CHALMERS
UNIVERSITY OF TECHNOLOGY

Symbol-Based Over-the-Air Digital Predistortion Using Reinforcement Learning

Symbol-Based Over-the-Air Digital Predistortion Using Reinforcement Learning

Yibo Wu*†, Jinxiang Song†, Christian Häger†, Ulf Gustavsson*,
Alexandre Graell i Amat†, and Henk Wymeersch†
*Ericsson Research, Gothenburg, Sweden
†Department of Electrical Engineeering, Chalmers University of Technology, Gothenburg, Sweden

***Abstract*—We propose an over-the-air digital predistortion optimization algorithm using reinforcement learning. Based on a symbol-based criterion, the algorithm minimizes the errors between downsampled messages at the receiver side. The algorithm does not require any knowledge about the underlying hardware or channel. For a generalized memory polynomial power amplifier and additive white Gaussian noise channel, we show that the proposed algorithm achieves performance improvements in terms of symbol error rate compared with an indirect learning architecture even when the latter is coupled with a full sampling rate ADC in the feedback path. Furthermore, it maintains a satisfactory adjacent channel power ratio.**

I. Introduction

Digital predistortion (DPD) is a technique to linearize the nonlinear power amplifier (PA) in a radio frequency (RF) chain to achieve the best energy efficiency while maintaining the spectral mask requirement [1]. It is customary to implement DPD using parametric models. For simplicity, DPD parameters are mostly optimized at the transmitter side, which requires a feedback data acquisition path to collect the PA output signal [2]. To capture the full-band behavior of the PA for DPD optimization, high sampling rate feedback analog-to-digital converters (ADCs) in the feedback path are needed, which is challenging for wideband signals. High sampling rate ADCs add a huge cost, which increases linearly with the number of RF chains in massive multiple-input multiple-output (MIMO) [3], where each RF chain may require a separate feedback path [2].

To tackle the high cost problem of the feedback path, recent works have shifted toward low sampling rate ADC methods [4]–[7], where the feedback path is coupled with a low sampling rate ADC. These works focus on recovering the full-rate PA output signal from the undersampled output signal of the low-rate ADC, and the recovered signal is then used for DPD optimization. To reduce the cost of the feedback path further, over-the-air (OTA) methods have become a promising solution for the DPD optimization [8]–[10]. Instead of using a feedback path, OTA methods utilize an observation receiver to acquire the PA output signal over the channel for DPD optimization. Such methods achieve promising linearization performance and cost savings compared with methods using the feedback path.

Several parametric models for DPD have been utilized, e.g., Volterra-series-based [4]–[11] or neural network (NN)-based models [12]–[15]. To optimize the models, the works [8]–[15] consider a sample-based criterion that minimizes the sample error between oversampled signals. This is optimal when the aim is to minimize the ACPR and normalized mean squared error (NMSE). When the aim is to minimize the symbol error rate (SER), however, this approach is not optimal, and a symbol-based criterion is more appropriate.

A symbol-based criterion has been used to optimize the constellation mapper and demapper in [16] and the pulse-shaping/matched filters in [17], [18] in an end-to-end manner via supervised learning (SL), but SL requires differential models for all hardware and channel, which limits the usage in real communication systems. Instead, reinforcement learning (RL)-based optimization of the constellation mapper and demapper has been proposed in [19], [20], which does not require any models for hardware and channel. However, in [19], [20], the (de)mapper operates at the same symbol level as the optimization criterion, and no memory effects of hardware are considered. It requires a generalization for RL-based DPD optimization with a symbol-based criterion due to the data rate difference between DPD parameters and criterion, and memory effects in the PA. To the best of our knowledge, OTA DPD optimization using RL with a symbol-based criterion has not yet been addressed.

In this work, we propose a symbol-based DPD optimization algorithm using OTA observations. Instead of using a sample-based criterion that minimizes the error between oversampled signals, the proposed algorithm minimizes the cross-entropy between transmitted and received symbols. Using the policy gradient theorem [21], we generalize the work [19] by connecting the symbol-based criterion with the sample-based policy optimization, which allows to optimize DPD parameters at the sample level. For a generalized memory polynomial (GMP) PA and additive white Gaussian noise (AWGN) channel, we show that the proposed symbol-based DPD optimization algorithm achieves symbol error rate (SER) gains over the indirect learning architecture (ILA)-based DPD optimization algorithm [22] even with full-rate ADCs. Error spectrum results show that, although the DPD optimized by the symbol-based criterion focuses more on reducing the in-band errors (i.e., symbol errors), out-of-band errors are still at a satisfactory level.

Notation: Lowercase and uppercase boldface letters denote column vectors and matrices such as $\boldsymbol{x}$ and $\boldsymbol{X}$. $\mathbb{R}, \mathbb{R}_{\geq 0}$,

This work was supported by the Swedish Foundation for Strategic Research (SSF), grant no. I19-0021.

