



## **Deep-Learning-Based Channel Estimation for Distributed MIMO with 1-bit Radio-Over-Fiber Fronthaul**

Downloaded from: <https://research.chalmers.se>, 2025-01-18 08:08 UTC

Citation for the original published paper (version of record):

Bordbar, A., Aabel, L., Häger, C. et al (2024). Deep-Learning-Based Channel Estimation for Distributed MIMO with 1-bit Radio-Over-Fiber Fronthaul. Proceedings of the International Symposium on Wireless Communication Systems. <http://dx.doi.org/10.1109/ISWCS61526.2024.10639157>

N.B. When citing this work, cite the original published paper.

# Deep-Learning-Based Channel Estimation for Distributed MIMO with 1-bit Radio-Over-Fiber Fronthaul

Alireza Bordbar<sup>1</sup>, Lise Aabel<sup>1,2</sup>, Christian Häger<sup>1</sup>, Christian Fager<sup>1</sup>, and Giuseppe Durisi<sup>1</sup>

<sup>1</sup>Chalmers University of Technology, 41296 Gothenburg, Sweden

<sup>2</sup>Ericsson Research, 41756 Gothenburg, Sweden

**Abstract**—We consider the problem of pilot-aided, uplink channel estimation in a distributed massive multiple-input multiple-output (MIMO) architecture, in which the access points are connected to a central processing unit via fiber-optical fronthaul links, carrying a two-level-quantized version of the received analog radio-frequency signal. We adapt to this architecture the deep-learning-based channel-estimation algorithm recently proposed by Nguyen *et al.* (2023), and explore its robustness to the additional signal distortions (beyond 1-bit quantization) introduced in the considered architecture by the automatic gain controllers (AGCs) and by the comparators. These components are used at the access points to generate the two-level analog waveform from the received signal. Via simulation results, we illustrate that the proposed channel-estimation method outperforms the Busgang linear minimum mean-square error channel estimator, and it is robust against the additional impairments introduced by the AGCs and the comparators.

## I. INTRODUCTION

In distributed massive multiple-input multiple-output (MIMO), a large number of remotely located access points (APs) serve a much smaller number of user equipments (UEs) in a coordinated way. Coordination is enabled by a central processing unit (CPU), which is connected to the APs via fronthaul links. With such an architecture, one can exploit macro-diversity and mitigate path-loss variations compared to co-located massive MIMO architectures. This yields a more uniform quality of service [1].

In the analysis of distributed MIMO systems, it is often assumed that the APs are equipped with local oscillators, which are used for up- and down-conversion, and that samples of the base-band (BB) signals are exchanged between the APs and the CPU over the fronthaul links. However, these local oscillators need to be synchronized for reciprocity-based joint coherent downlink beamforming to work, which may be costly or even unfeasible in certain deployments [2].

To solve this issue, an alternative architecture has been proposed in the literature (see, e.g., [3]–[7]), in which up- and down-

conversion are performed digitally at the CPU. This, however, implies that radio frequency (RF) signals need to be exchanged over the fronthaul links. To limit complexity and fronthaul requirements, in the architecture proposed in [3]–[7], only a two-level representation of the RF signal is exchanged over the fronthaul link. This implies that quantizers with just a single-bit resolution can be deployed at the CPU. The testbed measurements reported in, e.g., [6], [7] demonstrate that satisfactory performance can be achieved despite the nonlinearities introduced by this architecture, provided that the 1-bit quantizers at the CPU operate at a sufficiently high sampling rate—roughly two orders of magnitude larger than the bandwidth of the transmitted signal.

This paper focuses on the problem of acquiring accurate channel estimates via uplink pilot transmission, to enable reciprocity-based coherent downlink beamforming. The problem of channel estimation in the presence of the impairments caused by 1-bit quantization is well-studied in the literature, although most works focus on the scenario in which sampling is performed on the BB signal and not the RF signal. Relevant contributions involve the derivation of the linear minimum mean-squared error (LMMSE) estimator via Busgang decomposition [8] as well as of methods for the efficient evaluation of the maximum *a posteriori* and maximum likelihood (ML) channel estimators, by exploiting the convexity of the log-likelihood functions [9], [10]. More recently, several methods relying on deep neural networks have been proposed in the literature (see, e.g., [11] and references therein).

