Received 30 August 2023; revised 24 September 2023; accepted 4 October 2023. Date of publication 12 October 2023; date of current version 25 October 2023.

Digital Object Identifier 10.1109/OJCOMS.2023.3322989

Deep Learning-Based End-to-End Design for OFDM Systems With Hardware Impairments

CHENG-YU WU[1], YU-KAI LIN[1], CHUN-KUAN WU[1], AND CHIA-HAN LEE[1,2] (Member, IEEE)

[1]Institute of Communications Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan
[2]Department of Electronics and Electrical Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan

CORRESPONDING AUTHOR: C.-H. LEE (e-mail: chiahan@nycu.edu.tw)

This work was supported in part by Qualcomm through the Taiwan University Collaboration Project and in part by the National Science and Technology Council (NSTC), Taiwan, under Project 110-2221-E-A49-022-MY3 and Project 112-2221-E-A49-078-MY3.

ABSTRACT Orthogonal frequency-division multiplexing (OFDM) is a key technology for cellular and Wi-Fi systems, but its performance may be degraded by hardware impairments. Existing works focus mostly on single hardware impairment in OFDM systems, without considering the joint effect of hardware impairments on the entire system. In this paper, hardware impairments including nonlinear power amplification, clipping, in-phase/quadrature-phase (IQ) imbalance, phase noise, carrier frequency offset, and sampling clock offset in OFDM systems are simultaneously considered. We propose end-to-end deep learning-based designs, which jointly optimize transmitter and receiver, to effectively mitigate the performance loss due to hardware impairments. For single-antenna systems and 2×2 multiple-input and multiple-output (MIMO) systems, the proposed design featuring the dense layer neural network (DLNN) significantly outperforms traditional impairment-mitigating methods under both the additive white Gaussian noise (AWGN) channel and the Rayleigh fading channel. Meanwhile, the complexity of the proposed scheme is six times smaller. For 2×4 MIMO systems, the proposed design featuring the residual dense convolution dense neural network (ResNet-DCDNN) outperforms the traditional methods by a large margin. Additionally, transfer learning is applied to effectively address the issue of time-varying impairment levels.

INDEX TERMS Deep learning, orthogonal frequency-division multiplexing (OFDM), hardware impairments, end-to-end design, multiple antennas, transfer learning.

I. INTRODUCTION

THE ORTHOGONAL frequency-division multiplexing (OFDM) technology has been widely adopted for wireless communications, including Wi-Fi and cellular systems [1], [2]. However, the performance of OFDM systems is sensitive to the front-end non-idealities [3], [4], [5], which may cause problems such as signal nonlinearity, clipping, in-phase/quadrature-phase (IQ) imbalance, phase noise (PN), carrier frequency offset (CFO), and sampling clock offset (SCO). Algorithms have been proposed to compensate individual hardware impairments such as CFO [6], [7], [8], IQ imbalance [9], [10], phase noise [11], [12], [13], clipping [14], [15], and SCO [16], and deep learning-based approaches have also been proposed to tackle clipping [17], phase noise [18], and CFO [19] issues. However, an OFDM system design that takes multiple hardware impairments into account has never been proposed, as far as we know. Note that some previous works focus on the analysis or resource allocation of systems with hardware impairments instead of the mitigation of hardware impairments [20].

In this paper, we simultaneously consider all the aforementioned hardware impairments, and propose end-to-end joint transmitter-receiver designs to combat the hardware impairment effects. For single-input and single-output (SISO) and 2×2 multiple-input and multiple-output (MIMO) systems, we propose a design featuring the dense layer neural network (DLNN), which significantly outperforms the traditional schemes in terms of symbol error rate (SER) with lower complexity under both the additive white Gaussian noise

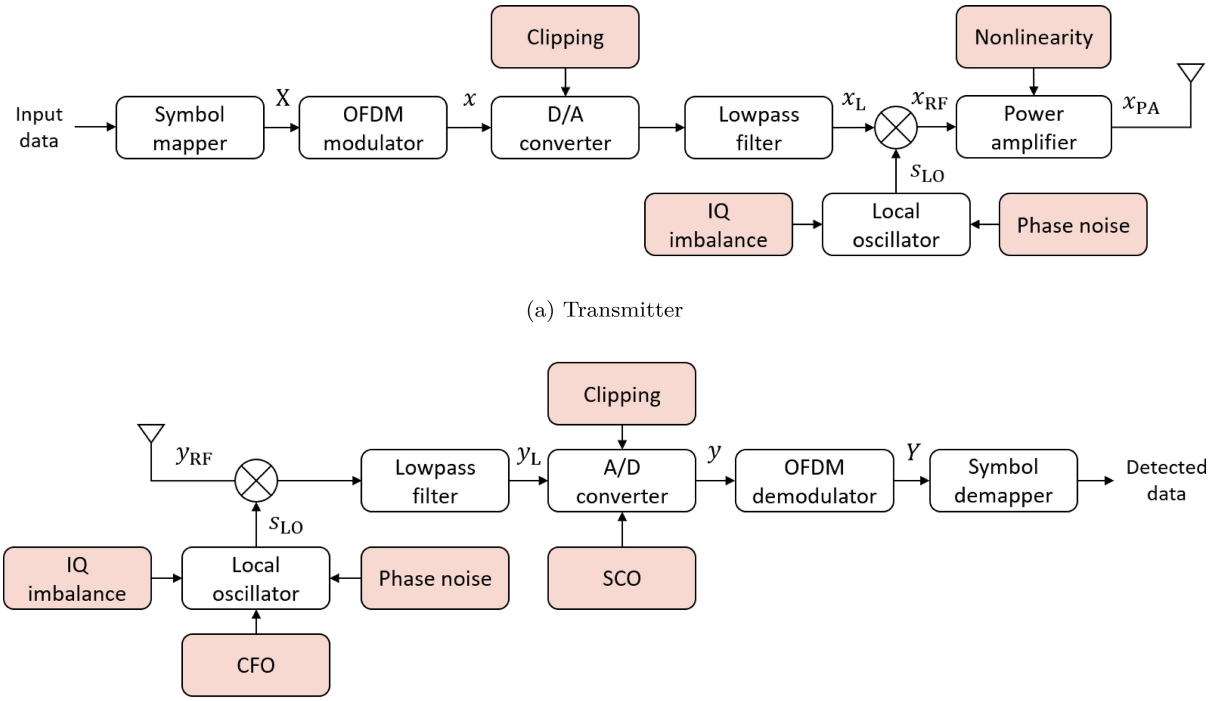


FIGURE 1. Hardware impairments considered in this paper.

(AWGN) channel and the Rayleigh fading channel. For 2×4 MIMO systems, we propose a design featuring the residual dense convolution dense neural network (ResNet-DCDNN), which achieves an SER much lower than the traditional methods. The proposed designs are shown to perform well for both cases of same and different impairments at different antennas. Additionally, transfer learning is applied to effectively address the issue of time-varying impairment levels. To the best of our knowledge, this is the first work that successfully mitigates multiple impairment effects simultaneously.

The rest of this paper is organized as follows. In Section II, the system model and the problem formulation are elaborated. In Section III, related works to be compared are reviewed. Then, in Section IV, the deep learning-based end-to-end designs for combating impairments are proposed. The simulation results are presented in Section V. Finally, Section VI concludes this paper.

II. SYSTEM MODEL

A. OFDM SYSTEM

We consider an OFDM system with hardware impairments at both the transmitter (TX) and the receiver (RX), as shown in FIGURE 1. The transmitter consists of symbol mapper, OFDM modulator, digital-to-analog (D/A) converter, lowpass filter, local oscillator, and power amplifier (PA), while the receiver consists of the corresponding functions in the reversed order, i.e., local oscillator, lowpass filter, analog-to-digital (A/D) converter, OFDM demodulator, and symbol demapper. An OFDM modulator typically performs serial-to-parallel (S/P) conversion, inverse fast Fourier transform (IFFT), and cyclic prefix (CP) insertion, and an OFDM demodulator performs CP removal, fast Fourier transform (FFT), and parallel-to-serial (P/S) conversion.

Let us assume a deterministic, complex lowpass filtered signal $x_{\mathrm{L}}(t) = x_{\mathrm{I}}(t) + jx_{\mathrm{Q}}(t)$ which is used to modulate the radio frequency (RF) carrier at frequency f_0. When the transmitter is ideal, i.e., without hardware impairments, the passband signal $x_{\mathrm{RF}}(t)$ is given by

$$\begin{aligned} x_{\mathrm{RF}}(t) &= \Re\left[x_{\mathrm{L}}(t)\mathrm{e}^{j\omega_0 t}\right] \\ &= x_{\mathrm{I}}(t)\cos(2\pi f_0 t) - x_{\mathrm{Q}}(t)\sin(2\pi f_0 t), \end{aligned} \tag{1}$$

where $x_I(t)$ and $x_{\mathrm{Q}}(t)$ are respectively the in-phase and quadrature-phase elements of the ideal complex baseband signal $x_{\mathrm{L}}(t)$, and $\omega_0 = 2\pi f_0$. The ideal receiver recovers $x_{\mathrm{L}}(t)$ from $x_{\mathrm{RF}}(t)$ by multiplying $x_{\mathrm{RF}}(t)$ with $\mathrm{e}^{-j\omega_0 t}$, followed by the filtering at frequency $-2f_0$.

The hardware impairments considered in this paper include nonlinear power amplification, clipping, IQ imbalance, phase noise, CFO, and SCO at both the transmitter and the receiver. The impairments are elaborated in detail in the following subsections. For ease of description, in the following introduction of impairments, $s(t)$ and $d(t)$ represent the ideal input signal and the distorted output signal, respectively.