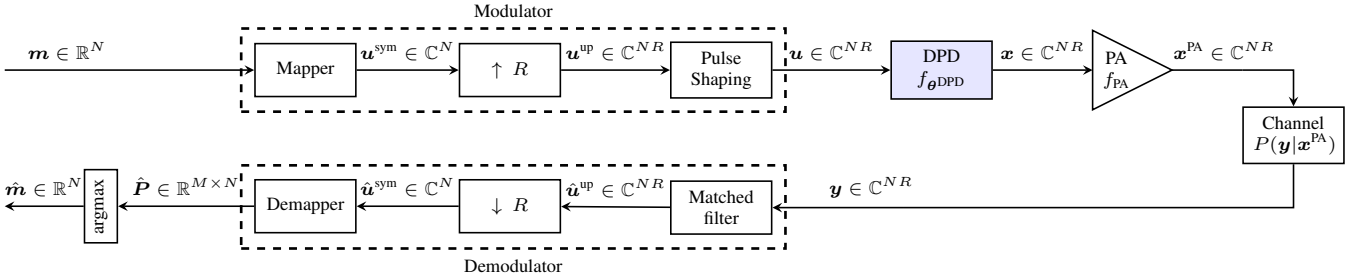

Fig. 1: System model of different blocks in a communication system with an unknown channel, where the block DPD linearizes the nonlinearity of the PA. The constellation mapping, upsampling, and pulse shaping are referred to as *modulator*. The constellation demapping, downsampling, and matched filtering are referred to as *demodulator*.

and $\mathbb{C}$ denote real, non-negative-real, and complex numbers, respectively. x_n or $[\boldsymbol{x}]_n$ denote the n-th element of $\boldsymbol{x}$, and $\boldsymbol{x}_{n:n+k}$ denotes a vector consisting of the n-th to $(n+k)$-th elements of $\boldsymbol{x}$. $\mathbb{E}_{\boldsymbol{x}}\{\cdot\}$ denotes the expectation operator taken over $\boldsymbol{x}$.

II. System model

A. System Model

We consider the communication system shown in Fig. 1. Let $\boldsymbol{m} \in \mathbb{R}^N$ be a message sequence of length N, generated from a message set $\mathbb{M} = \{1, ..., M\}$. A message sequence $\boldsymbol{m}$ is mapped to a sequence of symbols $\boldsymbol{u}^{\text{sym}} \in \mathbb{C}^N$ via a constellation mapping, upsampled with upsampling rate R to $\boldsymbol{u}^{\text{up}} \in \mathbb{R}^{NR}$, and pulse-shaped to a discrete-time baseband signal $\boldsymbol{u} = [u_1, ..., u_n, u_{NR}]^{\text{T}} \in \mathbb{C}^{NR}$, where u_n is the sample transmitted at time instant n. To compensate for the nonlinearity of the PA, DPD is applied to $\boldsymbol{u}$. The DPD is represented by a parametric model $f_{\boldsymbol{\theta}^{\text{DPD}}} : \mathbb{C}^{K_1+1} \to \mathbb{C}$ with parameters $\boldsymbol{\theta}^{\text{DPD}}$ and input memory length K_1. Given the input sequence $\boldsymbol{u}_{n:n-K_1} = [u_n, ..., u_{n-K_1}]^{\text{T}} \in \mathbb{C}^{K_1+1}$ to the DPD, the predistorted output x_n can be expressed as

$$x_n = f_{\boldsymbol{\theta}^{\text{DPD}}}(u_n, ..., u_{n-K_1}) = f_{\boldsymbol{\theta}^{\text{DPD}}}(\boldsymbol{u}_{n:n-K_1}). \tag{1}$$

The predistorted signal $\boldsymbol{x} = [x_1, ..., x_n, x_{NR}]^{\text{T}}$ is then amplified by the PA, which is represented by the nonlinear function $f_{\text{PA}} : \mathbb{C}^{K_2+1} \to \mathbb{C}$ with memory length K_2, input $\boldsymbol{x}_{n:n-K_2} = [x_n, ..., x_{n-K_2}]^{\text{T}} \in \mathbb{C}^{K_2+1}$, and output x_n^{PA}. The input–output relation of the PA can be expressed as

$$x_n^{\text{PA}} = f_{\text{PA}}(x_n, ..., x_{n-K_2}) = f_{\text{PA}}(\boldsymbol{x}_{n:n-K_2}). \tag{2}$$

The signal $\boldsymbol{x}^{\text{PA}} = [x_1^{\text{PA}}, ..., x_n^{\text{PA}}, x_{NR}^{\text{PA}}]^{\text{T}}$ is then sent through a discrete channel with conditional distribution $P(\boldsymbol{y}|\boldsymbol{x}^{\text{PA}})$, where $\boldsymbol{y}$ is the channel output. Note that elements in $\boldsymbol{u}_{n:n-K_1}$ and $\boldsymbol{x}_{n:n-K_1}$ with nonpositive indexes are set to zero. Under the assumption of perfect synchronization, the received signal $\boldsymbol{y}$ is demodulated via a matched filter and downsampled with a downsampling rate R to symbols $\hat{\boldsymbol{u}}^{\text{sym}} \in \mathbb{C}^N$. Each symbol $\hat{u}_n^{\text{sym}}$ is decoded to a probability vector $\hat{\boldsymbol{p}}_n \in \mathbb{R}_+^M$ over M messages, where $\sum_{i=1}^M \hat{p}_i = 1$. Finally, the estimated message is obtained by $\hat{m}_n = \arg\max_m [\hat{\boldsymbol{p}}_n]_m$. Overall, the estimated message sequence $\hat{\boldsymbol{m}}$ is obtained from the probability matrix $\hat{\boldsymbol{P}} = [\hat{\boldsymbol{p}}_1, ..., \hat{\boldsymbol{p}}_N] \in \mathbb{R}_+^{M \times N}$. The demapper can be implemented by an NN as in [16], [19], [20], which can be pretrained to have similar decoding performance as the maximum likelihood demapper.

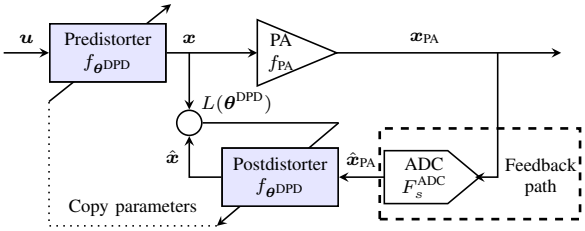

Fig. 2: Block diagram of the ILA. Learning the inverse behavior of the PA by minimizing waveform errors, a postdistorter is optimized, which is then utilized as the predistorter.