In this paper, we will focus on the machine-learning algorithm proposed in [11], where deep unfolding of the iterations of a first-order optimization method is used to tackle the ML channel-estimation problem. Specifically, we adapt the algorithm proposed in [11] to the 1-bit radio-over-fiber fronthaul architecture considered in this paper. Our extension accounts for the real-valued nature of the sampled signals, and for oversampling and dithering. Furthermore, we analyze the robustness of the resulting algorithm to the additional impairments introduced in our architecture by the automatic gain controller (AGC), which limits the range of the received signal power at which dithering is effective, and by the comparator, which introduces random bit flips when the difference between the signals at its input ports

This work was supported in part by the Swedish Foundation for Strategic Research (SSF), under grants ID19-0036 and FUS21-0004. The simulations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

is small.

## II. SYSTEM MODEL

### A. Distributed MIMO with 1-Bit Radio-Over-Fiber Fronthaul

In Fig. 1, we illustrate the key components of the distributed MIMO architecture with 1-bit radio-over-fiber fronthaul, and single-antenna APs, described in [6], [7].

*Downlink:* In the downlink, modulated and precoded BB signals intended for the UEs are up-converted digitally to the carrier frequency at the CPU, and then band-pass (BP) sigma-delta modulated. This yields an oversampled binary representation of the RF signals. Sigma-delta modulation relies on oversampling and noise shaping to place the quantization distortion outside the bandwidth of the desired signal [12]. Analog signals are then generated via 1-bit digital-to-analog converters (DACs). Subsequently, these signals are converted into the optical domain by electrical-to-optical (E/O) converters and conveyed to the desired APs via optical fibers. A reconstruction band-pass filter (BPF) at the AP suppresses the out-of-band distortion produced by the sigma-delta modulator and recovers the underlying (un-quantized) RF waveform, which is then transmitted over the single antenna of each AP.

*Uplink:* The signal received at the antenna port of each AP is fed to a BPF and then passed through a low noise amplifier (LNA) and an AGC. The output of the AGC is fed to the positive differential input port of a comparator (denoted in the figure by ‘‘Comp’’). The negative differential input port of the comparator is fed with a dither signal, which is provided to the AP via the downlink fiber-optical fronthaul. Specifically, this dither signal is generated digitally at the CPU, sigma-delta modulated, and recovered at the AP via a low-pass filter (LPF). The AGC has the important function to regulate the power of the received signal, in order to make dithering effective. The output of the comparator is converted into the optical domain and then conveyed to the CPU, where it is converted back to the electrical domain and then digitalized via a 1-bit analog-to-digital converter (ADC). The resulting samples are then digitally down-converted and further processed at the digital signal processing (DSP) unit.

The testbed described in [7] has a maximum bandwidth of 100 MHz, operates at a carrier frequency of 2.35 GHz, and involves a sampling rate of 10 GS/s. As detailed in [7], this architecture can be implemented using low-cost, off-the-shelf components.

### B. Pilot-Aided Channel Estimation: a Mathematical Model

We next consider the problem of uplink pilot-based channel estimation in the distributed MIMO architecture described in Fig. 1, and provide a mathematical model for the received signal, which we will use in Section III to generalize the channel-estimation algorithm proposed in [11].

To keep the notation compact, we will focus on the problem of estimating the channel between one arbitrary AP and  $U$  UEs.<sup>1</sup>

<sup>1</sup>The mathematical model we provide in this section can be readily extended to the joint estimation of the channel between an arbitrary number of APs and UEs. However, such generalization appears to be superfluous, since, in distributed MIMO, the channels corresponding to different APs are typically assumed to be independent.

We assume that the UEs transmit pilot signals of bandwidth  $W$ , centered at a frequency  $f_c \gg W$ . We let  $f_s \geq 2f_c + W$  denote the sampling rate at the ADCs and also assume that pilot transmission involves  $N_p$  signals (where, as we shall see,  $N_p$  is related to the number of pilot symbols transmitted per UE) of duration  $T = N/f_s$  seconds, for some integer  $N$ . Under these assumptions, we can write the resulting  $N \times N_p$  samples at the output of the 1-bit ADC as

$$\mathbf{Z} = \text{sgn}(\mathbf{Y}^{\text{rf}} + \mathbf{D}). \quad (1)$$

Here, the  $N \times N_p$  matrix  $\mathbf{Y}^{\text{rf}}$  contains the samples of the discrete-time RF signal received at the AP (after the BPF), taken at the sampling frequency  $f_s$ ; the  $N \times N_p$  matrix  $\mathbf{D}$  contains the samples of the dither signal added at the AP; finally,  $\text{sgn}$  denotes the sign function, which models 1-bit quantization, and is applied entrywise to the matrix  $\mathbf{Y}^{\text{rf}} + \mathbf{D}$ . For analytical convenience, we shall model the entries of the dither-signal matrix as independent  $\mathcal{N}(0, E_d/2)$  random variables. Furthermore, we model  $\mathbf{Y}^{\text{rf}}$  as

$$\mathbf{Y}^{\text{rf}} = \sqrt{2}\Re\{\text{diag}(\mathbf{u})\mathbf{Y}^{\text{bb}}\} \quad (2)$$

where the  $N$ -dimensional vector  $\mathbf{u}$ , whose  $n$ th entry is given by  $u_n = e^{j2\pi(f_c/f_s)n}$ ,  $n = 0, \dots, N-1$ , models the up-conversion operation, and  $\mathbf{Y}^{\text{bb}}$  is the complex envelope of the discrete-time received signal.