B. NONLINEAR POWER AMPLIFICATION

Most power amplifiers in the transmitter can be driven into saturation. The nonlinear behavior of power amplifiers can be modeled in general by a power series expansion with order K and corresponding coefficient a_k, i.e., the output $d(t)$ is given by [21]

$$d(t) = \sum_{k=1}^{K} a_k x_{\mathrm{RF}}^k(t), \tag{2}$$

where $x_{\mathrm{RF}}^k(t)$ represents the kth power of the ideal passband signal $x_{\mathrm{RF}}(t)$ in (1). Let $s(t)$ represent the ideal complex baseband signal $x_{\mathrm{L}}(t)$, then

$$\begin{aligned} x_{\mathrm{RF}}^k(t) &= \frac{1}{2^k}\left[s(t)\mathrm{e}^{\mathrm{j}\omega_0 t} + s^*(t)\mathrm{e}^{-j\omega_0 t}\right]^k \\ &= \frac{1}{2^k}\sum_{m=0}^{k}\binom{k}{m} s^m(t)\left(s^*(t)\right)^{k-m}\mathrm{e}^{(j\omega_0 t)(-k+2m)}. \end{aligned} \tag{3}$$

Under narrowband transmission, only the terms very close to frequency f_0 matter. By setting $-k+2m = \pm 1$ and $k = 2p+1$, $y(t)$ can be simplified as

$$d(t) = s(t)\sum_{p=0}^{(K-1)/2}\frac{a_{2p+1}}{2^{2p}}\binom{2p+1}{p+1}|s(t)|^{2p}. \tag{4}$$

If we choose $K = 3$, i.e., with third-order nonlinearity considered, $d(t)$ becomes

$$d(t) = s(t)\left[a_1 + \frac{3a_3}{4}|s(t)|^2\right]. \tag{5}$$

The level of distortion depends on the value of input power backoff (IBO) which defines the average reduced power of the signal in comparison with the power at the compression point. Then, IBO is written as [22].

$$\mathrm{IBO} = 10\log 10\left(P_{\mathrm{sat}}/P_{\mathrm{avg}}\right), \tag{6}$$

where P_{sat} is the average power of the transferred signal at the compression point and P_{avg} is the average power of the transferred signal.

C. CLIPPING

The sampling rate and the bit resolution of the required D/A and A/D converters are usually a limiting factor. Using more quantization bits increases the precision of the signal conversion between the digital and analog domains, but the cost is higher. Furthermore, OFDM has the large peak-to-average power ratio (PAPR) problem. Since the amplitude of the OFDM signal approximately follows a complex Gaussian process, which is the result of an addition of N random variables, the amplitude may vary from positive infinite to negative infinite. However, hardware components usually have finite input range and this causes non-linear distortions on the signal. Therefore, the resolution limit leads to the clipping effect during the conversion between analog and digital domains. The signal suffering clipping can be expressed as

$$d_i = \begin{cases} s_i, & |s_i| < A, \\ A\,\mathrm{sign}(s_i), & |s_i| \geq A, \end{cases} \tag{7}$$

where $i \in \{\mathrm{I}, \mathrm{Q}\}$ denotes the real or the imaginary part of the signal and A denotes the clipping level of the D/A or A/D converter. The clipping level is defined as the amplitude level above which the OFDM signal will be clipped. Clipping level has no unit because this level is set relative to the signal amplitude.

D. IQ IMBALANCE

IQ imbalance happens when two mixers driven by the local oscillator in quadrature, referred to as I and Q branches, have mismatched gains and/or phases. This imbalance can happen at the transmitter, receiver, or both. In the following we take the IQ imbalance at the receiver as an example, assuming that the transmitter is ideal and the receiver is affected by the IQ imbalance. It is assumed that the IQ imbalance happens at the receiver with the phase mismatch $\Delta\phi$ and the amplitude mismatch ϵ. The ideal passband signal $s(t)$, $s(t) = s_{\mathrm{I}}(t) + js_{\mathrm{Q}}(t)$, is converted to the baseband signal $d(t)$ by multiplying it with $s_{\mathrm{LO}}(t)$:

$$\begin{aligned} d(t) &= s(t)s_{\mathrm{LO}}(t) \\ &= \left[s_{\mathrm{I}}(t)\cos(\omega_0 t) - js_{\mathrm{Q}}(t)\sin(\omega_0 t)\right] \\ &\quad \times \left[\cos(\omega_0 t) - j\epsilon\sin(\omega_0 t + \Delta\phi)\right] \\ &= \left\{0.5 + 0.5\epsilon\left[\cos(\Delta\phi) - j\sin(\Delta\phi)\right]\right\} \\ &\quad \times \left[s_{\mathrm{I}}(t) + js_{\mathrm{Q}}(t)\right] \\ &\quad + \left\{0.5 - 0.5\epsilon\left[\cos(\Delta\phi) + j\sin(\Delta\phi)\right]\right\} \\ &\quad \times \left[s_{\mathrm{I}}(t) - js_{\mathrm{Q}}(t)\right] \\ &= K_1 s(t) + K_2 s^*(t), \end{aligned} \tag{8}$$

where

$$\begin{aligned} K_1 &= 0.5 + 0.5\epsilon(\cos(\Delta\phi) - j\sin(\Delta\phi)) \\ &= \left(1 + \epsilon e^{-j\Delta\phi}\right)/2 \end{aligned} \tag{9}$$

and

$$\begin{aligned} K_2 &= 0.5 - 0.5\epsilon(\cos(\Delta\phi) + j\sin(\Delta\phi)) \\ &= \left(1 - \epsilon e^{j\Delta\phi}\right)/2 \end{aligned} \tag{10}$$

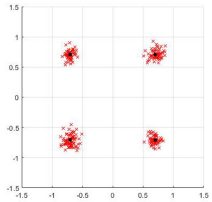
are the coefficients of the ideal signal $s(t)$ and the distortion term $s^*(t)$, respectively.

E. PHASE NOISE

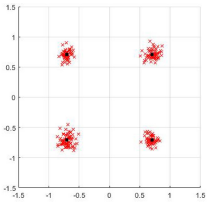
Non-ideal oscillators may cause random phase fluctuations at the output, resulting in the signal distortion. This effect can be formulated as

$$d(t) = s(t)\mathrm{e}^{j\theta(t)}, \tag{11}$$

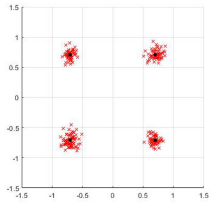
where $\theta(t)$ is the phase distortion. Note that for multiple-antenna systems with multiple RF chains, oscillators in different RF chains may suffer different phase noise [23].



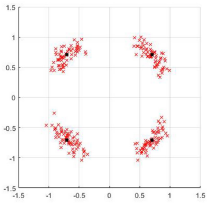
(a) Nonlinear PA IBO = 3 dB



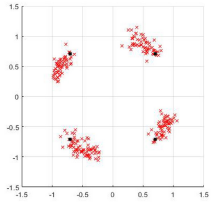
(b) Clipping at the TX = 1.5



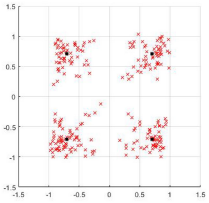
(c) Clipping at the RX = 1.5



(d) PN power = -15 dBc/Hz

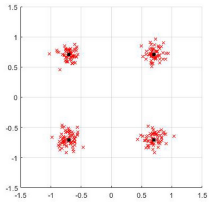


(e) CFO = 0.08

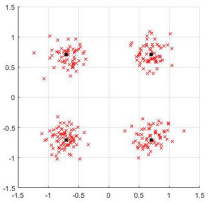


(f) SCO = 500 ppm

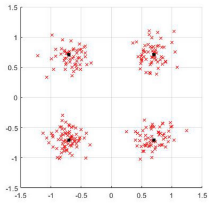
FIGURE 2. Typical constellations under the effects of impairments for single-carrier QPSK systems (AWGN channel, 20 dB SNR).



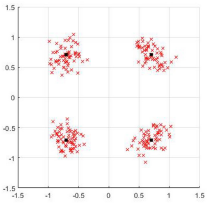
(a) Nonlinear PA IBO = 3 dB



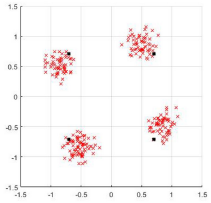
(b) Clipping at the TX = 1.5



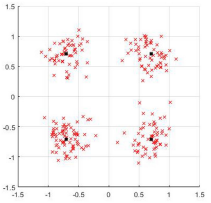
(c) Clipping at the RX = 1.5



(d) PN power = -15 dBc/Hz



(e) CFO = 0.08



(f) SCO = 500 ppm

FIGURE 3. Typical constellations under the effects of impairments for OFDM systems with QPSK modulation (AWGN channel, 20 dB SNR).

F. CARRIER FREQUENCY OFFSET

The CFO happens when there is an oscillator mismatch between the transmitter oscillator and the receiver oscillator. The CFO may destroy the orthogonality between subcarriers and cause inter-carrier interference (ICI). With the CFO, the received signal $d(t)$ becomes

$$d(t) = s(t)\mathrm{e}^{j\Delta ft}, \tag{12}$$

where Δf is the frequency difference between the transmitter oscillator and the receiver oscillator.

G. SAMPLING CLOCK OFFSET

Sampling clock offset happens when the sampling period T_s has an offset δ compared to the ideal sampling frequency. With the sampling clock offset, even if the initial sampling point is optimal, the subsequent sampling points slowly drift with time. If the receiver clock is faster (or slower) than the transmitter clock, fractional samples should be added (or removed) from the received symbol. With the sampling clock offset, the sampled received signal becomes

$$\begin{aligned} d[n] &= s(n \cdot (1+\delta)T_s) \\ &= s(nT_s + n\delta T_s), \quad 0 \le n \le N-1. \end{aligned} \tag{13}$$

H. IMPAIRMENT CHARACTERISTICS

FIGURE 2 shows typical QPSK constellations of a single-carrier system suffering from nonlinear power amplification, clipping, phase noise, CFO, and SCO, respectively. Phase noise adds uncertainty to the phase of the signal and makes the points on the constellation spread out, leading to a phenomenon known as "constellation spreading" and a random rotation of the constellation. The CFO also causes a rotation of the constellation points, and the degree of rotation is affected by the offset. The SCO causes symbol timing misalignment between the transmitter and the receiver, causing received symbols misplaced on the constellation diagram. The effects of power amplifier nonlinearity and clippling, however, are less significant. For OFDM systems, the characteristics of different impairments are less distinct, as shown in FIGURE 3. The nonlinearity in power amplifiers and the clipping have more impacts on OFDM systems, distorting the amplitude and phase of signals. The effects of phase noise, CFO, and SCO are more like added noise but CFO also causes a rotation of the constellation points.