Given a parametric DPD model $f_{\boldsymbol{\theta}^{\text{DPD}}}$ with parameters $\boldsymbol{\theta}^{\text{DPD}}$, our objective is to optimize $\boldsymbol{\theta}^{\text{DPD}}$ with respect to a given loss function $\mathcal{L}$,

$$\hat{\boldsymbol{\theta}}^{\text{DPD}} = \underset{\boldsymbol{\theta}^{\text{DPD}}}{\arg\min}\ \mathcal{L}(\boldsymbol{\theta}^{\text{DPD}}). \tag{3}$$

B. Indirect Learning Architecture

The ILA [22] indirectly optimizes the DPD parameters by estimating an inverse behavior of the PA as shown in Fig. 2. The optimization of the parameters of the distorter is conducted after the PA, and hence the distorter is referred to as the *postdistorter*. After the parameter optimization, the postdistorter is used as the *predistorter*, placed before the PA. To optimize the postdistorter, a feedback path is required for signal acquisition of the PA. Here we consider an ADC in the feedback path with a sampling rate F_s^{ADC}. We denote the output of the ADC by $\hat{\boldsymbol{x}}^{\text{PA}}$. The parameters of the postdistorter are usually optimized by minimizing the mean squared error (MSE) between the postdistorter output signal $\hat{\boldsymbol{x}}$ and the PA input signal $\boldsymbol{x}$. In this case, the loss function $\mathcal{L}$ can be expressed as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}^{\text{DPD}}) &= \mathbb{E}_{\boldsymbol{x}}\left\{|x_n - \hat{x}_n|^2\right\} \\ &= \mathbb{E}_{\boldsymbol{x}}\left\{|x_n - f_{\boldsymbol{\theta}^{\text{DPD}}}(\hat{\boldsymbol{x}}_{n:n-K_1}^{\text{PA}}))|^2\right\},\end{aligned} \tag{4}$$

where $\hat{\boldsymbol{x}}_{n:n-K_1}^{\text{PA}} = [\hat{x}_n^{\text{PA}}, ..., \hat{x}_{n-K_1}^{\text{PA}}]^{\text{T}}$ is the input of the postdistorter with memory K_1. Substituting (4) into (3), $\boldsymbol{\theta}^{\text{DPD}}$ is optimized by minimizing the sample difference between the postdistoter output $\hat{\boldsymbol{x}}$ and the PA output $\hat{\boldsymbol{x}}^{\text{PA}}$ collected by the feedback path, i.e., the optimization is based on a *sample-based* criterion.

III. Proposed Symbol-Based DPD Optimization Algorithm

We propose to optimize the DPD based on a *symbol-based* criterion using OTA measurements. Specifically, the symbol-based criterion minimizes the cross-entropy between the transmitted and received messages $\boldsymbol{m}$ and $\hat{\boldsymbol{m}}$.

A. Symbol-Based Criterion

To implement the symbol-based criterion for DPD optimization, we consider a symbol-based cross-entropy loss function, which defines the transmitted message sequence $\boldsymbol{m}$ and received message probabilities $\hat{\boldsymbol{P}}$ as

$$\mathcal{L}(\boldsymbol{\theta}^{\text{DPD}}) = -\mathbb{E}_{\boldsymbol{m}}\{\underbrace{\log([\hat{\boldsymbol{p}}_n]_{m_n})}_{l_n^{\text{CE}}}\}, \tag{5}$$

where the loss function between the transmitted message m_n and the corresponding vector of probabilities $\hat{\boldsymbol{p}}_n$ is denoted by l_n^{CE}, referred to as the cross-entropy per-example loss.

B. Supervised Learning

Assuming that all blocks in the communication system are differentiable, the derivatives of the loss function at the receiver side with respect to any trainable parameter in the system can be analytically calculated using the chain rule. Thus, $\boldsymbol{\theta}^{\text{DPD}}$ can be updated through back-propagation via a mini-batch stochastic gradient descent (SGD) algorithm as

$$\boldsymbol{\theta}_{j+1}^{\text{DPD}} = \boldsymbol{\theta}_{j}^{\text{DPD}} - \eta \nabla_{\boldsymbol{\theta}^{\text{DPD}}} \mathcal{L}(\boldsymbol{\theta}_{j}^{\text{DPD}}), \tag{6}$$

where $\eta > 0$ denotes the learning rate and $\nabla_{\boldsymbol{\theta}^{\text{DPD}}} \mathcal{L}(\boldsymbol{\theta}_{j}^{\text{DPD}})$ is the derivative of $\mathcal{L}$ with respect to $\boldsymbol{\theta}_{j}^{\text{DPD}}$ at step j.

However, in a real communication system, most of the blocks are non-differentiable, and thus it is infeasible to apply the chain rule to calculate $\nabla_{\boldsymbol{\theta}^{\text{DPD}}} \mathcal{L}$. Although we can circumvent this problem using surrogate parametric models of the hardware components, e.g., pretrained PA model in [23], it is cumbersome to pretrain such models, and the performance highly depends on the model accuracy.

C. Reinforcement Learning

RL is defined as a learning process of how an *agent* takes *actions* in an environment to minimize a given loss [21]. The optimization of the DPD parameters can be viewed through the lens of a RL problem. The DPD acts as an agent that takes actions following a *policy*, which is optimized to minimize the loss $\mathcal{L}$. RL has already been used in the context of communications, e.g., to optimize the constellation mapper and demapper [19], [20]. Here, we consider a different scenario with more components, which raises the problem of how to optimize the DPD parameters $\boldsymbol{\theta}^{\text{DPD}}$ at the sample-based using a loss function $\mathcal{L}$ at the symbol-based.