We let  $S = WT$ , and assume for simplicity that  $S$  is an odd integer. Furthermore, we define the following set:

$$S = \{0, 1, \dots, (S-1)/2, N - (S-1)/2, N - (S-1)/2 + 1, \dots, N-1\}. \quad (3)$$

To account for oversampling, we model  $\mathbf{Y}^{\text{bb}}$  as

$$\mathbf{Y}^{\text{bb}} = \mathbf{F}^{\text{inv}}(\mathbf{H}\mathbf{P} + \mathbf{W}). \quad (4)$$

Here, the  $S \times U$  matrix  $\mathbf{H}$  contains the channel coefficients (expressed in the frequency domain) that need to be estimated; the  $U \times N_p$  matrix  $\mathbf{P}$  contains the pilot symbols, which we assume have power  $E_s$ ; the  $S \times N_p$  matrix  $\mathbf{W}$ , whose entries are drawn independently from a  $\mathcal{CN}(0, N_0)$  distribution, denotes the additive Gaussian noise; finally, the  $N \times S$  matrix  $\mathbf{F}^{\text{inv}}$  is a truncated inverse discrete Fourier transform (IDFT) matrix, obtained by removing all columns of a  $N \times N$  IDFT matrix whose indices do not belong to the set  $S$  defined in (3). Note that, in our notation, the oversampling ratio is given by  $N/S$ .

It will turn out convenient to vectorize the matrix  $\mathbf{Z}$  in (1). Specifically, we set  $\mathbf{z} = \text{vec}(\mathbf{Z}) \in \mathbb{R}^{N N_p \times 1}$ . Similarly, we vectorize also  $\mathbf{Y}^{\text{rf}}$ ,  $\mathbf{D}$ ,  $\mathbf{Y}^{\text{bb}}$ ,  $\mathbf{H}$ , and  $\mathbf{W}$ , obtaining the vectors  $\mathbf{y}^{\text{rf}}$ ,  $\mathbf{d}$ ,  $\mathbf{y}^{\text{bb}}$ ,  $\mathbf{h}$ , and  $\mathbf{w}$ . Using this notation, we can equivalently express (2) and (4) as

$$\mathbf{y}^{\text{rf}} = \sqrt{2}\Re\{\mathbf{U}\mathbf{y}^{\text{bb}}\} \quad (5)$$

and

$$\mathbf{y}^{\text{bb}} = \tilde{\mathbf{F}}^{\text{inv}}(\tilde{\mathbf{P}}\mathbf{h} + \mathbf{w}). \quad (6)$$

Here,  $\mathbf{U} = \mathbf{I}_{N_p} \otimes \text{diag}(\mathbf{u})$ ,  $\tilde{\mathbf{F}}^{\text{inv}} = \mathbf{I}_{N_p} \otimes \mathbf{F}^{\text{inv}}$ , and  $\tilde{\mathbf{P}} = \mathbf{P}^T \otimes \mathbf{I}_S$ , where  $\otimes$  indicates the Kronecker product. To summarize, we can compactly express the output signal in vectorized form as

$$\mathbf{z} = \text{sgn}\left(\sqrt{2}\Re\{\mathbf{U}\tilde{\mathbf{F}}^{\text{inv}}(\tilde{\mathbf{P}}\mathbf{h} + \mathbf{w})\} + \mathbf{d}\right). \quad (7)$$

