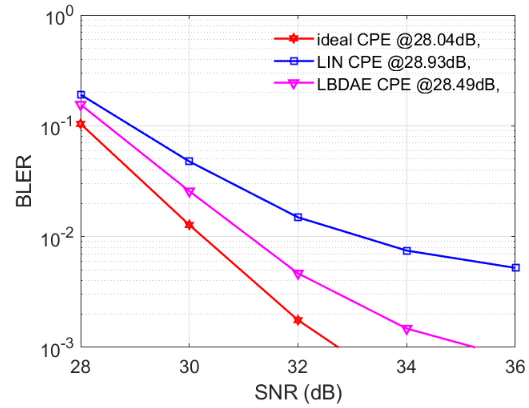
The proposed learning-based CPE estimator shows improvement on SNR, compared to the linear interpolation CPE estimator. In the following figures, ideal CPE denotes the ideal frequency domain CPE serving as genie estimator and is the lower bound of any CPE estimators. Linear interpolation CPE is the baseline to compare with (denoted as LIN CPE on plots). The learning-based denoising autoencoder CPEs result from passing linear interpolation CPEs through the neural network (denoted as LBDAE on plots). Simulation results show significant SNR gain with proposed method and the results are summarized in Table III. In addition, the trained single network from SCS = 120 kHz could work well for SCS = 60 kHz (not shown here).

TABLE III: SNRs at BLER = 10 % for different channel.

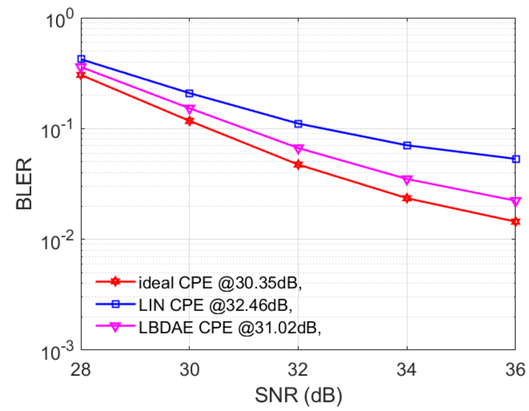
Channel	proposed (dB)	linear (dB)	SNR Gain (dB)
EPA-5	28.49	28.93	0.44
EPA-78	31.02	32.46	1.43
TDL-A-78	29.35	30.22	0.87
TDL-B-78	28.94	29.70	0.76
TDL-C-78	29.36	30.22	0.86
CDL-A-78	24.52	24.78	0.26
CDL-B-78	24.46	25.02	0.56

V. Conclusion

This paper proposes a learning-based CPE algorithm via the denoising autoencoder, which learns to approximate the correct CPEs governed by the low dimension manifold of phase noise distributions. The algorithm is robust to different FR2 channels, Doppler effects and various 5G NR numerologies. Simulation results demonstrate that the proposed learning-based CPE method could achieve substantial SNR gains on BLER performance compared to the existing linear interpolation CPE estimator.



(a) $f_d = 5$



(b) $f_d = 78$

Fig. 5: EPA channel

References

[1] 3GPP, "NR; Physical layer procedures for data," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.211-214, version 16.0.0.

[2] S. Weinstein and P. Ebert, "Data transmission by frequency-division multiplexing using the discrete fourier transform," *IEEE Transactions on Communication Technology*, vol. 19, no. 5, pp. 628–634, 1971.

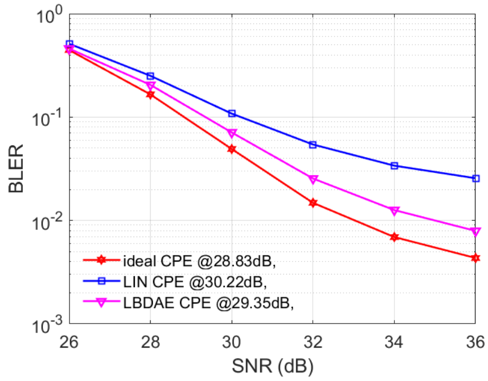
[3] P. H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *IEEE Transactions on Communications*, vol. 42, no. 10, pp. 2908–2914, 1994.

[4] T. Pollet, M. Van Bladel, and M. Moeneclaey, "Ber sensitivity of ofdm systems to carrier frequency offset and wiener phase noise," *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 191–193, 1995.