As shown in Fig. 3, we consider a Gaussian policy $\pi(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ that converts the deterministic actions $\boldsymbol{x}$ to stochastic actions $\tilde{\boldsymbol{x}}$, which enables the *exploration* of possible actions. For an arbitrary action x_n, we consider an independent Gaussian policy

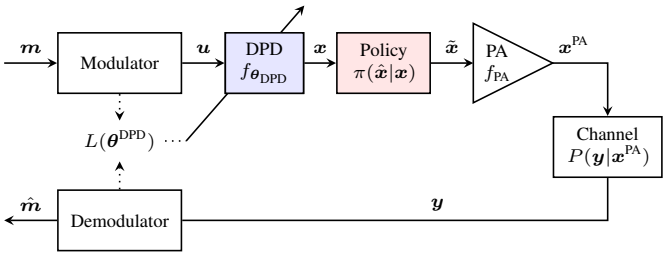

Fig. 3: Block diagram of the proposed symbol-based optimization for DPD. The DPD parameters $\boldsymbol{\theta}^{\text{DPD}}$ are optimized using OTA observations at the receiver side by minimizing a symbol-based cross-entropy between symbols.

$\pi(\tilde{x}_n|x_n)$, which generates output $\tilde{x}_n$ by adding a Gaussian perturbation $w \sim \mathcal{CN}(0, \sigma_\pi^2)$ as

$$\tilde{x}_n = \sqrt{1-\sigma_\pi^2} x_n + w, \tag{7}$$

where σ_π^2 is the variance of the perturbation, which is a hyper-parameter that is fixed during the training. Thus, the policy $\pi(\tilde{x}_n|x_n)$ is the probability density function (PDF) of a complex Gaussian variable with mean $\sqrt{1-\sigma_\pi^2} x_n = \sqrt{1-\sigma_\pi^2} f_{\boldsymbol{\theta}^{\text{DPD}}}(\boldsymbol{u}_{n:n-K_1})$ and variance σ_π^2,

$$\pi(\tilde{x}_n|x_n) \propto \exp\left(-\frac{\left| \tilde{x}_n - \sqrt{1-\sigma_\pi^2} f_{\boldsymbol{\theta}^{\text{DPD}}}(\boldsymbol{u}_{n:n-K_1}) \right|^2}{\sigma_\pi^2} \right). \tag{8}$$

Based on the received observation y_n, the receiver can compute the corresponding per-example loss l_n by l_n^{CE}. The loss l_n is related to a subset of the entire sequence $\boldsymbol{x}$ because of the memory effects of the PA and convolution operation in the matched filtering. Denote this subset by $\boldsymbol{x}_{(G)} = \{x_{nR-G}, ..., x_{nR+G}\}$, where G denotes the number of signals being related. Because of the convolution operation in the pulse shaping, the subset $\boldsymbol{x}_{(G)}$ depends on a subset of the messages in $\boldsymbol{m}$, denoted by $\boldsymbol{m}_{(G)} = \{m_{nR-G}, ..., m_{nR+G}\}$. Similarly, we can define $\tilde{\boldsymbol{x}}_{(G)} = \{\tilde{x}_{nR-G}, ..., \tilde{x}_{nR+G}\}$ and $\boldsymbol{y}_{(G)} = \{y_{nR-G}, ..., y_{nR+G}\}$.

The objective of the DPD agent is to minimize the loss function $\mathcal{L}(\boldsymbol{\theta}^{\text{DPD}})$, defined as

$$\mathcal{L}(\boldsymbol{\theta}^{\text{DPD}}) \triangleq \mathbb{E}_{\boldsymbol{m},\tilde{\boldsymbol{x}},\boldsymbol{y}}\{l_n\}, \tag{9}$$

where l_n is l_n^{CE} (see (5)). In order to minimize the loss function, we compute the gradient with respect to $\boldsymbol{\theta}^{\text{DPD}}$ according to the following proposition.

Proposition 1: The gradient of $\mathcal{L}(\boldsymbol{\theta}^{\text{DPD}})$ with respect to $\boldsymbol{\theta}^{\text{DPD}}$ is approximated by

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}^{\text{DPD}}} \mathcal{L}(\boldsymbol{\theta}^{\text{DPD}}) \\ &\approx \frac{2\sqrt{1-\sigma_\pi^2}}{N\sigma_\pi^2} \sum_{n=1}^{N} l_n \sum_{g=-G}^{G} \left(\tilde{x}_n - \sqrt{1-\sigma_\pi^2} f_{\boldsymbol{\theta}^{\text{DPD}}}(\boldsymbol{u}_{n:n-K_1}) \right) \\ &\times \nabla_{\boldsymbol{\theta}^{\text{DPD}}} f_{\boldsymbol{\theta}^{\text{DPD}}}(\boldsymbol{u}_{n:n-K_1}). \end{aligned} \tag{10}$$

Proof: Exploiting the policy gradient theorem [21] and using the fact that $\nabla \log(\pi) = (\nabla \pi)/\pi$, we can write

$$\nabla_{\boldsymbol{\theta}^{\text{DPD}}} \mathcal{L}(\boldsymbol{\theta}^{\text{DPD}}) = \mathbb{E}_{\boldsymbol{m},\tilde{\boldsymbol{x}},\boldsymbol{y}}\{l_n \ \nabla_{\boldsymbol{\theta}^{\text{DPD}}} \log(\pi(\tilde{\boldsymbol{x}}|\boldsymbol{x}))\}. \tag{11}$$

Algorithm 1 : Symbol-based optimization for $\boldsymbol{\theta}^{\text{DPD}}$.