III. REVIEW OF EXISTING METHODS COMPARED IN THIS PAPER

In this section, existing methods which will be compared in this paper are reviewed. Both traditional (non-DL) methods and deep learning-based methods are discussed. Here we define $\boldsymbol{x}$ and X as transmitted signal in the time domain and in the frequency domain, respectively. The corresponding

received signal are $\boldsymbol{y}$ and Y, and the symbol "$'$" denotes signal that is distorted.

A. TRADITIONAL (NON-DL) METHODS

- *Clipping Effect Compensation:* The method proposed by Kim et al. [14] models the clipped signal as the combination of the original signal and the clipping distortion. Assuming that the channel is known, the estimated signal $\hat{X}$ can be obtained by the maximum likelihood method after equalization. By using the selection criterion and the orthogonal matching pursuit (OMP) algorithm, the recovered clipping noise can be obtained to find the clipping-corrected signal. Rietman et al. [15] propose a simple clipping correction algorithm which can be implemented easily in the digital receiver using empty subcarriers. The recovered complex signal $\tilde{Y}$ in the frequency domain can be expressed as

$$\tilde{Y} = Y' - \frac{N}{M} F P_{\text{clip},\underline{y}} F^{-1} P_{\text{empty}} Y', \tag{14}$$

where N is the number of subcarriers, Y' is the clipped received signal, F is the Fourier transform, and F^{-1} is the inverse Fourier transform. P_{empty} projects the signal onto M empty subcarriers, i.e., $(P_{\text{empty}}\, Y)_m = Y_m$ if $m \in \mathcal{M}$ and zero otherwise. $P_{\text{clip},\underline{y}}$ projects the real and imaginary parts of the signal onto the subcarriers according to the clipping ratio A. The real part and the imaginary part of the signal are zero if $\underline{y}$ is not equal to the clipping ratio. The formula can be expressed as

$$\begin{aligned} \text{Re}\left(P_{\text{clip},\underline{y}} x\right)_n &= \begin{cases} \text{Re}(x_n), & \text{if } |\text{Re}(y_n)| = A, \\ 0, & \text{otherwise}, \end{cases} \\ \text{Im}\left(P_{\text{clip},\underline{y}} x\right)_n &= \begin{cases} \text{Im}(x_n), & \text{if } |\text{Im}(y_n)| = A, \\ 0, & \text{otherwise}. \end{cases} \end{aligned} \tag{15}$$

- *IQ Imbalance Compensation:* Due to the IQ imbalance, the distorted signal at the transmitter and the receiver can be modeled as

$$\boldsymbol{x}' = K_{t1} x + K_{t2} x^*, \tag{16}$$

$$\boldsymbol{y}' = K_{r1} y + K_{r2} y^*, \tag{17}$$

where $\boldsymbol{x}'$ and $\boldsymbol{y}'$ are respectively the signal distorted by IQ imbalance at the transmitter and receiver, and K_{t1}/K_{r1} and K_{t2}/K_{r2} are respectively the same as (9) and (10). If the IQ imbalance parameters are accurately acquired, the compensation at the transmitter and the receiver can be carried out respectively as

$$\tilde{x} = \frac{x' - \eta_{\text{t}} x'^*}{K_{t1}\left(1 - |\eta_{\text{t}}|^2\right)}, \quad \tilde{y} = \frac{y' - \eta_{\text{r}} y'^*}{K_{r1}\left(1 - |\eta_{\text{r}}|^2\right)}, \tag{18}$$

where $\eta_{\text{t}} = \frac{K_{t2}}{K_{t1}}$ and $\eta_{\text{r}} = \frac{K_{r2}}{K_{r1}^*}$. The algorithm proposed by Zhang et al. [10] roughly estimates the imbalance parameters by using least square. Then, the problem is divided into two subproblems, one for the estimation of η_{t} and the other for the estimation of η_{r}. The algorithm first fixes one parameter as the initial estimated value and then calculates another parameter until convergence. After the first parameter converges, it takes turn to update another parameter until convergence.

- *Phase Noise Compensation:* The method proposed by Mir and Buttar [12] suppresses both the common phase error and the ICI by using the least square method to minimize the cost function. The received signal can be expressed as $\boldsymbol{y}' = \boldsymbol{h} \otimes \boldsymbol{x} \cdot e^{j\theta} + \boldsymbol{n}$, where $\boldsymbol{x}$ denotes the transmitted signal, $\boldsymbol{h}$ denotes the channel response, $\otimes$ denotes the linear convolution operation, and θ denotes the phase noise. After removing the cyclic prefix and taking the discrete Fourier transform, the frequency domain signal can be expressed as

$$Y'(k) = X(k)H(k)I(0) + W, \tag{19}$$

where W represents the summation of the noise and ICI terms. By using the zero-forcing equalization, the transmitted signal can be estimated by $\hat{X}(k) = Y'(k)C(k)$, where C is obtained by

$$C(k) = \frac{I^*(0)H^*(k)}{|I(0)H(k)|^2}. \tag{20}$$

The least square method is applied to minimize the cost function to find $I(0)$ and the equation can be expressed as $\min_{I(0)} \sum |Y'(k) - I(0)X(k)H(k)|^2$, which leads to the estimation

$$\hat{I}(0) = \frac{Y'(k)X^*(k)H^*(k)}{|X(k)H(k)|^2}. \tag{21}$$

Therefore, the estimation of $\hat{I}(0)$ can be carried out by using (21). Since the result may not be accurate, the decision feedback is used for the further enhancement of the performance. In MIMO systems, the combined effect of phase noise and IQ imbalance in the frequency domain can be modeled as $Y'(k) = K_1 X(k)H(k)I(0) + K_2 X^*(k)H^*(k)I^*(0) + W$ [13], where W represents the summation of the noise and ICI terms. The algorithm is designed to estimate the IQ imbalance effect first and then the phase noise effect.

The paper by Bogana [13] estimates the IQ imbalance in MIMO systems by using two consecutive training symbols. After the estimation of the IQ imbalance, the phase noise is estimated and updated by iterative techniques.

- *Carrier Frequency Offset Compensation:* To estimate the carrier frequency offset ξ, the phase difference between the cyclic prefix and the corresponding rear part of an OFDM signal is $2\pi N\xi/N = 2\pi\xi$ if there is no channel effect. Therefore, in the paper by Nishad and Singh [7], the carrier frequency offset is estimated from the phase angle of the product of the rear part of the OFDM signal and the cyclic prefix. The equation is formulated as $\hat{\xi} = (1/2\pi)\arg\{y'^*[n]y'[n+N]\}$, where $n = -1, -2, \ldots, -N_g$ and N_g is the length of cyclic

prefix. The average is taken over the samples in a cyclic prefix interval and can be expressed as

$$\hat{\xi} = (1/2\pi)\arg\left\{\sum_{n=-N_g}^{-1} y'^{*}[n]y'[n+N]\right\}, \quad (22)$$

where $\arg(\cdot)$ denotes $\tan^{-1}(\cdot)$. Zhang et al. [8] extend this algorithm to MIMO systems. By denoting $X_{\mu,g}(k)$ as the transmitted data at the kth subcarrier in the gth block from the μth transmit antenna, the received time domain signal at the mth receive antenna is expressed as $\mathbf{y}'_{m,g} = \mathbf{E}\{\hat{\xi}\}\mathbf{F}^H \sum_{\mu=1}^{M_t} \text{diag}(\mathbf{X}_{\mu,g})\mathbf{H}_{\mu,m} + \mathbf{w}_{m,g}$, where $\mathbf{w}$ is the white Gaussian noise, $\mathbf{F}$ is the discrete Fourier transform, $\mathbf{E}\{\hat{\xi}\}$ is the phase rotation caused by the CFO, and M_t is the total number of transmit antennas. With the first P subcarriers used for data transmission and the others being null subcarriers, the recovered received signal in the frequency domain can be expressed as $\tilde{Y}_g\{\hat{\xi}\} = \mathbf{F}\mathbf{E}^H\{\hat{\xi}\}\mathbf{Y}'_g$ after performing the CFO compensation with a trial CFO value $\hat{\xi}$. They further define $\tilde{Y}_g\{\hat{\xi}\} = [\boldsymbol{\xi}_g^{(0)}\{\hat{\xi}\}, \boldsymbol{\xi}_g^{(1)}\{\hat{\xi}\}, \ldots, \xi_g^{(N-1)}\{\hat{\xi}\}]^T$, where $\xi_g^{(k)}\{\hat{\xi}\}$ corresponds to the $(k+1)$th row vector of $\tilde{Y}_g\{\hat{\xi}\}$ and $\eta^{(k)}\{\hat{\xi}\} = [\xi_1^{(k)}\{\hat{\xi}\}, \xi_2^{(k)}\{\hat{\xi}\}, \ldots, \xi_{M_r}^{(k)}\{\hat{\xi}\}]$. By taking the first order derivative of the cost function, $C\{\hat{\xi}\} = \sum_{k=0}^{N-1} \det(\eta^{(k)}\{\hat{\xi}\}\eta^{(k)}\{\hat{\xi}\}^H)$, the minimum can be derived.

- *Sampling Clock Offset Compensation:* To compensate for the sampling clock offset, an additional fractional number of samples can be added or removed so that the ICI caused by the SCO can be largely mitigated. The method proposed by Briggs et al. [16] takes advantage of the modulation of a fractional delay filter which can produce the delay for resampling the signal. However, an ideal fractional delay filter does not exist in practical systems, resulting in the limitation on the precision. For practical reasons, the method proposed by Briggs et al. [16] is only used to produce the possible delay values which are in the interval between zero and one.

