Fast Deep Unfolded Hybrid Beamforming in Multiuser Large MIMO Systems

Nhan Thanh Nguyen*, Ly Van Nguyen†, Nir Shlezinger‡, A. Lee Swindlehurst†, and Markku Juntti*
*Centre for Wireless Communications, University of Oulu, P.O.Box 4500, FI-90014, Finland
†Department of EECS, University of California, Irvine, CA, USA
‡School of ECE, Ben-Gurion University of the Negev, Beer-Sheva, Israel
Emails: {nhan.nguyen, markku.juntti}@oulu.fi; {vanln1, swindle}@uci.edu, nirshl@bgu.ac.il

2023 57th Asilomar Conference on Signals, Systems, and Computers | 979-8-3503-2574-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/IEEECONF59524.2023.10476967

***Abstract*—Hybrid beamforming (HBF) is a key enabler for massive multiple-input multiple-output (MIMO) systems thanks to its capability to maintain significant spatial multiplexing gains with low hardware cost and power consumption. However, HBF optimizations are often challenging due to the nonconvexity and highly coupled analog and digital beamformers. In this paper, we propose an efficient HBF method based on deep unfolding to maximize the sum rate of large multiuser MIMO systems. We first derive closed-form expressions for the gradients of the sum rate with respect to the analog and digital beamformers to develop a projected gradient ascent (PGA) framework. We then incorporate this framework with the deep unfolding technique in an unfolded PGA deep neural network, which efficiently outputs reliable hybrid beamformers with low complexity and fast execution thanks to the well-trained hyperparameters. Numerical results show that the proposed method converges much faster than the conventional PGA scheme and significantly outperforms the conventional PGA and the successive convex approximation counterparts.**

***Index Terms*—mmWave, hybrid beamforming, massive MIMO, deep learning, AI, deep unfolding.**

I. Introduction

High spatial beamforming gains offered by large arrays in massive multiple-input multiple-output (MIMO) systems can efficiently compensate for the severe propagation loss in millimeter-wave (mmWave) channels [1]. In mmWave massive MIMO systems, hybrid beamforming (HBF) transceivers can maintain significant beamforming gains with reduced hardware cost and power consumption with respect to the conventional fully digital beamformers [2]–[5]. However, HBF designs and optimization problems are often highly challenging due to constant modulus constraints for analog beamforming coefficients and strongly coupled variables [6], [7]. Conventional HBF methods, such as Riemannian manifold minimization (RMM) [6] and alternating optimization (AO) [7] ensure good performance. However, they have slow convergence and high computational complexity [8], [9], which can be prohibitive for practical applications.

A potential solution to avoid cumbersome numerical algorithms in wireless communications systems is to leverage the learning capabilities of deep learning (DL) models [10], [11]. For example, deep neural networks (DNNs) [12]–[14] and/or convolutional neural networks (CNNs) [15]–[18] can be built and trained to generate HBF beamformers. Such methods usually lead to black-box models, which can offer satisfactory performance but lack explainability and have limitations on resource constraints, long training time, and complicated tuning of hyper-parameters [19], [20]. In contrast, model-based machine learning methods are developed based on both expert knowledge and learning capability of DL, offering more flexibility to configure and optimize the learning model [9], [11], [21]–[23]. A typical model-based learning technique is deep unfolding [9], [20], [24]. It constructs a DNN unrolling a well-developed iterative optimizer. Based on this advantage, efficient deep unfolding models have been proposed [25]–[28] for HBF designs with reduced feedback and complexity and improved convergence speed. However, these schemes require the implementation of highly-parameterized DNNs [25] and/or multiple CNNs [27]. A projected gradient ascent (PGA)–based fast and robust deep unfolding HBF design was proposed in [26] for broadcast systems. In [8] and [9], hybrid beamformers for point-to-point mmWave and THz massive MIMO systems were developed via unfolding the Ao and lease square methods. However, these unfolding methods are not applicable to multiuser scenarios.