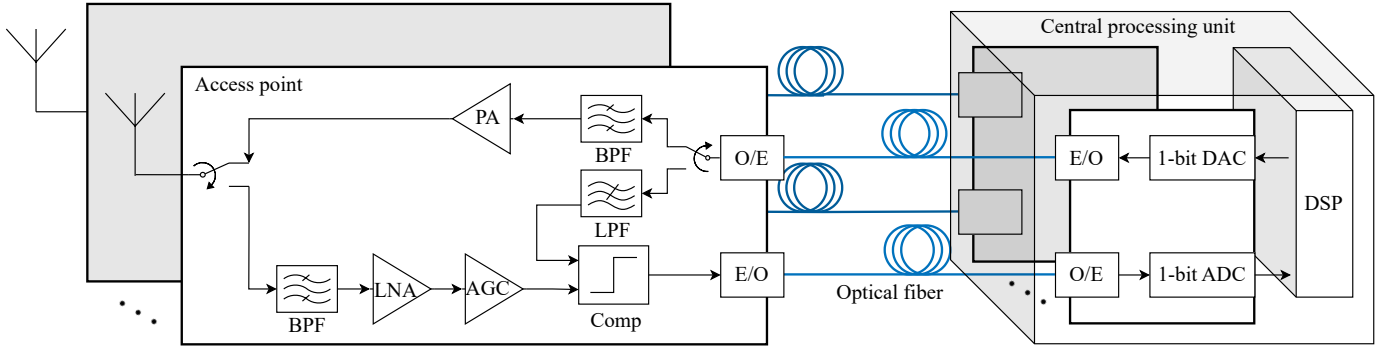


Fig. 1. Block diagram of one AP and the CPU in the distributed MIMO architecture with 1-bit radio-over-fiber fronthaul considered in the paper.

### III. DEEP-LEARNING-BASED CHANNEL ESTIMATION

#### A. Maximum-Likelihood Channel Estimation

We start by noting that the input-output relation (7) contains two random vectors, i.e.,  $\mathbf{w} \in \mathbb{C}^{NN_p \times 1}$  and  $\mathbf{d} \in \mathbb{R}^{NN_p \times 1}$ . The  $NN_p$  entries of  $\mathbf{U}\tilde{\mathbf{F}}^{\text{inv}}\mathbf{w}$  are correlated, which makes calculating the ML estimator mathematically difficult. To work around this problem, we ignore the additive noise vector  $\mathbf{w}$  while deriving the ML solution to the channel estimation problem.<sup>2</sup> Let

$$\mathbf{M} = \mathbf{U}\tilde{\mathbf{F}}^{\text{inv}}\tilde{\mathbf{P}} = [\mathbf{m}_1, \dots, \mathbf{m}_{NN_p}]^T. \quad (8)$$

Then, by ignoring  $\mathbf{w}$ , we can write (7) as

$$\mathbf{z} = \text{sgn}\left(\sqrt{2}\Re\{\mathbf{M}\mathbf{h}\} + \mathbf{d}\right). \quad (9)$$

Let  $p(\mathbf{z}|\mathbf{h})$  be the conditional probability of  $\mathbf{z}$  in (9) given  $\mathbf{h}$ . Then the ML channel estimate  $\hat{\mathbf{h}}$  is given by

$$\begin{aligned} \hat{\mathbf{h}} &= \arg \max_{\mathbf{h}} p(\mathbf{z}|\mathbf{h}) \\ &= \arg \min_{\mathbf{h}} \sum_{i=1}^{NN_p} -\log \left[ \Phi\left(\sqrt{\rho}(q_i^{\text{up}} - \Re\{\mathbf{m}_i^T \mathbf{h}\})\right) \right. \\ &\quad \left. - \Phi\left(\sqrt{\rho}(q_i^{\text{low}} - \Re\{\mathbf{m}_i^T \mathbf{h}\})\right) \right]. \end{aligned} \quad (10)$$

Here,  $\Phi(\cdot)$  denotes the cumulative distribution function of the normal distribution,  $\rho = \sqrt{2}/E_d$ , and  $q_i^{\text{up}}, q_i^{\text{low}} \in \{-\infty, 0, \infty\}$  denote the upper and lower thresholds of the quantization bin to which the  $i$ th entry  $z_i$  of the vector  $\mathbf{z}$  belongs.

It is now crucial to realize that the optimization problem in (10) is convex [11], [13], [10]. Let  $f(\mathbf{h})$  be the objective function in (10). Since  $f(\mathbf{h})$  is convex, we can solve (10) via an iterative gradient descent algorithm. However, the gradient of  $f(\mathbf{h})$  is undefined at certain points [11]. In addition, a lack of a closed-form expression for  $\Phi(\cdot)$  makes the optimization process more involved. To address these problems, we follow [11] and approximate  $\Phi(\cdot)$  as

$$\Phi(x) \approx \sigma(cx) = \frac{1}{1 + e^{-cx}} \quad (11)$$

<sup>2</sup>The impact of this assumption will be discussed in Section V.

with  $c = 1.702$ . Using (11), we can approximate  $f(\mathbf{h})$  as

$$\begin{aligned} \tilde{f}(\mathbf{h}) &= \sum_{i=1}^{NN_p} -\log \left[ \sigma\left(c\sqrt{\rho}(q_i^{\text{up}} - \Re\{\mathbf{m}_i^T \mathbf{h}\})\right) \right. \\ &\quad \left. - \sigma\left(c\sqrt{\rho}(q_i^{\text{low}} - \Re\{\mathbf{m}_i^T \mathbf{h}\})\right) \right]. \end{aligned} \quad (12)$$