Input: N,N_{B} σ_π^2, R, G

1: **for** A number of iterations N_{B} **do**
2: Messages to symbols: $\boldsymbol{m} \rightarrow \boldsymbol{u}^{\text{sym}}$
3: Upsampled and pulse-shaped with an upsampling rate R: $\boldsymbol{u}^{\text{sym}} \rightarrow \boldsymbol{u}^{\text{up}} \rightarrow \boldsymbol{u}$
4: DPD: $f_{\boldsymbol{\theta}^{\text{DPD}}}(\boldsymbol{u}) \rightarrow \boldsymbol{x}$ via (1)
5: Policy: $\pi(\cdot|\boldsymbol{x}) \rightarrow \tilde{\boldsymbol{x}}$ via (7)
6: PA: $f_{\text{PA}}(\tilde{\boldsymbol{x}}) \rightarrow \boldsymbol{x}^{\text{PA}}$ via (2)
7: Channel: $P(\cdot|\boldsymbol{x}^{\text{PA}}) \rightarrow \boldsymbol{y}$
8: Matched filtered and downsampled with a downsampling rate R: $\boldsymbol{y} \rightarrow \hat{\boldsymbol{u}}^{\text{up}} \rightarrow \hat{\boldsymbol{u}}^{\text{sym}}$
9: Symbols to messages: $\hat{\boldsymbol{u}}^{\text{sym}} \rightarrow \hat{\boldsymbol{m}}$
10: Per-example losses: l_n
11: Update $\boldsymbol{\theta}^{\text{DPD}}$: SGD$(\boldsymbol{\theta}^{\text{DPD}}, \mathcal{L}) \rightarrow \boldsymbol{\theta}^{\text{DPD}}$ via (6) and (10)
12: **end for**
13: Remove policy $\pi(\cdot)$

The loss l_n is related to a fraction of $\boldsymbol{x}$ consisting of G signals. Restricting (11) to this subset of signals, we can approximate it as

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{m},\tilde{\boldsymbol{x}},\boldsymbol{y}}\left\{l_n \, \nabla_{\boldsymbol{\theta}^{\text{DPD}}} \log\left(\pi\left(\tilde{\boldsymbol{x}}_{(G)}|\boldsymbol{x}_{(G)}\right)\right)\right\} \qquad (12)\\ &= \mathbb{E}_{\boldsymbol{m},\tilde{\boldsymbol{x}},\boldsymbol{y}}\Big\{l_n \, \nabla_{\boldsymbol{\theta}^{\text{DPD}}} \log\Big(\prod_{g=-G}^{G} \pi(\tilde{x}_{nR+g}|x_{nR+g})\Big)\Big\}, \end{aligned}$$

where (12) follows as the conditional probability $\pi(\tilde{\boldsymbol{x}}_{(G)}|\boldsymbol{x}_{(G)})$ reduces to the product of conditional probabilities $\pi(\tilde{x}_{nR+g}|x_{nR+g})$ due to the independence of each action. Now, using the fact that the product of logarithms can be written as the logarithm of a sum and approximating the expectation by the average of N (correlated) samples from the underlying distribution $p(\boldsymbol{m}, \tilde{\boldsymbol{x}}, \boldsymbol{y})$, we obtain

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}^{\text{DPD}}} \mathcal{L}(\boldsymbol{\theta}^{\text{DPD}}) \qquad (13)\\ &\approx \frac{1}{N}\sum_{n=1}^{N} l_n \sum_{g=-G}^{G} \nabla_{\boldsymbol{\theta}^{\text{DPD}}} \log(\pi(\tilde{x}_{nR+g}|x_{nR+g})) \, . \end{aligned}$$

From (8), it follows that

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}^{\text{DPD}}} \log(\pi(\tilde{x}_{nR+g}|x_{nR+g})) \qquad (14)\\ &= \frac{2\sqrt{1-\sigma_\pi^2}}{\sigma_\pi^2}\big(\tilde{x}_n - \sqrt{1-\sigma_\pi^2} f_{\boldsymbol{\theta}^{\text{DPD}}}(\boldsymbol{u}_{n:n-K_1})\big)\nabla_{\boldsymbol{\theta}^{\text{DPD}}} f_{\boldsymbol{\theta}^{\text{DPD}}}(\cdot). \end{aligned}$$

Substituting (14) into (13) completes the proof. ∎

The approximation of the gradient of the loss function $\mathcal{L}$ with respect to $\boldsymbol{\theta}^{\text{DPD}}$ in Proposition 1 can be used to optimize $\boldsymbol{\theta}^{\text{DPD}}$ via any SGD-based algorithm as in (6).

The details of the symbol-based DPD optimization procedure of $\boldsymbol{\theta}^{\text{DPD}}$ are given in Algorithm 1. First, each batch of N messages, $\boldsymbol{m}$, is generated and transformed to a sequence of symbols $\boldsymbol{u}^{\text{sym}}$, then upsampled and pulse-shaped to $\boldsymbol{u}$ with an upsampling rate R. After the DPD model $f_{\boldsymbol{\theta}^{\text{DPD}}}(\cdot)$, the Gaussian policy is applied by adding a perturbation w to generate exploration samples $\tilde{\boldsymbol{x}}$ as in (7). These samples are sent through the PA and channel. The output of the channel, $\boldsymbol{y}$, is transformed to $\hat{\boldsymbol{u}}^{\text{sym}}$ by the matched filter, downsampled with a downsampling rate R, and eventually decoded to $\hat{\boldsymbol{m}}$. Then, the cross-entropy per-example losses, l_n^{CE}, are calculated, which are assumed to be known at the transmitter via a reliable feedback channel [19], [20], [24]. Finally, $\boldsymbol{\theta}^{\text{DPD}}$ is updated by an SGD-based algorithm as in (6), where $\nabla_{\boldsymbol{\theta}^{\text{DPD}}}\mathcal{L}(\boldsymbol{\theta}^{\text{DPD}})$ is computed using (10). The whole procedure is iterated for a number of iterations N_{B}, and the policy $\pi(\cdot)$ is removed.