In this work, we consider the downlink of a multiuser massive MIMO system, where the base station (BS) is equipped with a hybrid analog-digital beamformer. For the challenging HBF design, we propose a deep unfolded PGA method that unrolls the PGA optimizer [26]. Unlike [26], we herein investigate a practical downlink multiuser scenario, wherein the hybrid precoders need to combat the inter-user interference to maximize the system sum rate. We first derive the closed-form gradients of the system sum rate with respect to the analog and digital precoders. These allow us to develop the general iterative PGA framework for calculating the precoders in an alternating manner. This scheme can offer good performance, but its convergence is slow, which causes high complexity and latency. Therefore, we develop a deep unfolded PGA model to efficiently output the precoders within a fixed and limited number of iterations. Our simulation results show that the proposed unfolded PGA scheme performs much faster than its conventional counterparts while achieving substantial performance improvements compared to the combined successive convex approximation (SCA) and RMM approaches. In particular, the unfolded PGA scheme requires only a few iterations to converge to the same objective value attained by the conventional PGA with 30 iterations.

II. Signal Model and Problem Formulation

A. Signal Model

We consider a downlink multiuser MIMO system, where a base station (BS) equipped with N antennas serves K single-antenna users (UEs). The BS employs a fully connected HBF architecture with analog precoder $\mathbf{F} \in \mathbb{C}^{N\times M}$ and digital precoder $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K] \in \mathbb{C}^{M\times K}$, where M is the number of RF chains. The transmit power constraint is given as $\|\mathbf{FW}\|_{\mathcal{F}}^2 = P_{\mathrm{t}}$, where $\|\cdot\|_{\mathcal{F}}^2$ is the Frobenius norm of a matrix. We denote by $\mathbf{s} = [s_1, s_2, \ldots, s_K] \in \mathbb{C}^{K\times 1}$ the transmitted vector from the BS, with s_k and $\mathbf{w}_k$ being the symbol and precoding vectors intended for UE k. The received signal at UE k is given by

$$y_k = \mathbf{h}_k^{\mathsf{H}}\mathbf{F}\mathbf{w}_k s_k + \mathbf{h}_k^{\mathsf{H}} \sum\nolimits_{\ell\neq k}^{K} \mathbf{F}\mathbf{w}_\ell s_\ell + n_k, \tag{1}$$

where $n_k \sim \mathcal{CN}(0, \sigma_{\mathrm{n}}^2)$ is additive white Gaussian noise, and $\mathbf{h}_k \in \mathbb{C}^{N\times 1}$ is the channel vector from the BS to UE k. The channel $\mathbf{h}_k$ is modeled following the extended Saleh-Valenzuela model [6], [7]. Thus, we can express $\mathbf{h}_k$ as [6]

$$\mathbf{h}_k = \sum_{p=1}^{P} \alpha_{pk}\mathbf{a}(\phi_{pk}), \tag{2}$$

where P is the number of propagation paths, α_{pk} and ϕ_{pk} are the complex gain and angle of departure of the p-th path to the k-th user, respectively. Furthermore, $\mathbf{a} \in \mathbb{C}^{N\times 1}$ denotes the transmit array response vectors. Assuming that the BS deploys a uniform linear array (ULA) with half-wavelength antenna spacing, we have [6]

$$\mathbf{a}(\phi_{pk}) = \frac{1}{\sqrt{N}}\left[1, e^{j\pi\sin(\phi_{pk})}, \ldots, e^{j\pi(N-1)\sin(\phi_{pk})}\right]^{\mathsf{T}}. \tag{3}$$

B. Problem Formulation

Based on (1), the achievable sum rate of the system is given as

$$R = \sum_{k=1}^{K} \log\left(1 + \frac{|\mathbf{h}_k^{\mathsf{H}}\mathbf{F}\mathbf{w}_k|^2}{\sum_{\ell\neq k}^{K}|\mathbf{h}_k^{\mathsf{H}}\mathbf{F}\mathbf{w}_\ell|^2 + \sigma_{\mathrm{n}}^2}\right). \tag{4}$$

We aim to find the HBF design that maximizes R, formulated as:

$$\underset{\mathbf{F},\mathbf{W}}{\text{maximize}} \quad R \tag{5a}$$

$$\text{subject to} \quad |[\mathbf{F}]_{nm}| = 1, \forall n, m, \tag{5b}$$

$$\|\mathbf{FW}\|_{\mathcal{F}}^2 = P_{\mathrm{t}}, \tag{5c}$$

where (5b) enforces the unity modulus of the analog precoding coefficients, and (5c) constrains the transmit power to be equal to the power budget P_{t}. Problem (5) is nonconvex due to the constant modulus constraint, while $\mathbf{F}$ and $\mathbf{W}$ are highly coupled in the objective function and constraint (5c). Therefore, this problem is challenging to solve. Next, we develop the PGA procedure and the deep unfolded PGA model solving (5).