B. DEEP LEARNING-BASED METHODS

- *Clipping Effect Compensation:* Sang and Xu [17] investigate the clipping effect with the help of neural networks. Their proposed method based on deep learning reconstructs the signal in the frequency domain by minimizing the mean squared error. Their method combines both the advantages of neural networks and the decision-aided reconstruction (DAR) algorithm. A four-layer fully connected neural network is used and the hyperbolic tangent is chosen as the activation function. The number of units is three times the size of the input data. The mean squared error and the Adam optimizer are chosen as the loss function and the optimizer, respectively. The neural network resides after equalization to reconstruct the signal. Then, the signal is transformed to the time domain and the reconstructed signal is compared with the clipping level. If the signal amplitude is greater than the clipping level, the DAR algorithm is used to detect the signal.
- *Phase Noise Compensation:* Park et al. [18] take advantage of neural networks to compensate the phase noise effect. Their proposed neural network is composed of nine fully-connected layers with the number of units four times the input size, and the softmax activation function as the output layer for data detection. The leaky ReLU is used as the nonlinear activation function, and the stochastic gradient descent with momentum (SGDM) optimizer is adopted to train the neural network.
- *Carrier Frequency Offset Compensation:* After the signal is received at the receiver, the signal is directly fed into the CFO equalization network to reduce the effect of CFO, as proposed by Kumari et al. [19].

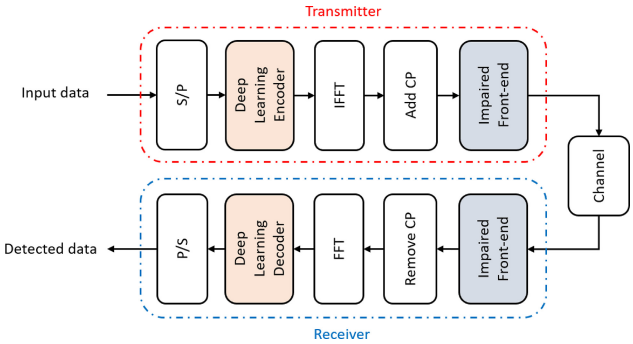


FIGURE 4. Proposed deep learning-based OFDM architecture for single-antenna systems.

IV. PROPOSED END-TO-END DESIGNS

We propose deep learning-based transmitter and receiver for single-antenna and multi-antenna OFDM systems. Two deep neural network (DNN) implementations of the codec in the transceiver, DLNN and ResNet-DCDNN, are proposed to combat hardware impairments issues for both single-antenna systems and multi-antenna systems. We will see later that DLNN performs better for single-antenna systems and 2×2 MIMO systems while ResNet-DCDNN performs better for 2×4 MIMO systems.

A. PROPOSED ARCHITECTURE FOR SINGLE-ANTENNA SYSTEMS

We first consider the single-antenna systems. The idea is to insert a DNN-based encoder between S/P and IFFT at the transmitter and a DNN-based decoder between FFT and P/S at the receiver to combat the hardware impairments, as illustrated in FIGURE 4. The DNN-based encoder and decoder can be implemented by either DLNN or ResNet-DCDNN. Note that the encoder and the decoder not only cover functions originally served by symbol mapper and

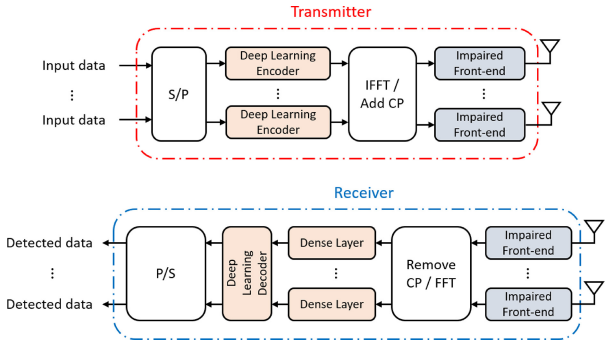


FIGURE 5. Proposed deep learning-based OFDM architecture for multi-antenna systems.

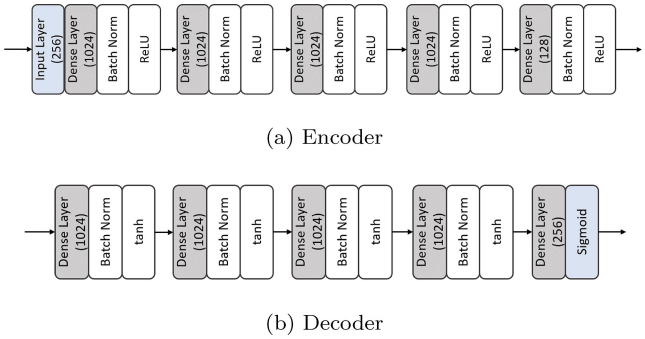


FIGURE 6. Proposed DLNN codec.

symbol demapper, respectively, but also mitigate the hardware impairments. Suppose that the input sequence data $\boldsymbol{s}$ consists of symbols s_k, where k denotes the index of the symbol. To comply with the structure of the deep learning-based encoder, s_k is a one-hot vector which consists of zeros and only one '1'. After S/P, the input sequence is fed into the encoder composed of the deep neural network. Denoting the kth output of the neural network as X_k, the encoder is functioned as $f_{\theta_t}(\boldsymbol{s}) = X_k$, where θ_t is the neural network parameter set. Then, the output of the neural network, X_k, is fed to the IFFT and then the time domain signal $x(t)$ is obtained. The decoder at the receiver takes the encoded symbol X_k as the input and then outputs $\hat{s}_k$, i.e., the decoder functions as $g_{\theta_r}(X_k) = \hat{s}_k$, where θ_r is the neural network parameter set. Note that the only difference between the proposed deep learning-based OFDM architecture and the original OFDM structure is the extra neural network encoder at the transmitter and the extra neural network decoder at the receiver.

Note that this paper does not consider channel coding since we would like to focus on how the impairments affect the OFDM systems and how the proposed novel designs can effectively mitigate the impairments.

B. PROPOSED ARCHITECTURE FOR MULTI-ANTENNA SYSTEMS

For multi-antenna systems, we consider that each transmit antenna has its own RF chain [23], [24] and suffers from the same or different extent of hardware impairments, as shown in FIGURE 5. Similar to single-antenna systems, a DNN-based encoder is inserted between S/P and IFFT at the transmitter and a DNN-based decoder is inserted between FFT and P/S. Again, the DNN-based encoder and decoder can be implemented by either DLNN or ResNet-DCDNN.

C. PROPOSED DLNN CODEC

The DLNN encoder is composed of five dense layers with batch normalization and the ReLU activation function, while the DLNN decoder is composed of four dense layers with batch normalization and the tanh activation function, and one dense layer with the sigmoid output, as shown in FIGURE 6

TABLE 1. Parameters of the proposed DLNN.

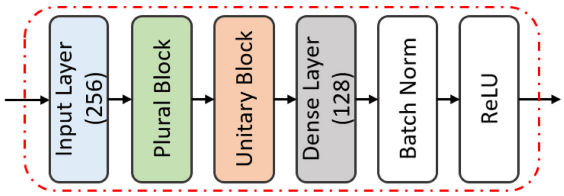
DLNN encoder	
Number of dense layers	5
Number of neurons in a dense layer	1024
Activation function	ReLU
Weight initialization	Xavier
DLNN decoder	
Number of dense layers	5
Number of neurons in a dense layer	1024
Activation function	tanh
Weight initialization	Xavier
Loss function	Mean squared error

and summarized in TABLE 1. The encoder and the decoder are trained as an autoencoder by minimizing the mean squared error loss function.

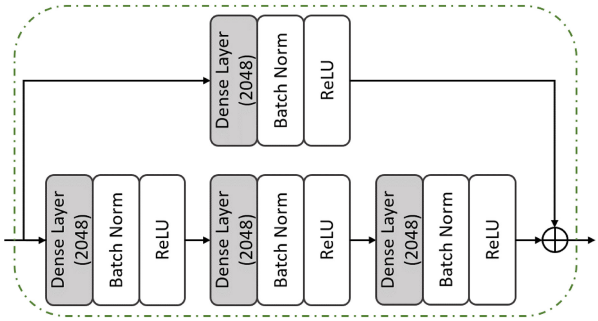
D. PROPOSED RESNET-DCDNN CODEC

The ResNet-DCDNN encoder is composed of a plural block, a unitary block, and a dense layer, followed by batch normalization and the ReLU activation function, as shown in FIGURE 7. The plural block consists of two paths, with the lower path having three dense layers and the upper path having one dense layer. All dense layers are followed by batch normalization and the ReLU activation function. This design is inspired by residual networks to avoid the vanishing gradient problem. However, according to our experiments, this design alone does not deliver satisfactory performance at high signal-to-noise ratio (SNR). As a result, a unitary block with two dense layers and the skip connection architecture is cascaded to the plural block. The dense layers are followed by batch normalization and the ReLU activation function. The parameters of the ResNet-DCDNN encoder are summarized in TABLE 2.

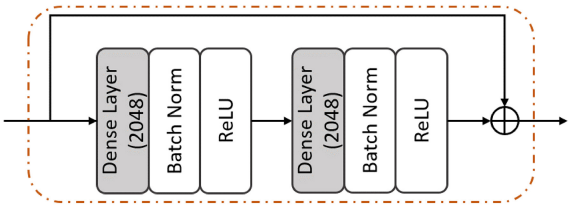
The ResNet-DCDNN decoder is composed of two convolution blocks along with the residual architecture, followed by a convolutional layer, batch normalization, the tanh activation function, a dense layer, and the sigmoid output. A convolution block has two convolutional layers with each convolutional layer followed by batch normalization and the tanh activation function. The kernel stride is one and the zero



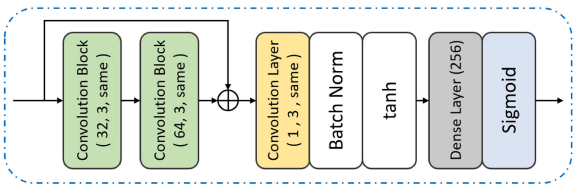
(a) ResNet-DCDNN encoder



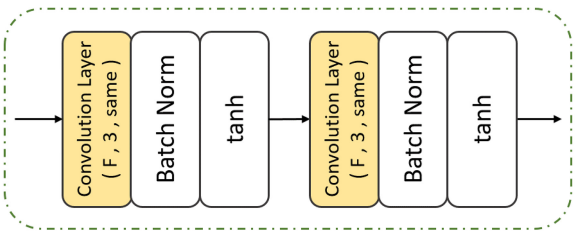
(b) Plural block



(c) Unitary block



(d) ResNet-DCDNN decoder



(e) Convolution block

FIGURE 7. Proposed ResNet-DCDNN codec. 'F' in the convolution block means the filter size specified in the ResNet-DCDNN decoder.

padding is applied. The parameters of the ResNet-DCDNN decoder are summarized in TABLE 3. Note that the filter size is determined by experiments.