TABLE I: Parameter setup

M	R	N	G	σ_π^2	σ_{ch} [V]
64	4	1024	3	0.08	0.3

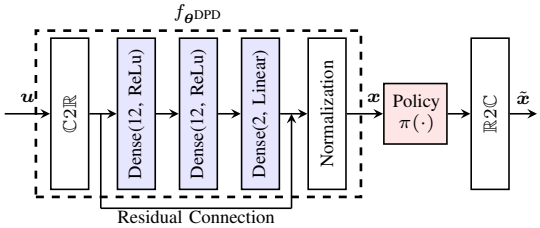

Fig. 4: Structure of the DPD model $f_{\boldsymbol{\theta}^{\text{DPD}}}$ using R2TDNN [15], which is placed before the policy block. The normalization block aligns the average output power of DPD to be the same as its input.

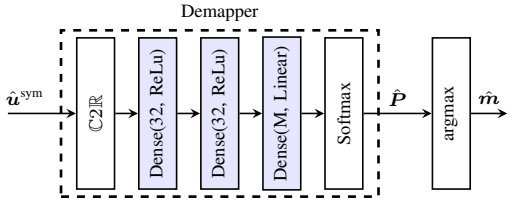

Fig. 5: Structure of the NN-based 64 QAM decoder, whose parameters are pretrained and frozen during the training of DPD.

IV. Numerical Results

A. Setup

1) Parameters: We consider a GMP model as the PA with nonlinear orders $K_a = K_b = K_c = 7$, memory lengths $L_a = L_b = Lc = 3$, and cross-term lengths $M_b = M_c = 1$ [11, Eq. (23)]. The corresponding parameters are estimated from the measurements of the RF WebLab[1] using the ILA and a 55 MHz signal. The measured saturation point and measurement noise standard deviation of the PA are 20.9 V ($\approx$ 36.4 dBm) and 0.053 V. We consider a 50 Ω load impedance. The remaining parameters are given in Table I. The number of training messages in $\boldsymbol{m}$ for each batch is $N = 1024$. The optimizer for gradient descent is Adam [25] with a learning rate 0.001. The messages are mapped to a sequence of symbols $\boldsymbol{u}^{\text{sym}}$ according to a $M = 64$ quadrature amplitude modulation (QAM) constellation. The sampling rate is 200 MHz, with upsampling and downsampling rate $R = 4$. The pulse shaping

[1]The RF WebLab is a PA measurement setup that can be remotely accessed at www.dpdcompetition.com.

filter is a root-raised cosine (RRC) filter with a roll-off factor 0.1, so the bandwidth of the baseband signal $\boldsymbol{u}$ is 55 MHz. The peak-to-average power ratio (PAPR) of the signal is 10.3 dB, which is similar to that of an orthogonal frequency division multiplexing (OFDM) signal. We set $G = 3$ in (10), and the perturbation variance $\sigma_\pi^2 = 0.08$. We consider an AWGN channel with fixed noise standard deviation $\sigma_{\text{ch}} = 0.3$ V. We consider a simulated ADC in the feedback path of the ILA with infinite resolution but three different sampling rates $F_s^{\text{ADC}} = 110$ MHz, 220 MHz, and 550 MHz (referred to full-rate ADC). Note that F_s^{ADC} needs to be larger than the Nyquist rate (110 MHz) of the distortion-free 55 MHz signal to capture the out-of-band behavior of the PA output signal.

2) Model Structures: For the DPD model, we choose both the GMP [11] and the residual real-valued time-delay neural network (R2TDNN), where the latter is from our previous work [15] and shows to outperform many other NNs and GMP for DPD in terms of complexity versus performance. We consider the same parameter settings (i.e., nonlinear order, memory length, and cross-term length) for the GMP DPD. The specific structure of R2TDNN along with policy $\pi(\cdot)$ is shown in Fig. 4. The block $\mathbb{C}2\mathbb{R}$ transforms the complex-valued signal, $\boldsymbol{u}$, to a real-valued signal. We consider two hidden layers with 12 neurons each, and 3 input memory length as the same as the PA model. The output of the linear layer is added with the input via the residual connection and then normalized by the normalization layer, which ensures that the average output power of the DPD is the same as its average input power.

We consider an NN-based $M = 64$ QAM decoder as shown in Fig. 5, which are trained to have similar performance as the maximum likelihood decoder. The learned detector is frozen during the training of the DPD. Specifically, the softmax layer outputs a probability vector over M messages, where the largest probability represents the predicted message.

B. Simulation results

1) SER versus average PA output power: Fig. 6 shows the testing SER results versus the average PA output power for the cases of no DPD, ILA-optimized NN DPD [15] with under-sampling ($F_s^{\text{ADC}} = 110$ and 220 MHz) and full-rate ($F_s^{\text{ADC}} = 550$ MHz) ADCs in the feedback path, ILA-optimized GMP DPD with full-rate ADC [11], the proposed RL-optimized DPD, the case with linear-clipping PA [26],[2] and the theoretical SER bound of 64 QAM, where the average energy per bit is converted to the average PA output power considering a distortion-free PA. The RL-optimized DPD is optimized using the cross-entropy loss function. Note that the policy $\pi(\cdot)$ is removed once the training ends, and the input of the PA becomes $\boldsymbol{x}$. The number of QAM symbols used to calculate the SER is 10^7.