III. Proposed Unfolded PGA Design

A. PGA Procedure

We leverage the PGA method in combination with AO. Specifically, in each iteration of the PGA procedure, one precoder is solved with the other fixed. For a fixed $\mathbf{W}$, $\mathbf{F}$ can be updated at the $(i+1)$-th iteration as

$$\mathbf{F}_{(i+1)} = \mathbf{F}_{(i)} + \mu_{(i)}\nabla_{\mathbf{F}} R\Big|_{\mathbf{F}=\mathbf{F}_{(i)}}, \tag{6}$$

$$[\mathbf{F}_{(i+1)}]_{nm} = \frac{[\mathbf{F}_{(i+1)}]_{nm}}{\left|[\mathbf{F}_{(i+1)}]_{nm}\right|}, \ \forall n, m, \tag{7}$$

where $\nabla_{\mathbf{Z}} R$ is the gradient of the sum rate R with respect to $\mathbf{Z}$. Similarly, given $\mathbf{F}$, $\mathbf{W}$ can be updated at iteration $i+1$ as:

$$\mathbf{W}_{(i+1)} = \mathbf{W}_{(i)} + \lambda_{(i)}\nabla_{\mathbf{W}} R\Big|_{\mathbf{W}=\mathbf{W}_{(i)}}, \tag{8}$$

$$\mathbf{W}_{(i+1)} = P_{\mathrm{t}}\frac{\mathbf{W}_{(i+1)}}{\left\|\mathbf{F}_{(i+1)}\mathbf{W}_{(i+1)}\right\|_{\mathcal{F}}}. \tag{9}$$

In (6) and (8), $\{\mu_{(i)}, \lambda_{(i)}\}$ are the step sizes. Furthermore, the solutions to $\mathbf{F}$ and $\mathbf{W}$ are projected into the feasible space in (7) and (9). The closed-form gradients $\nabla_{\mathbf{F}} R$ and $\nabla_{\mathbf{W}} R$ are derived in the following theorem.

***Theorem** 1:* The gradients of the achievable rate with respect to the analog and digital precoders, i.e., $\nabla_{\mathbf{F}} R$ and $\nabla_{\mathbf{W}} R$, are given by

$$\nabla_{\mathbf{F}} R = \sum_{k=1}^{K}\left(\frac{\tilde{\mathbf{H}}_k\mathbf{F}\mathbf{V}}{\mathrm{tr}(\mathbf{F}\mathbf{V}\mathbf{F}^{\mathsf{H}}\tilde{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2} - \frac{\tilde{\mathbf{H}}_k\mathbf{F}\mathbf{V}_{\bar{k}}}{\mathrm{tr}(\mathbf{F}\mathbf{V}_{\bar{k}}\mathbf{F}^{\mathsf{H}}\tilde{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2}\right), \tag{10}$$

$$\nabla_{\mathbf{W}} R = \sum_{k=1}^{K}\left(\frac{\bar{\mathbf{H}}_k\mathbf{W}}{\mathrm{tr}(\mathbf{W}\mathbf{W}^{\mathsf{H}}\bar{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2} - \frac{\bar{\mathbf{H}}_k\mathbf{W}_{\bar{k}}}{\mathrm{tr}(\mathbf{W}_{\bar{k}}\mathbf{W}_{\bar{k}}^{\mathsf{H}}\bar{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2}\right), \tag{11}$$

where

$$\tilde{\mathbf{H}}_k \triangleq \mathbf{h}_k\mathbf{h}_k^{\mathsf{H}}, \ \bar{\mathbf{H}}_k \triangleq \mathbf{F}^{\mathsf{H}}\tilde{\mathbf{H}}_k\mathbf{F},$$
$$\mathbf{V} \triangleq \mathbf{W}\mathbf{W}^{\mathsf{H}}, \ \mathbf{V}_{\bar{k}} \triangleq \mathbf{W}_{\bar{k}}\mathbf{W}_{\bar{k}}^{H},$$

and $\mathbf{W}_{\bar{k}} \in \mathbb{C}^{M\times K}$ is obtained by replacing the k-th column of $\mathbf{W}$ with all zeros.