To perform iterative gradient descent on  $\tilde{f}(\mathbf{h})$ , we need to compute the vector-valued Wirtinger derivative [14, Cor. 5.0.1]

$$\begin{aligned} \frac{\partial \tilde{f}(\mathbf{h})}{\partial \mathbf{h}^*} &= \sum_{i=1}^{NN_p} c\sqrt{\rho} \left[ 1 - \sigma\left(c\sqrt{\rho}(q_i^{\text{up}} - \Re\{\mathbf{m}_i^T \mathbf{h}\})\right) \right. \\ &\quad \left. - \sigma\left(c\sqrt{\rho}(q_i^{\text{low}} - \Re\{\mathbf{m}_i^T \mathbf{h}\})\right) \right] \frac{\partial \Re\{\mathbf{m}_i^T \mathbf{h}\}}{\partial \mathbf{h}^*}. \end{aligned} \quad (13)$$

Note that

$$\begin{aligned} \frac{\partial \Re\{\mathbf{m}_i^T \mathbf{h}\}}{\partial \mathbf{h}^*} &= \frac{\partial}{\partial \mathbf{h}^*} \left( \frac{1}{2} [(\mathbf{m}_i^H \mathbf{h}^*)^T + \mathbf{m}_i^T \mathbf{h}] \right) \\ &= \mathbf{m}_i^*/2. \end{aligned} \quad (14)$$

Setting  $\mathbf{q}^{\text{up}} = [q_1^{\text{up}}, \dots, q_{NN_p}^{\text{up}}]^T$ ,  $\mathbf{q}^{\text{low}} = [q_1^{\text{low}}, \dots, q_{NN_p}^{\text{low}}]^T$  and substituting (14) into (13), we can write  $\partial \tilde{f}(\mathbf{h})/\partial \mathbf{h}^*$  as

$$\begin{aligned} \frac{\partial \tilde{f}(\mathbf{h})}{\partial \mathbf{h}^*} &= \frac{c\sqrt{\rho}}{2} \mathbf{M}^* \left[ \mathbf{1} - \sigma\left(c\sqrt{\rho}(\mathbf{q}^{\text{up}} - \Re\{\mathbf{M}\mathbf{h}\})\right) \right. \\ &\quad \left. - \sigma\left(c\sqrt{\rho}(\mathbf{q}^{\text{low}} - \Re\{\mathbf{M}\mathbf{h}\})\right) \right] \end{aligned} \quad (15)$$

where  $\sigma(\cdot)$  is applied element-wise to its vector-valued input.

We can now write the iterative gradient descent algorithm for minimizing the objective function  $\tilde{f}(\mathbf{h})$  as

$$\mathbf{h}^{(\ell)} = \mathbf{h}^{(\ell-1)} + \alpha^{(\ell)} \frac{\partial \tilde{f}(\mathbf{h})}{\partial \mathbf{h}^*} \Big|_{\mathbf{h}=\mathbf{h}^{(\ell-1)}} \quad (16)$$

where  $\ell$  is the current iteration index and  $\alpha^{(\ell)}$  is the step size.

### B. Structure of the deep neural network (DNN)

Following [11], we use the deep unfolding technique [15] to implement each iteration of (16) as a DNN layer. Each layer  $\ell \in \{1, \dots, L\}$  of the network receives  $\mathbf{h}^{(\ell-1)} \in \mathbb{C}^{SU \times 1}$  and calculates the gradient in (16) as

$$\alpha^{(\ell)} \mathbf{M}^* \left[ \mathbf{1} - \sigma \left( \beta^{(\ell)} (\mathbf{q}^{\text{up}} - \Re\{\mathbf{M}\mathbf{h}^{(\ell-1)}\}) \right) - \sigma \left( \beta^{(\ell)} (\mathbf{q}^{\text{low}} - \Re\{\mathbf{M}\mathbf{h}^{(\ell-1)}\}) \right) \right] \quad (17)$$

where  $\alpha^{(\ell)} \in \mathbb{R}$  and  $\beta^{(\ell)} \in \mathbb{R}$  are the trainable parameters in the  $\ell$ th layer. Note that the constants  $(c\sqrt{\rho})/2$  and  $c\sqrt{\rho}$  in (15) are absorbed into the trainable parameters in (17). Training  $\alpha^{(\ell)}$  and  $\beta^{(\ell)}$  corresponds to learning the step size and controlling the accuracy of the approximation in (11).