We observe that the SER curves exhibit two different behaviors. When the average PA output power is below 29 dBm, most of the PA input signal $\boldsymbol{x}$ is in the linear region

[2]The linear-clipping PA has a linear behavior before the clipping region, which has the minimum distortion that any DPDs can achieve.

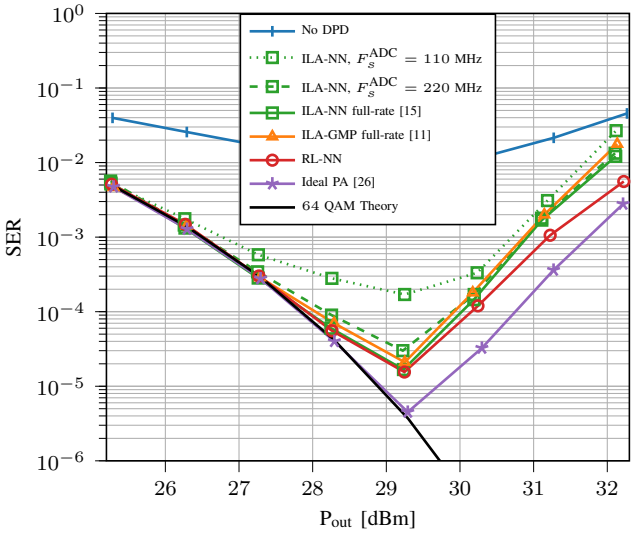

Fig. 6: SER as a function of the average PA output power.

of the PA, and the SER of all scenarios with DPD improve quickly with increasing average PA output power P_{out}. The proposed RL-optimized NN DPD achieves similar SER as the ILA-optimized NN and GMP DPDs with full-rate ADCs, and has substantial SER gain over the ILA-optimized NN DPDs with undersampling ADCs. As the average PA output power increases above 29 dBm, $\boldsymbol{x}$ starts to be clipped by the saturation region of the PA. While the clipping effect makes the SER of all DPD cases degrade rapidly, we note that the RL-optimized DPD exhibits considerable SER gains over other ILA-optimized DPDs at the highly nonlinear region of the PA, i.e., 30 dBm $< P_{\text{out}} <$ 32 dBm. This indicates the advantage of the symbol-based criterion for DPD optimization over the sample-based criterion in terms of SER. Overall, the results prove the effectiveness of the sample-based DPD optimization using a symbol-based criterion without knowing the hardware and channel for a PA with memory and the AWGN channel. The SER gap between all the DPD cases and the linear-clipping case may come from some residual distortions due to irreversible nonlinearity.

2) In-band and Out-of-band Errors: Fig. 7 shows the error spectrum (i.e., the power spectral density (PSD) of the difference between the real and ideal PA output signals) of schemes in Fig. 6 at the average PA output power $P_{\text{out}} =$ 30.2 dBm. The corresponding NMSE and ACPR results are presented in Table II.

Due to the aliasing and band-limiting effects of the undersampling ADC, the linearization performance of the ILA-NN DPD with $F_s^{\text{ADC}} = 110$ MHz is affected severely, exhibiting large in-band and out-of-band errors compared with DPDs of full-rate ADCs and even larger out-of-band errors (-35.4 dBc ACPR) compared with No DPD case (-37 dBc ACPR). With a full-rate ADC, the power spectral errors of the ILA-NN are reduced (-38.7 dB NMSE and -40.2 dBc ACPR). As expected, the in-band spectral error of the RL-NN-based DPD is better than its out-of-band

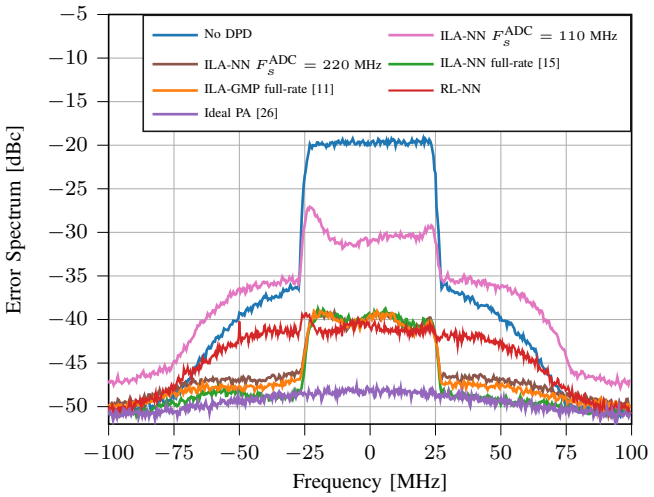

Fig. 7: Error spectrum between the actual and desired PA output signals of schemes in Fig. 6 at the average PA output power $P_{\text{out}} = 30.2$ dBm.

TABLE II: NMSE and ACPR results of cases in Fig. 7 at the average PA output power $P_{\text{out}} = 30.2$ dBm.