Proof: See Appendix A. □

B. Unfolded PGA Scheme

1) Network Structure and Training: The performance and convergence of the PGA procedure in (6)–(9) are critically affected by the step sizes $\boldsymbol{\mu} \triangleq \{\mu_{(i)}\}_{i=0}^{I-1}, \boldsymbol{\lambda} \triangleq \{\lambda_{(i)}\}_{i=0}^{I-1}$. Manually determining $\{\boldsymbol{\mu}, \boldsymbol{\lambda}\}$ may not ensure an expected convergence while employing a line search and backtracking to optimize them would require excessively high additional computational and time complexity, especially in massive MIMO systems. Instead, we propose leveraging the learning capability of DNNs to train and tune the hyperparameters $\{\boldsymbol{\mu}, \boldsymbol{\lambda}\}$ to enable good convergence of the PGA scheme. To this end, we incorporate I iterations of the PGA procedure into I layers of a deep unfolded DNN model, which is illustrated

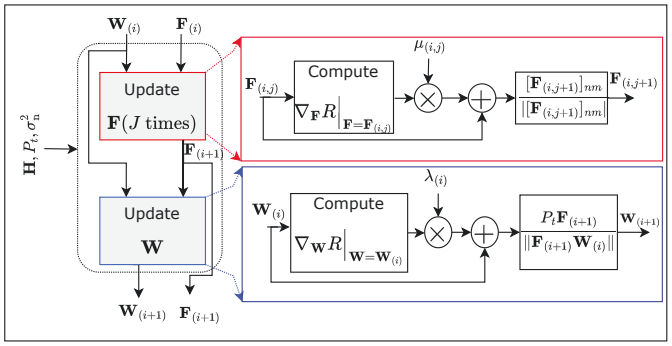


Fig. 1. Illustration of the $(i+1)$-th layer of the proposed unfolded PGA model.

Algorithm 1 Proposed HBF Design

Input: $\mathbf{H}$, P_{t}, ω, and the trained step sizes $\{\boldsymbol{\mu}, \boldsymbol{\lambda}\}$.
Output: $\mathbf{F}$ and $\mathbf{W}$

1: **Initialization:** Initialize $\{\mathbf{F}_{(0)}, \mathbf{W}_{(0)}\}$ based on (12) and (13).
2: **for** $i = 0 \to I-1$ **do**
3: Set $\mathbf{F}_{(i,0)} = \mathbf{F}_{(i)}$.
4: **for** $j = 0 \to J-1$ **do**
5: Compute gradient $\nabla_{\mathbf{F}} R$ at $(\mathbf{F}, \mathbf{W}) = (\mathbf{F}_{(i,j)}, \mathbf{W}_{(i)})$ based on (10).
6: Update $\mathbf{F}_{(i,j+1)}$ based on (6) and (7).
7: **end for**
8: Set $\mathbf{F}_{(i+1)} = \mathbf{F}_{(i,J)}$ and apply the projection in (7).
9: Compute gradient $\nabla_{\mathbf{W}} R$ at $(\mathbf{F}, \mathbf{W}) = \big(\mathbf{F}_{(i+1)}, \mathbf{W}_{(i)}\big)$ based on (11).
10: Obtain $\mathbf{W}_{(i+1)}$ based on (8) and (9).
11: **end for**
12: **return** $\mathbf{F}_{(I)}$ and $\mathbf{W}_{(I)}$ as the solution to $\mathbf{F}$ and $\mathbf{W}$.

in Fig. 1. This approach follows the updating process in (6)–(9). More specifically, the $(i+1)$-th layer takes $\{\mathbf{F}_{(i)}, \mathbf{W}_{(i)}\}$, $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_K]^{\mathsf{H}}$, P_{t}, and σ_{n}^2 as input data, and outputs $\{\mathbf{F}_{(i+1)}, \mathbf{W}_{(i+1)}\}$. To train the DNN, the loss function is set to