### IV. REFINED SYSTEM MODEL

As pointed out in [7, Figs. 11 and 12], the input/output relation in (7) does not match measurement results, since it ignores the impact of the AGC in the architecture presented in Fig. 1, as well as the distortion introduced by the comparator. In this section, we enhance the system model in (7) to include the effect of these two components.

*AGC:* The AGC is implemented in [7] via a feedback-controlled variable gain amplifier, which maintains the output power at  $-53$  dBW within a 45 dB dynamic range. Specifically, the maximum gain of the AGC is 15 dB. Let  $P^{\text{rf}} = \text{Tr}(\mathbb{E}[\mathbf{y}^{\text{rf}}(\mathbf{y}^{\text{rf}})^T]) / N$  be the average power of  $\mathbf{y}^{\text{rf}}$ . We model the variable gain in dB introduced by the AGC as

$$G(P^{\text{rf}}) = \begin{cases} 15 \text{ dB}, & \text{if } P^{\text{rf}} < -68 \text{ dBW} \\ -30 \text{ dB}, & \text{if } P^{\text{rf}} > -23 \text{ dBW} \\ -P^{\text{rf}} - 53 \text{ dB}, & \text{otherwise.} \end{cases} \quad (18)$$

The input to the comparator can then be modeled as  $a(P^{\text{rf}}) \mathbf{y}^{\text{rf}} + \mathbf{d}$ , where  $a(P^{\text{rf}}) = 10^{G(P^{\text{rf}})/20}$ .

*Comparator:* As discussed in [7], the comparator in Fig. 1 produces a lowpass-filtered version of a two-level waveform, because of its bandwidth limitations. As illustrated in [7, Fig. 12], when the synthesized waveform deviates significantly from the expected two-level waveform, the bits at the output of the ADCs are essentially generated at random, due to limitations in the decision circuitry. Typically, this occurs whenever the magnitude of the signal at the input of the comparator is small. To model this effect, we let  $\mathbf{B}$  be a diagonal matrix whose diagonal entries are drawn uniformly at random from the set  $\{-1, 1\}$  if the corresponding entry of the vector  $a(P^{\text{rf}}) \mathbf{y}^{\text{rf}} + \mathbf{d}$  is smaller in absolute value than a threshold  $t$ , and are set to 1 otherwise. In our simulations, we set  $t = 2.6 \cdot 10^{-4}$ , because this value matches the measurements reported in [7].

To summarize, in the proposed refined system model, the input-output relation (7) is replaced by

$$\mathbf{z} = \mathbf{B} \text{sgn} \left( a(P^{\text{rf}}) \sqrt{2} \Re \left\{ \mathbf{U} \tilde{\mathbf{F}}^{\text{inv}} \left( \tilde{\mathbf{P}} \mathbf{h} + \mathbf{w} \right) \right\} + \mathbf{d} \right). \quad (19)$$

TABLE I

THE THREE DIFFERENT UPLINK MODELS USED FOR GENERATING DATA TO TRAIN AND TEST THE DNN-BASED CHANNEL ESTIMATOR.

Model	Description
1	Simplified input-output relation (see (9))
2	Input-output relation without AGC effects and ADC bit-flips (see (7))
3	Refined input-output relation (see (19))

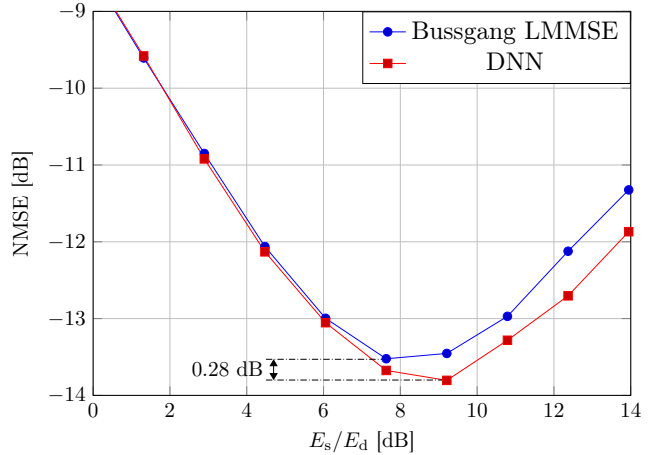


Fig. 2. NMSE as a function of signal-to-dither ratio  $E_s/E_d$  for Bussgang LMMSE and DNN-based channel estimators.