TABLE 2. Parameters of the proposed ResNet-DCDNN encoder.

Plural block	
Number of dense layers	1 and 3
Number of neurons in a dense layer	2048
Activation function	ReLU
Weight initialization	Xavier
Unitary block	
Number of dense layers	2
Number of neurons in a dense layer	2048
Activation function	ReLU
Weight initialization	Xavier

TABLE 3. Parameters of the proposed ResNet-DCDNN decoder.

Convolution block 1	
Filter size	32
Kernel size	3
Zero padding	same
Convolution block 2	
Filter size	64
Kernel size	3
Zero padding	same
Convolutional layer	
Filter size	1
Kernel size	3
Zero padding	same
Number of neurons in a dense layer	2048
Activation function	tanh
Weight initialization	Xavier
Loss function	Mean squared error

For both DLNN and ResNet-DCDNN, the mean squared error is adopted as the loss function, and the Xavier initialization is adopted during training.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed deep learning-based OFDM architectures and the encoder/decoder designs. Both the AWGN channel and the Rayleigh fading channel are considered in the single- and multi-antenna systems. The training parameters for the single-antenna systems and the multi-antenna systems are summarized in TABLE 4. The training data is generated as follows. Since the proposed end-to-end architecture can be regarded as an autoencoder, the input bits and the output bits are the same. With 64 subcarriers each assigned 2 bits, in each training epoch 64 2-bit information bits are generated and then converted to one-hot vectors, or 64 4-bit one hot vectors are generated directly. The training and testing of the neural networks are run on a computer equipped with Intel Core i7-8700 CPU@3.20GHz, 16GB DRAM, and NVIDIA GeForce GTX 1080Ti graphics card.

TABLE 4. Training parameters.

Parameter	
Number of subcarriers	64
Size of cyclic prefix	5
Number of channel taps	4
Mini-batch size	100
Adam optimizer	$\alpha = 0.0001$, $\beta_1 = 0.99$, $\beta_2 = 0.999$
Training SNR	14, 18, and 22 dB (single antenna) 6 and 8 dB (multiple antennas)
Training epochs	30000
Learning rate	0.0001

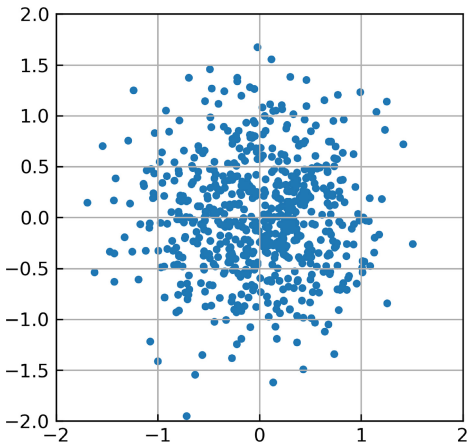


FIGURE 8. Example constellation points (resulted from 10 bit-sequence realizations) after the DLNN encoder (i.e., before the IFFT) of the proposed deep learning-based OFDM system.

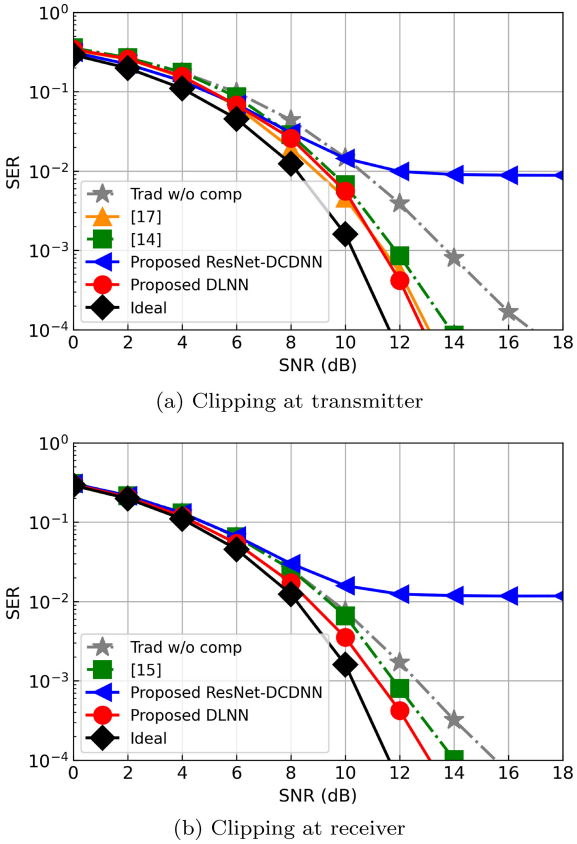


FIGURE 9. Performance comparison of various methods under (a) the clipping effect at the transmitter with clipping level 1.5 and (b) the clipping effect at the receiver with clipping level 1.5.

The following schemes are compared:

- *Trad w/o comp* represents the traditional method without any compensation.
- *Proposed ResNet−DCDNN* represents the method using the proposed ResNet-DCDNN neural network.
- *Proposed DLNN* represents the method using the proposed DLNN neural network.
- *Ideal* represents the situation when there are no hardware impairment effects.
- *[X]* means the method proposed in the reference paper [X].
- *Non−DL Algorithms* represents the combination of all non-DL algorithms mentioned in Section III-A.

Note that although our compared schemes use different equalizers such as zero forcing (ZF) and minimum mean squared error (MMSE), their designs of algorithms are not based on the equalizer. Therefore, we adopt the ZF equalizer in all simulations. Also note that the impairment settings that are widely considered in the literature are chosen.

A. CONSTELLATION

Before showing the performance of the proposed system, example constellation points of the proposed system are shown in FIGURE 8. This plot is generated by 10 bit-sequence realizations and grouping every two outputs of the DLNN encoder to form a complex vector. It can be observed that the dense layers in the DLNN encoder result in a very high order modulation, which is common for deep learning-based systems with an end-to-end design.

B. PERFORMANCE OF SINGLE-ANTENNA SYSTEMS

We first show the performance of the proposed designs under individual hardware impairments. Note that the same neural network is used for dealing with both the individual impairments and all impairments considered simultaneously.

FIGURE 9 compares the symbol error rate of various schemes under the clipping with saturation level 1.5 over the AWGN channel. As shown in FIGURE 9(a) when only the transmitter has the clipping effect, the proposed DLNN outperforms all other methods. Note that the performance of the proposed ResNet-DCDNN is worse than that of the proposed DLNN for single-antenna systems. As shown in FIGURE 9(b) when only the receiver has the clipping effect, the proposed DLNN again outperforms all other methods. This shows that unlike the other methods, the proposed DLNN can deal with the clipping effect no matter the clipping happens at the transmitter or at the receiver.

In FIGURE 10, the symbol error rate of various methods under the phase noise with the noise power –10 dBc/Hz (with the signal power set to unit) at both transmitter and receiver over the AWGN channel is compared. It is shown

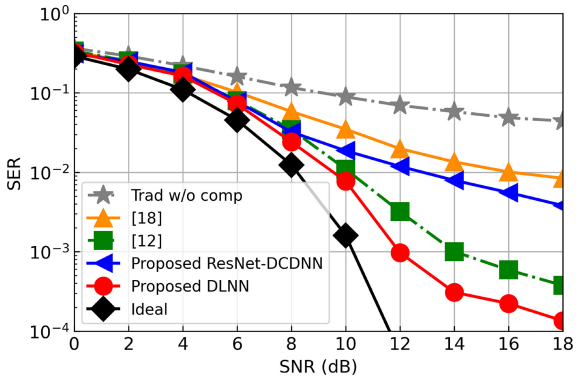


FIGURE 10. Performance comparison of various methods under the phase noise power -10 dBc/Hz at both the transmitter and the receiver.

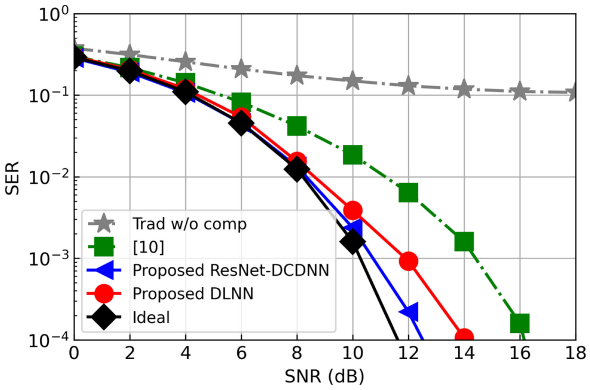


FIGURE 11. Performance comparison of various methods under the IQ imbalance with amplitude mismatch 0.3 and phase mismatch 23° at both the transmitter and the receiver.

that the proposed DLNN outperforms all other methods. Note that although the proposed ResNet-DCDNN outperforms some other methods, there is still a performance gap to the proposed DLNN.

FIGURE 11 compares the symbol error rate of various methods under the IQ imbalance with the amplitude mismatch 0.3 and the phase mismatch 23° at both transmitter and receiver over the AWGN channel. While both ResNet-DCDNN and DLNN outperform the other methods, the performance of ResNet-DCDNN is now superior and approaches that of the ideal case.

The symbol error rate of various methods under the carrier frequency offset when δT_s equals to 0.08 over the AWGN channel is compared in FIGURE 12. Although the proposed designs do not estimate and compensate the carrier frequency offset as well as the other methods, they still have a significant performance gain over the method without compensation.