	NMSE [dB]	ACPR [dBc]
No DPD	-22.8	-37.0
ILA-NN, $F_s^{\text{ADC}} = 110$ MHz	-28.3	-35.4
ILA-NN, $F_s^{\text{ADC}} = 220$ MHz	-37.9	-39.2
ILA-NN, full-rate [15]	-38.7	-40.2
ILA-GMP, full-rate [11]	-38.2	-39.8
RL-NN	-35.0	-38.1
Ideal PA [26]	-42.3	-41.3

spectral error since the optimization criterion, i.e., symbol-based criterion, focuses more on the in-band errors. Nevertheless, the RL-NN-based DPD still maintains a satisfactory ACPR (-38.1 dBc) considering that the lower bound is -41.3 dBc.

V. Conclusion

We proposed a novel DPD optimization algorithm that optimizes DPD parameters using a symbol-based criterion at the receiver side instead of a sample-based criterion at the transmitter side, which avoids the cumbersome feedback path in the transmitter. The proposed optimization algorithm, based on RL, does not require a model for the hardware or channel, which is an attractive feature in practice. Exploiting the policy gradient theorem, we connect the symbol-based criterion with the sample-based policy optimization. The proposed algorithm is verified by simulation results for a GMP PA modeled from a real PA over the AWGN channel. The proposed RL-NN-based DPD achieves SER gains over the ILA-based DPD even the latter uses a full-rate ADC in the feedback path, while it also maintains satisfactory out-of-band errors. It is expected to see a performance improvement for a real PA as the performance of ILA is limited by the nature of its identification process. Some limitations of this work are the AWGN channel and hardware models, which can be generalized to a more realistic scenario considering more realistic channels and more hardware impairments (e.g., quadrature imbalance).

References

[1] S. C. Cripps, *RF power amplifiers for wireless communications*. Artech house Norwood, MA, 2006, vol. 2.

[2] X. Liu *et al.*, "Beam-oriented digital predistortion for 5G massive MIMO hybrid beamforming transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 7, pp. 3419–3432, May, 2018.

[3] E. G. Larsson *et al.*, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[4] Y. Liu *et al.*, "Novel technique for wideband digital predistortion of power amplifiers with an under-sampling ADC," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 11, pp. 2604–2617, Oct. 2014.

[5] Z. Wang *et al.*, "Low feedback sampling rate digital predistortion for wideband wireless transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 11, pp. 3528–3539, Sep. 2016.

[6] N. Guan *et al.*, "Digital predistortion of wideband power amplifier with single undersampling ADC," *IEEE Microw. Wireless Compon. Lett.*, vol. 27, no. 11, pp. 1016–1018, Sep. 2017.

[7] Y. Beltagy *et al.*, "Direct learning algorithm for digital predistortion training using sub-Nyquist intermediate frequency feedback signal," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 1, pp. 267–277, Nov. 2018.

[8] K. Hausmair *et al.*, "Modeling and linearization of multi-antenna transmitters using over-the-air measurements," in *Proc. IEEE ISCAS'18*. IEEE, May. 2018, pp. 1–4.

[9] X. Liu *et al.*, "Linearization for hybrid beamforming array utilizing embedded over-the-air diversity feedbacks," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 12, pp. 5235–5248, Oct. 2019.

[10] X. Wang *et al.*, "Real-time single channel over-the-air data acquisition for digital predistortion of 5G massive MIMO wireless transmitters," in *Proc. MTT-S IWS, IEEE*, May, 2019, pp. 1–3.

[11] D. R. Morgan *et al.*, "A generalized memory polynomial model for digital predistortion of RF power amplifiers," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3852–3860, Oct. 2006.

[12] T. Liu *et al.*, "Dynamic behavioral modeling of 3G power amplifiers using real-valued time-delay neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 3, pp. 1025–1033, Mar. 2004.

[13] M. Isaksson *et al.*, "Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 11, pp. 3422–3428, Nov. 2005.

[14] M. Rawat *et al.*, "Adaptive digital predistortion of wireless power amplifiers/transmitters using dynamic real-valued focused time-delay line neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 1, pp. 95–104, Jan. 2010.

[15] Y. Wu *et al.*, "Residual neural networks for digital predistortion," in *Proc. IEEE Globecom '20*, Dec. 7-11 2020, pp. 1–6.

[16] T. O'Shea *et al.*, "An introduction to deep learning for the physical layer," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[17] B. Karanov *et al.*, "End-to-end deep learning of optical fiber communications," *J. Lightw. Techn.*, vol. 36, no. 20, pp. 4843–4855, Aug. 2018.

[18] F. A. Aoudia *et al.*, "Waveform learning for next-generation wireless communication systems," *arXiv preprint arXiv:2109.00998*, 2021.

[19] ——, "End-to-end learning of communications systems without a channel model," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput*, Oct. 2018, pp. 298–303.

[20] ——, "Model-free training of end-to-end communication systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2503–2516, Aug. 2019.

[21] R. S. Sutton *et al.*, *Reinforcement learning: An introduction*. MIT press, 2018.

[22] C. Eun *et al.*, "A new Volterra predistorter based on the indirect learning architecture," *IEEE Trans. Signal Process.*, vol. 45, no. 1, pp. 223–227, Jan. 1997.

[23] H. Paaso *et al.*, "Comparison of direct learning and indirect learning predistortion architectures," in *IEEE Int. Symp. Wireless Commun. Syst.* IEEE, Oct. 2008, pp. 309–313.

[24] V. Raj *et al.*, "Backpropagating through the air: Deep learning at physical layer without channel models," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2278–2281, Nov. 2018.

[25] D. P. Kingma *et al.*, "Adam: A method for stochastic optimization," *Proc. ICLR*, 2015.

[26] J. Chani-Cahuana *et al.*, "Lower bound for the normalized mean square error in power amplifier linearization," *IEEE Microw. Wireless Compon. Lett.*, vol. 28, no. 5, pp. 425–427, May. 2018.