### V. SIMULATION RESULTS

*Simulation Parameters:* Throughout this section, we assume that a single UE ( $U = 1$ ) transmits  $N_p = 10$  pilot symbols with power  $E_s$  to the AP. We also assume that  $W = 240$  MHz,  $f_c = 2.4$  GHz,  $f_s = 10$  GS/s,  $S = 9$ , and  $N = 189$ . Finally, we let  $\mathbf{h} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_S)$ . The DNN used for channel estimation consists of  $L = 7$  layers. Denoting  $\hat{\mathbf{h}} = \mathbf{h}^{(L)}$ , we take as loss function the  $\ell_2$  norm of  $\mathbf{h} - \hat{\mathbf{h}}$ . We use the Adam optimizer [16] with a learning rate that starts at 0.002 and is multiplied by 0.95 after every 100 training epochs. The size of each training batch is 1000, and the number of training epochs is 1000. The performance metric used to evaluate and compare the channel estimation performance is the normalized mean square error (NMSE) averaged over 5000 channel realizations. The initial values for  $\alpha^{(\ell)}$  and  $\beta^{(\ell)}$  are 1.0 and 5.0, respectively. Furthermore,  $\mathbf{h}^{(1)} = \mathbf{0}$ .

In Table I, we summarize the three different system models that are used for evaluating the performance of the proposed channel estimation algorithm.

*Impact of Dithering:* We start by considering Model 1, where we ignore the additive noise  $\mathbf{w}$ , and investigate the impact of the dither signal on the NMSE. The network is trained for 20  $E_s/E_d$  values, uniformly spaced in the interval  $[-5 \text{ dB}, 25 \text{ dB}]$ . The results are shown in Fig. 2. We see that, for Model 1, the proposed DNN-based channel estimator outperforms the Bussgang LMMSE (BLMMSE) estimator by 0.28 dB at the respective optimal  $E_s/E_d$  values for BLMMSE (7.6 dB) and DNN (9.2 dB). These results are in line with the ones reported in [11, Fig. 7(a)] for a different channel model.



*Impact of Additive Noise:* Next, we evaluate the test performance of a DNN trained at the optimal  $E_s/E_d$  value in Fig. 2 using Model 1, for the case in which the test samples are generated according to Model 2. In Fig. 3 we show the NMSE as a function of  $E_s/N_0$ . We observe that ignoring the additive noise during the training phase of the DNN (red curve) does not degrade its performance when noise is present. Indeed, the achieved NMSE value is in agreement with the optimal  $E_s/E_d$  value reported in Fig. 2 for sufficiently large  $E_s/N_0$ . The DNN-based ML estimator outperforms the Bussgang LMMSE (black curve) by around 0.22 dB on average in the interval [20 dB, 50 dB] when additive noise is present.

*Impact of AGC and Random Bit Flips:* Next, we investigate the impact of the additional impairments described in Section IV. We consider two additional scenarios: i) We test the trained DNN at the optimal  $E_s/E_d$  obtained from Fig. 2 using Model 1 on a dataset generated from Model 3. ii) We train the DNN on datasets generated using Model 3, by considering 30  $E_s/N_0$  values, spaced uniformly in the interval [0 dB, 50 dB]; the ratio  $E_s/E_d$  is still chosen according to Fig. 2. Then, we test each trained DNN on a new dataset generated from Model 3 using the same  $E_s/N_0$ .

The NMSE achieved for these two additional scenarios is also shown in Fig. 3. We first notice that the DNN network trained on the dataset that does not take into account the effects of the AGC and ADC random bit flips (blue curve) performs poorly when  $E_s/N_0$  is outside the dynamic range of the AGC. Indeed, in agreement with (18), outside this interval, the AGC cannot enforce the optimal signal-to-dither ratio. Furthermore, averaged over the interval [20 dB, 40 dB], we see around 0.22 dB performance degradation compared to the case when Model 2 is used (red curve), due to the effect of the random bit flips.

Interestingly, we also see that if we train the DNN on data generated according to the refined input–output relation (19) (green curve), the performance is increased, despite the DNN possessing only 14 trainable parameters, and despite the impairments not been explicitly modeled when deriving the gradient update step.

## VI. CONCLUSIONS

We have considered the problem of pilot-based uplink channel estimation in the distributed MIMO 1-bit radio-over-fiber architecture recently demonstrated in [7]. Specifically, we have adapted to this architecture the deep-unfolding-based ML channel estimation algorithm recently proposed in [11], and analyzed its robustness to the additional impairments introduced in the considered architecture by the AGC (dynamic range) and the comparator (random bit flips). In future works, we will measure over-the-air the performance achievable with the proposed algorithm using the testbed described in [7], and explore additional data-driven methods for channel estimation and data detection.

## REFERENCES

[1] Ö. T. Demir, E. Björnson, and L. Sanguinetti, *Foundations of user centric cell-free massive MIMO*, ser. Foundations and Trends in Signal Processing, now Publishers, 2020, vol. 14, no. 3-4.