FIGURE 13 compares the symbol error rate of various methods under the sampling clock offset over the AWGN channel. Although the proposed designs are not superior to the method proposed by Briggs et al. [16] when the offset is 400 ppm, the proposed DLNN outperforms their method when the offset is 500 ppm. This means that when

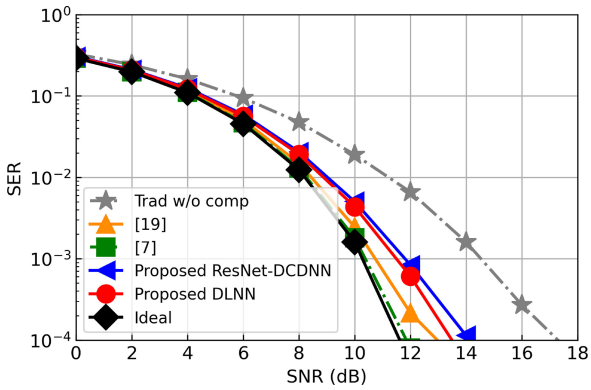
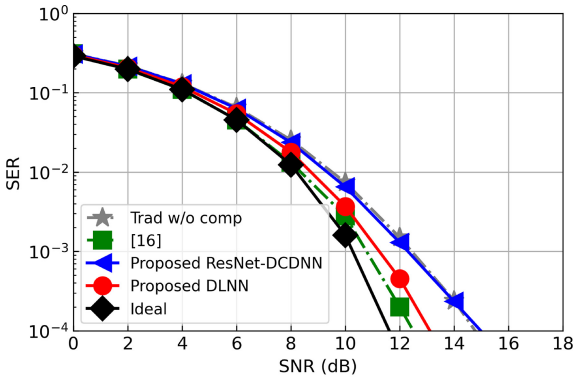
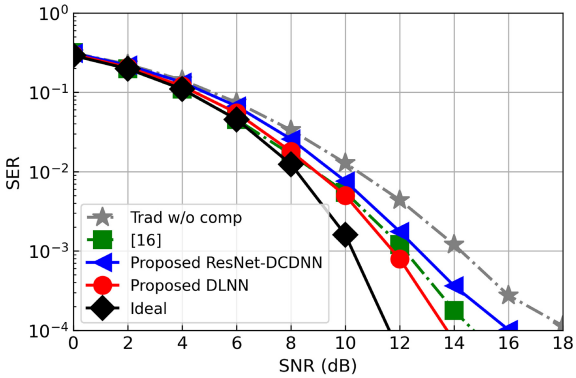


FIGURE 12. Performance comparison of various methods under the receiver carrier frequency offset when δT_s equals to 0.08.



(a) Sampling clock offset 400 ppm



(b) Sampling clock offset 500 ppm

FIGURE 13. Performance comparison of various methods when (a) the sampling clock offset is 400 ppm and (b) the sampling clock offset is 500 ppm.

the sampling clock offset is larger, the proposed approach has advantages.

Finally, FIGURE 14 shows the performance of DLNN under different impairment levels of various individual hardware impairments. It can be observed that the proposed DLNN is robust to different impairment levels. Note that DLNN does not necessarily perform worse when the impairment is more severe.

Now we consider all the hardware impairments simultaneously. Two hardware impairment settings are considered

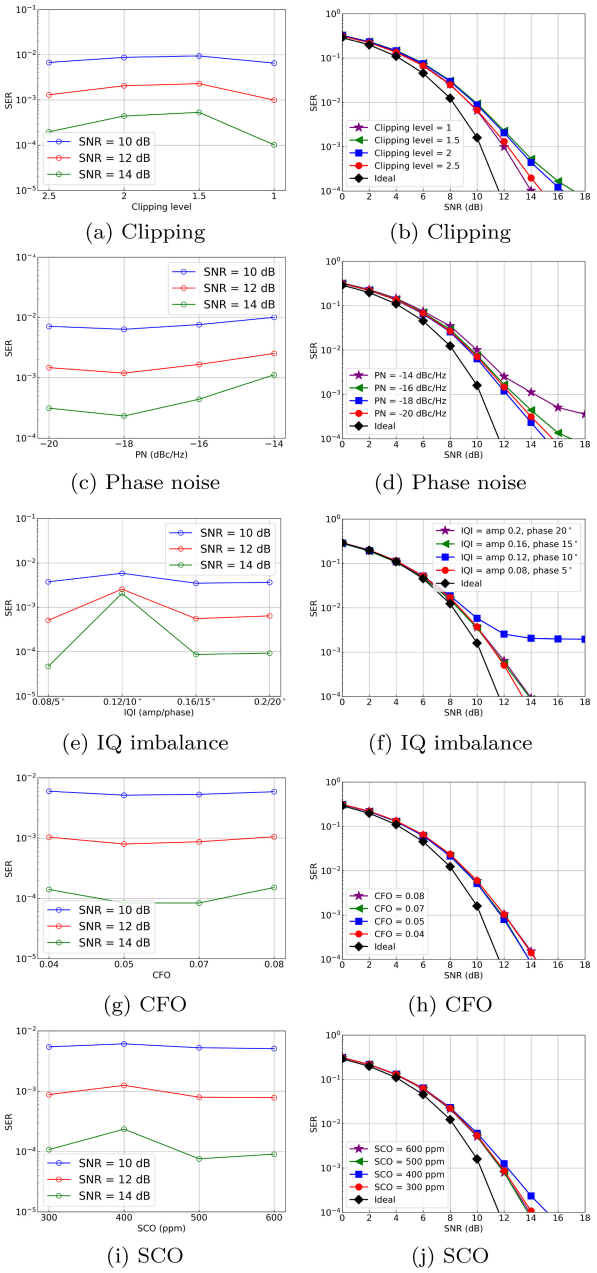


FIGURE 14. Performance of DLNN under different impairment levels of various individual hardware impairments.

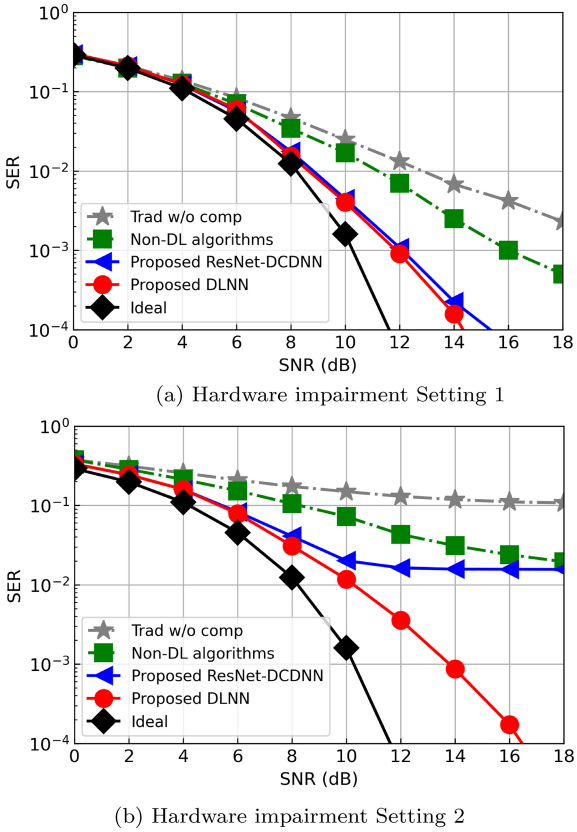


FIGURE 15. Performance comparison of various methods for the single-antenna system over the AWGN channel when all hardware impairments are considered.

TABLE 5. Hardware impairment settings for the single-antenna system (AWGN channel).

Transmitter	Setting 1	Setting 2
Clipping level	2.5	1.5
Phase noise power	-25 dBc/Hz	-18 dBc/Hz
IQ imbalance amplitude/phase	0.2/11°	0.3/23°
Input power backoff	5 dB	4 dB
Receiver	**Setting 1**	**Setting 2**
Clipping level	2.5	1.5
Phase noise power	-25 dBc/Hz	-18 dBc/Hz
IQ imbalance amplitude/phase	0.2/11°	0.3/23°
Carrier frequency offset	0.03	0.08
Sampling clock offset	300 ppm	500 ppm

under the AWGN channel, with Setting 2 having more severe hardware impairments than Setting 1, as shown in TABLE 5. FIGURE 15(a) shows the results under Setting 1. It can be seen that both the proposed designs significantly outperform the other methods. In particular, although the performance of the other methods may be comparable to the proposed designs when individual hardware impairments are considered, their performances are much worse than the proposed designs when all hardware impairments are taken into account. Under Setting 2, as shown in FIGURE 15(b), the performance gap between the proposed DLNN and the other methods is even larger.

When the Rayleigh fading channel is considered, the hardware impairment settings are summarized in TABLE 6. It can be seen in FIGURE 16 that the performance of the proposed DLNN significantly outperforms all other methods, although showing an error floor around SER 10^{-3}.

C. PERFORMANCE OF MULTI-ANTENNA SYSTEMS

Now we evaluate the performance of the proposed designs for MIMO systems. The 2×2 MIMO system is firstly evaluated, with two scenarios considered: the same hardware impairments (with the settings summarized in TABLE 7)

TABLE 6. Hardware impairment settings for the single-antenna system (Rayleigh fading channel).

Transmitter	
Clipping level	2.5
Phase noise power	-25 dBc/Hz
IQ imbalance amplitude/phase	0.08/4.5°
Input power backoff	4 dB
Receiver	
Clipping level	2.5
Phase noise power	-25 dBc/Hz
IQ imbalance amplitude/phase	0.08/4.5°
Carrier frequency offset	0.05
Sampling clock offset	400 ppm

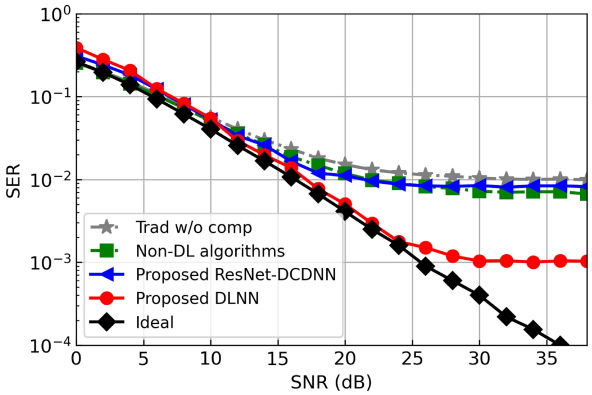


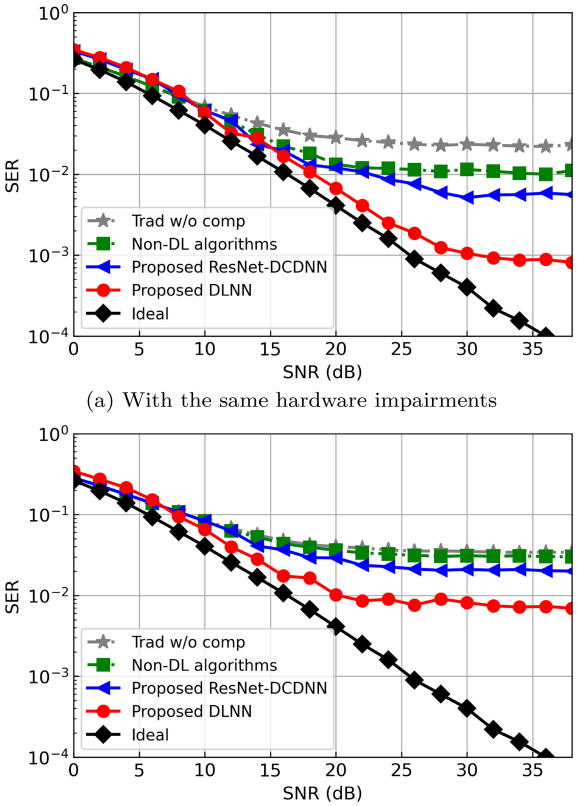
FIGURE 16. Performance comparison of various methods for the single-antenna system over the Rayleigh fading channel when all hardware impairments are considered.