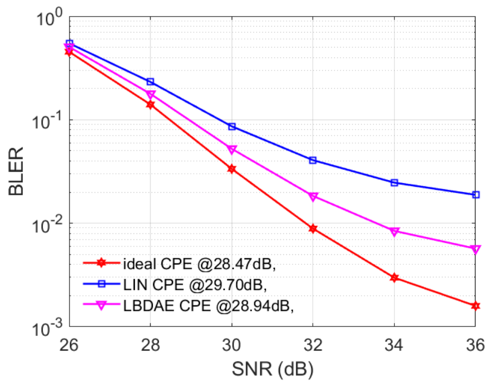
[5] B. Razzvi, *RF Microelectronics*. Pearson, 2nd edition, 2011.

[6] P. Robertson and S. Kaiser, "Analysis of the effects of phase-noise in orthogonal frequency division multiplex (ofdm) systems," in *Proceedings IEEE International Conference on Communications ICC '95*, vol. 3, 1995, pp. 1652–1657 vol.3.

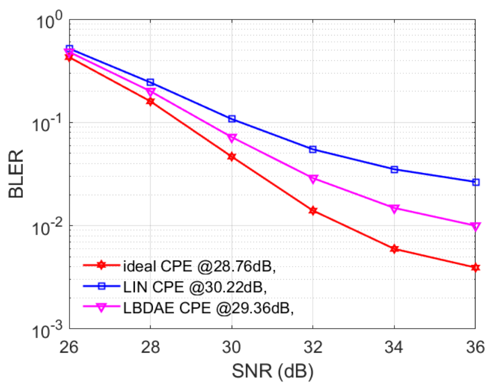
[7] P. Rabiei, W. Namgoong, and N. Al-Dhahir, "A non-iterative technique for phase noise ici mitigation in packet-based ofdm systems," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5945–5950, 2010.



(a) TDL-A

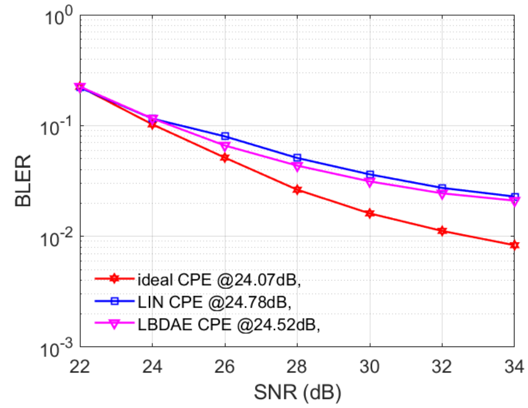


(b) TDL-B

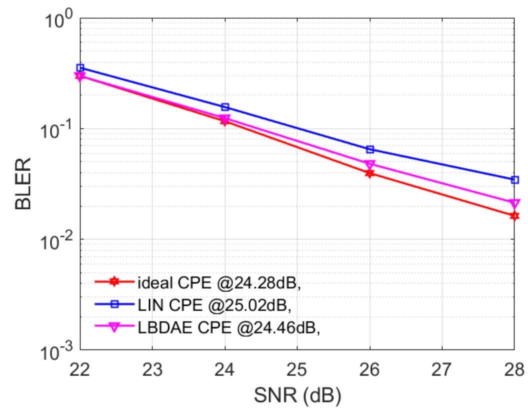


(c) TDL-C

Fig. 6: TDL channel, dmrs [1,3,1], $f_d = 78$



(a) CDL-A



(b) CDL-B

Fig. 7: CDL channel, dmrs [1,3,0], $f_d = 78$

[8] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 07 2006.

[9] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[11] H. Kwon, "Bundle-based channel estimation for 5g nr with supervised learning," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[12] S. Chaudhari, H. Kwon, and K. B. Song, "Mimo detector selection for multiple high-order modulations with unified neural network," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[13] A. Baknina and H. Kwon, "Adaptive cqi and ri estimation for 5g nr: A shallow reinforcement learning approach," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[14] Y. Ding and H. Kwon, "Doppler spread estimation for 5g nr with supervised learning," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–7.

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

[16] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," *Jounral of Machine Learning Research*, vol. 15, no. 1, pp. 3743–3773, 2014.

[17] Songping Wu and Y. Bar-Ness, "A phase noise suppression algorithm for ofdm-based wlans," *IEEE Communications Letters*, vol. 6, no. 12, pp. 535–537, 2002.

[18] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 1096–1103.