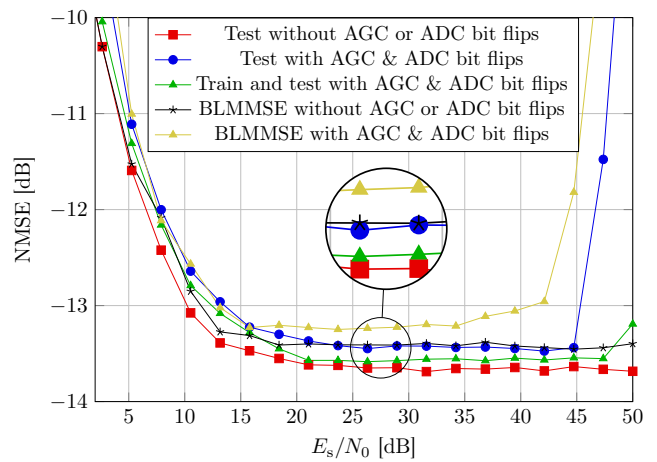


Fig. 3. The effect of AGC and ADC bit flips on the performance of the proposed DNN-based channel estimator.

[2] E. G. Larsson, “Massive synchrony in distributed antenna systems,” *IEEE Trans. Signal Process.*, pp. 855–866, Jan. 2024.

[3] R. F. Cordeiro, A. Prata, A. S. Oliveira, J. M. Vieira, and N. B. De Carvalho, “Agile all-digital RF transceiver implemented in FPGA,” *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 11, pp. 4229–4240, Nov. 2017.

[4] J. Wang, Z. Jia, L. A. Campos, and C. Knittle, “Delta-Sigma modulation for next generation fronthaul interface,” *J. Lightw. Technol.*, vol. 37, no. 12, pp. 2838–2850, Jun. 2019.

[5] C.-Y. Wu, H. Li, O. Caytan, J. Van Kerrebrouck, L. Breyne, J. Bauwelink, P. Demeester, and G. Torfs, “Distributed multi-user MIMO transmission using real-time Sigma-Delta-over-fiber for next generation fronthaul interface,” *J. Lightw. Technol.*, vol. 38, no. 4, pp. 705–713, Feb. 2020.

[6] I. C. Sezgin, L. Aabel, S. Jacobsson, G. Durisi, Z. S. He, and C. Fager, “All-digital, radio-over-fiber, communication link architecture for time-division duplex distributed antenna system,” *J. Lightw. Technol.*, vol. 39, no. 9, pp. 2769–2779, Feb. 2021.

[7] L. Aabel, S. Jacobsson, M. Coldrey, F. Olofsson, G. Durisi, and C. Fager, “A TDD distributed MIMO testbed using a 1-bit radio-over-fiber fronthaul architecture,” *IEEE Trans. Microw. Theory Techn.*, 2024.

[8] Y. Li, C. Tao, G. Seco-Granados, A. Mezghani, A. L. Swindlehurst, and L. Liu, “Channel estimation and performance analysis of one-bit massive MIMO systems,” *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4075–4089, Aug. 2017.

[9] J. Choi, J. Mo, and R. W. Heath Jr, “Near maximum-likelihood detector and channel estimator for uplink multiuser massive MIMO systems with one-bit ADCs,” *IEEE Trans. Commun.*, vol. 64, no. 5, pp. 2005–2018, May 2016.

[10] C. Studer and G. Durisi, “Quantized massive MU-MIMO-OFDM uplink,” *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2387–2399, Jun. 2016.

[11] L. V. Nguyen, D. H. N. Nguyen, and A. L. Swindlehurst, “Deep learning for estimation and pilot signal design in few-bit massive MIMO systems,” *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 379–392, Jan. 2023.

[12] S. Pavan, R. Schreier, and G. C. Temes, *Understanding Delta-Sigma Data Converters*, 2nd ed., R. Jacob Baker, Ed. Hoboken, NJ, USA: John Wiley & Sons, 2017.

[13] J. W. Pratt, “Concavity of the log likelihood,” *Journal of the American Statistical Association*, vol. 76, no. 373, pp. 103–106, 1981.

[14] R. Hunger, “An introduction to complex differentials and complex differentiability,” 2007. [Online]. Available: <https://mediatum.ub.tum.de/doc/631019/631019.pdf>

[15] J. R. Hershey, J. L. Roux, and F. Weninger, “Deep unfolding: Model-based inspiration of novel deep architectures,” *arXiv:1412.6980*, 2009. [Online]. Available: <https://arxiv.org/abs/1409.2574>

[16] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Int. Conf. Learning Representation (ICLR)*, San Diego, CA, USA, 2015.