TABLE 7. Hardware impairment settings for the 2×2 MIMO system with the same hardware impairments (Rayleigh fading channel).

Transmitter	
Clipping level	2.5
Phase noise power	-25 dBc/Hz
IQ imbalance amplitude/phase	0.02/1°
Input power backoff	4 dB
Receiver	
Clipping level	2.5
Phase noise power	-25 dBc/Hz
IQ imbalance amplitude/phase	0.02/1°
Carrier frequency offset	0.03
Sampling clock offset	500 ppm

TABLE 8. Hardware impairment settings for the 2×2 MIMO system with different hardware impairments (Rayleigh fading channel).

Transmitter	TX1	TX2
Clipping level	2.5	2.5
Phase noise power	-20 dBc/Hz	-25 dBc/Hz
IQ imbalance amplitude/phase	0.01/1°	0.02/2°
Input power backoff	4 dB	5 dB
Receiver	**RX1**	**RX2**
Clipping level	2.5	2.5
Phase noise power	-25 dBc/Hz	-30 dBc/Hz
IQ imbalance amplitude/phase	0.03/1°	0.01/2.5°
Carrier frequency offset	0.02	0.01
Sampling clock offset	600 ppm	600 ppm



(a) With the same hardware impairments

(b) With different hardware impairments

FIGURE 17. Performance comparison of various methods for the 2×2 MIMO system over the Rayleigh fading channel when all hardware impairments are considered.

and different hardware impairments (with the settings summarized in TABLE 8). Both the proposed DLNN and ResNet-DCDNN outperform the other methods, and DLNN has a clear advantage, as shown in FIGURE 17. Although the performance of all methods degrades when there are different hardware impairments at different antenna chains, the proposed DLNN performs significantly better than the other methods.

FIGURE 18 shows the performance evaluation for the 2 × 4 MIMO system. Two scenarios are also considered: the same hardware impairments (with the settings summarized in TABLE 9) and different hardware impairments (with the settings summarized in TABLE 10). Similar to the case of the 2 × 2 MIMO system, both the proposed DLNN and ResNet-DCDNN outperform the other methods. However, now ResNet-DCDNN has the advantage, and it approaches the ideal case for the scenario of same hardware impairments. For the scenario of different hardware impairments at different antenna chains, the proposed DLNN and the proposed ResNet-DCDNN significantly outperform the other methods.

TABLE 9. Hardware impairment settings for the 2×4 MIMO system with the same hardware impairments (Rayleigh fading channel).

Transmitter	
Clipping level	2.5
Phase noise power	-30 dBc/Hz
IQ imbalance amplitude/phase	0.1/6°
Input power backoff	3 dB
Receiver	
Clipping level	2.5
Phase noise power	-30 dBc/Hz
IQ imbalance amplitude/phase	0.1/6°
Carrier frequency offset	0.03
Sampling clock offset	500 ppm

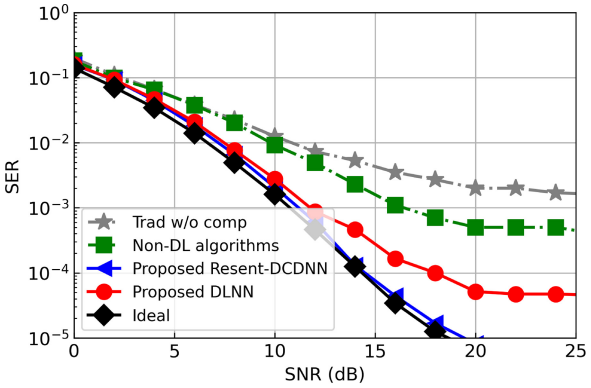
TABLE 10. Hardware impairment settings for the 2×4 MIMO system with different hardware impairments (Rayleigh fading channel).

Transmitter	TX1	TX2
Clipping level	2.5	2.5
Phase noise power	-30 dBc/Hz	-25 dBc/Hz
IQ imbalance amplitude/phase	0.2/6°	0.08/4.5°
Input power backoff	4 dB	5 dB
Receiver	**RX1**	**RX2**
Clipping level	2.5	2.5
Phase noise power	-25 dBc/Hz	-20 dBc/Hz
IQ imbalance amplitude/phase	0.1/1°	0.05/11.5°
Carrier frequency offset	0.04	0.02
Sampling clock offset	500 ppm	400 ppm
Receiver	**RX3**	**RX4**
Clipping level	2.5	2.5
Phase noise power	-30 dBc/Hz	-25 dBc/Hz
IQ imbalance amplitude/phase	0.02/1°	0.1/11.5°
Carrier frequency offset	0.01	0.02
Sampling clock offset	500 ppm	600 ppm

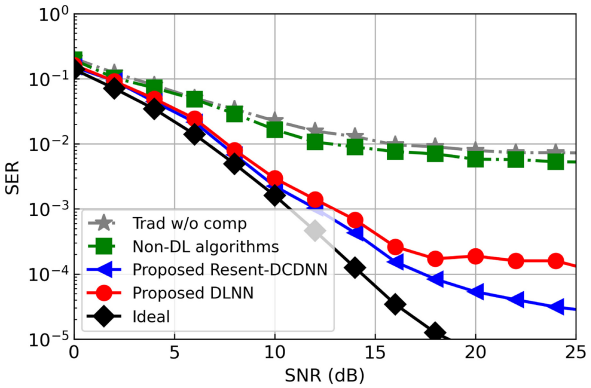
Overall, the two proposed deep learning-based designs effectively combat hardware impairments for not only SISO systems but also MIMO systems. In general, DLNN is outstanding for SISO and 2×2 MIMO systems while ResNet-DCDNN finds advantages for 2×4 MIMO systems. This can be clearly observed in FIGURE 19 comparing performances of DLNN and ResNet-DCDNN for 2×2 and 2×4 MIMO systems under the same settings shown in TABLE 11. It can also be seen that 2×4 MIMO systems are more sensitive to hardware impairments and have more significant SER degradation compared to 2×2 MIMO systems when no compensation is applied. However, with the proposed designs, the performance of 2×4 MIMO systems is significantly improved.

D. TRANSFER LEARNING FOR TIME-VARYING IMPAIRMENT LEVELS

To address the issue of time-varying impairments, transfer learning may be applied to fine-tune the trained weights



(a) With the same hardware impairments.



(b) With different hardware impairments.

FIGURE 18. Performance comparison of various methods for the 2×4 MIMO system over the Rayleigh fading channel when all hardware impairments are considered.

TABLE 11. Hardware impairment settings for comparing 2×2 and 2×4 MIMO systems (different hardware impairments at different antennas, Rayleigh fading channel).

Transmitter		TX1	TX2
Clipping level		2.5	2.5
Phase noise power		-30 dBc/Hz	-25 dBc/Hz
IQ imbalance amplitude/phase		0.2/6°	0.08/4.5°
Input power backoff		4 dB	5 dB
Receiver	**2×2**	**RX1**	**RX2**
	2×4	**RX1, RX2**	**RX3, RX4**
Clipping level		2.5	2.5
Phase noise power		-25 dBc/Hz	-30 dBc/Hz
IQ imbalance amplitude/phase		0.03/1°	0.01/2.5°
Carrier frequency offset		0.02	0.01
Sampling clock offset		600 ppm	600 ppm

according to the new impairment level. In FIGURE 20, the training process with varying impairment levels is shown. It can be observed that the loss becomes large at the moment of change of the impairment level, but then decreases with further training. FIGURE 21 compares the performances of DLNN with and without transfer learning when the impairment level changes. It is shown that transfer learning effectively solves the issue of time-varying impairments.

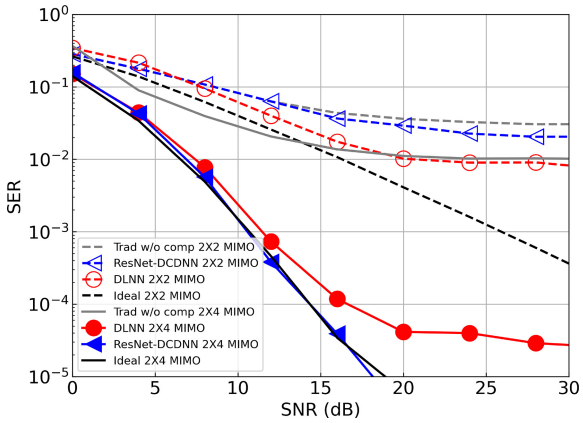


FIGURE 19. Performance comparison of DLNN and ResNet-DCDNN for 2×2 and 2×4 MIMO systems under the same setting (different hardware impairments at different antennas, Rayleigh fading channel).

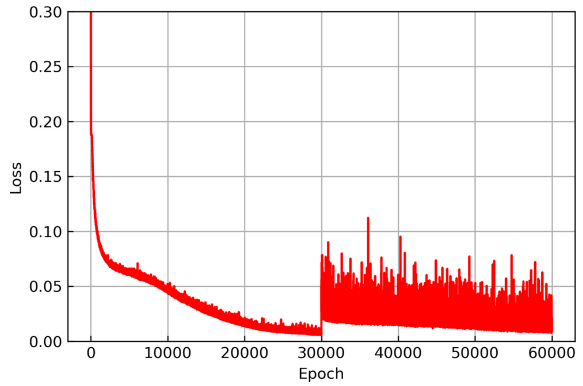


FIGURE 20. Training loss of DLNN with transfer learning under change of impairment levels.

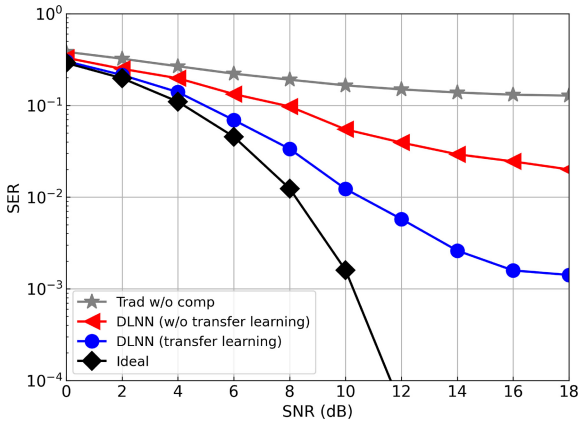


FIGURE 21. Comparison of performances of DLNN with and without transfer learning under change of impairment levels.

E. COMPLEXITY

The execution time and the computational complexity of the proposed designs and the non-DL algorithms are compared in TABLE 12, where N denotes the number of subcarriers and L denotes the length of the training sequence. When N is set to 64 and L is set to 2, the computational complexity of the non-DL algorithms is about 2^{25}. It can be seen that the proposed designs have lower complexity. In particular, the proposed DLNN is at least 6 times lower in execution time compared to the non-DL algorithms. Even with the extra neural network encoder and decoder, the complexity of the proposed deep learning-based OFDM system is lower than that of the OFDM system with traditional impairment-mitigation algorithms.

TABLE 12. Comparison of execution time and computational complexity of various methods.

Method	Time (sec)	Computation
Proposed DLNN	0.00215	3×2^{21}
Proposed ResNet-DCDNN	0.00862	2^{24}
Non-DL algorithms	0.0134	N^4L $(=2^{25}$ if $N=64,\ L=2)$

VI. CONCLUSION

In this paper, we have proposed two deep learning-based transmitter-receiver end-to-end designs for OFDM systems with hardware impairments. The simulation results have shown that the proposed DLNN and ResNet-DCDNN significantly outperform existing methods. In particular, the proposed DLNN has a superior performance for SISO and 2×2 MIMO systems, while the proposed ResNet-DCDNN has the advantage for 2×4 MIMO systems. Moreover, no matter the same or different hardware impairments are considered in MIMO systems, the proposed designs can deliver excellent performance. Finally, when the impairment level changes, transfer learning can fine-tune the weights of neural networks to achieve satisfactory performance.

REFERENCES

[1] H. Bolcskei, "MIMO-OFDM wireless systems: Basics, perspectives, and challenges," *IEEE Wireless Commun.*, vol. 13, no. 4, pp. 31–37, Aug. 2006.
[2] Z. Wu and X. Gao, "Improved MIMO-OFDM scheme for the next generation WLAN," *J. Syst. Eng. Electron.*, vol. 24, no. 1, pp. 52–59, Feb. 2013.
[3] M. Wu, D. Wüebben, A. Dekorsy, P. Baracca, V. Braun, and H. Halbauer, "Hardware impairments in millimeter wave communications using OFDM and SC-FDE," in *Proc. 20th Int. ITG Workshop Smart Antennas*, Mar. 2016, pp. 1–8.
[4] U. Gustavsson et al., "On the impact of hardware impairments on massive MIMO," in *Proc. IEEE GLOBECOM Workshops (GC Wkshps)*, Dec. 2014, pp. 294–300.
[5] S. Bucher, G. Yammine, R. F. H. Fischer, and C. Waldschmidt, "On the impact of hardware impairments in noncoherent massive MIMO systems," in *Proc. 24th Int. ITG Workshop Smart Antennas*, Feb. 2020, pp. 1–6.
[6] M. Morelli and M. Moretti, "Joint maximum likelihood estimation of CFO, noise power, and SNR in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 2, no. 1, pp. 42–45, Feb. 2013.
[7] P. K. Nishad and P. Singh, "Carrier frequency offset estimation in OFDM systems," in *Proc. IEEE Conf. Inf. Commun. Technol.*, Apr. 2013, pp. 885–889.
[8] W. Zhang, Q. Yin, and F. Gao, "Computationally efficient blind estimation of carrier frequency offset for MIMO-OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7644–7656, Nov. 2016.
[9] Z. Zhu, X. Huang, and H. Leung, "Blind compensation of frequency-dependent I/Q imbalance in direct conversion OFDM receivers," *IEEE Commun. Lett.*, vol. 17, no. 2, pp. 297–300, Feb. 2013.

[10] W. Zhang, R. C. de Lamare, C. Pan, and M. Chen, "Joint TX/RX IQ imbalance parameter estimation using a generalized system model," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 4704–4709.
[11] A. Leshem and M. Yemini, "Phase noise compensation for OFDM systems," *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5675–5686, Nov. 2017.
[12] M. D. Mir and A. S. Buttar, "Phase noise mitigation techniques in OFDM system: A survey," in *Proc. IEEE Int. Conf. Elect. Comput. Commun. Technol. (ICECCT)*, Mar. 2015, pp. 1–4.
[13] S. Bogana, "Compensation of IQ-imbalance and phase noise in MIMO-OFDM systems," 2012, *arXiv:1209.0061*.
[14] K.-H. Kim, H. Park, J.-S. No, H. Chung, and D.-J. Shin, "Clipping noise cancelation for OFDM systems using reliable observations based on compressed sensing," *IEEE Trans. Broadcast.*, vol. 61, no. 1, pp. 111–118, Mar. 2015.
[15] R. Rietman, J.-P. Linnartz, and E. P. de Vries, "Clip correction in wireless LAN receivers," in *Proc. Eur. Conf. Wireless Technol.*, Oct. 2008, pp. 174–177.
[16] E. Briggs, B. Nutter, and D. McLane, "Sample clock offset detection and correction in the LTE downlink," *J. Signal Process. Syst.*, vol. 69, pp. 31–39, Oct. 2012.
[17] T.-H. Sang and Y.-C. Xu, "Clipping noise compensation with neural networks in OFDM systems," *Signals*, vol. 1, no. 1, pp. 100–109, 2020.
[18] H. S. Park, E.-Y. Choi, Y. S. Song, S. Noh, and K. Seo, "DNN-based phase noise compensation for sub-THz communications," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 866–868.
[19] S. Kumari, K. K. Srinivas, and P. Kumar, "Channel and carrier frequency offset equalization for OFDM based UAV communications using deep learning," *IEEE Commun. Lett.*, vol. 25, no. 3, pp. 850–853, Mar. 2021.
[20] Z. Liu, C.-H. Lee, W. Xu, and S. Li, "Energy-efficient design for massive MIMO with hardware impairments," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 843–857, Feb. 2021.
[21] F. Horlin and A. Bourdoux, *Digital Compensation for Analog Front-Ends: A New Approach to Wireless Transceiver Design*. Hoboken, NJ, USA: Wiley, 2008.
[22] Y. G. Li and G. L. Stuber, *Orthogonal Frequency Division Multiplexing for Wireless Communications*. New York, NY, USA: Springer, 2006.
[23] B. Yang, Z. Yu, J. Lan, R. Zhang, J. Zhou, and W. Hong, "Digital beamforming-based massive MIMO transceiver for 5G millimeter-wave communications," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 7, pp. 3403–3418, Jul. 2018.
[24] M.-H. Jeong, T.-J. Lee, M.-J. Kim, and Y.-C. Ko, "Effect of phase noise in IEEE 802.11ad MIMO-OFDM systems with common/independent local oscillators," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2015, pp. 1332–1335.

CHENG-YU WU received the B.S. degree from the Department of Electrical and Computer Engineering, National Chiao Tung University, Taiwan, in 2019, and the M.S. degree from the Institute of Communications Engineering, National Yang Ming Chiao Tung University, Taiwan, in 2021. His main research includes deep learning and wireless communication systems.

YU-KAI LIN received the B.S. degree from the Department of Communication Engineering, National Central University, Taiwan, in 2020, and the M.S. degree from the Institute of Communications Engineering, National Yang Ming Chiao Tung University, Taiwan, in 2023. His main research includes deep learning and wireless communication systems.

CHUN-KUAN WU received the B.S. degree from the Department of Electrical and Computer Engineering, Tamkang University, Taiwan, in 2021. He is currently pursuing the M.S. degree with the Institute of Communications Engineering, National Yang Ming Chiao Tung University, Taiwan. His research interests include mmwave communications, beamforming, and deep learning in wireless communication systems.

CHIA-HAN LEE (Member, IEEE) received the B.S. degree in electrical engineering from National Taiwan University in 1999, the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, in 2003, and the Ph.D. degree in electrical engineering from Princeton University in 2008. From 1999 to 2001, he served as a Missile Operation Officer with ROC Army. From 2008 to 2009, he was a Postdoctoral Research Associate with the University of Notre Dame, USA. From 2010 to 2016, he was with Academia Sinica as an Assistant Research Fellow and then an Associate Research Fellow. In 2016, he joined National Yang Ming Chiao Tung University as an Associate Professor and the Hwa Tse Roger Liang Junior Chair Professor from 2018 to 2019, and became a Professor in 2019. His research interest is deep learning-based wireless communications and networks. He received Intel Labs Distinguished Collaborative Research Awards in 2014 and was named Intel Labs Distinguished Collaborator in 2015 (for years 2010–2015). He serves as the Industry Presentations and Demonstrations Co-Chair for IEEE GLOBECOM 2017, the Symposium Co-Chair for IEEE GLOBECOM 2019, ICC 2021, ICC 2022, and ICC 2023, the Tutorials Co-Chair for IEEE GLOBECOM 2020, and the TPC Co-Chair for IEEE GLOBECOM 2025. He was an Editor of IEEE COMMUNICATIONS LETTERS from 2014 to 2018 and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2014 to 2019. He has been an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS since 2019. He is the Chair of IEEE ComSoc Taipei Chapter from 2021 to 2022.