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\lambda}) = \sum_{k=1}^{K} \log_2 \left(1 + \frac{\left| \mathbf{h}_k^{\mathsf{H}} \mathbf{F}_{(I)} \mathbf{w}_{k(I)} \right|^2}{\sum_{\ell \neq k}^{K} \left| \mathbf{h}_k^{\mathsf{H}} \mathbf{F}_{(I)} \mathbf{w}_{j(I)} \right|^2 + \sigma_{\mathrm{n}}^2} \right),$$

based on (4). The loss $\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\lambda})$ is a function of the trainable step sizes $\{\boldsymbol{\mu}, \boldsymbol{\lambda}\}$ because $\{\mathbf{F}_{(I)}, \mathbf{W}_{(I)}\}$ depends on $\{\mathbf{F}_{(i)}\}_{i=0}^{I-1}$, $\{\mathbf{W}_{(i)}\}_{i=0}^{I-1}$, and $\{\boldsymbol{\mu}, \boldsymbol{\lambda}\}$. The unfolded PGA model is trained to optimize $\{\boldsymbol{\mu}, \boldsymbol{\lambda}\}$ to maximize R within a predetermined number of layers, i.e., I. This is unsupervised training and does not require any labels.

2) Overall HBF Design: We outline the proposed HBF design based on the unfolded PGA scheme in Algorithm 1. First, the analog precoder is initialized as [29]

$$[\mathbf{F}_{(0)}]_{nm} = e^{-j\vartheta_{nm}}, \tag{12}$$

where ϑ_{nm} is the phase of the (n,m)-th entries of $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_K]$. This offers large array gains as the phases of the analog precoder are aligned well with the channel. With $\mathbf{F}_{(0)}$, the initial solution to the digital precoder is set to

$$\mathbf{W}_{(0)} = \mathbf{F}_{(0)}^{\dagger} \mathbf{H}^{\dagger}, \ \mathbf{W}_{(0)} = \sqrt{P_{\mathrm{t}}} \frac{\mathbf{W}_{(0)}}{\left\| \mathbf{F}_{(0)} \mathbf{W}_{(0)} \right\|_{\mathcal{F}}}. \tag{13}$$

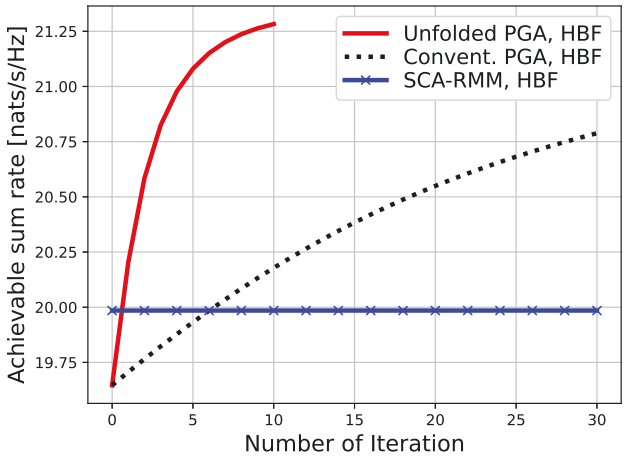
With (13), $\mathbf{W}_{(0)}$ has the least-squares distance to the zero-forcing digital precoder [24]. The unfolded model uses the trained step sizes $\{\boldsymbol{\mu}, \boldsymbol{\lambda}\}$ to perform the updates in (6)–(8) and the projections (7) and (9), as outlined in steps 2–11 of Algorithm 1. Specifically, steps 3–8 compute the output $\mathbf{F}_{(i+1)}$ over the J layers. Then, $\mathbf{W}_{(i+1)}$ is obtained in step 10 based on the updated $\mathbf{F}_{(i+1)}$. The outcome of the algorithm is the final output of the unfolded PGA model. Here, compared to $\mathbf{W}$, $\mathbf{F}$ is updated over more iterations to ensure that the updating speed of these two are comparable. Indeed, our numerical results show that the gradient of R with respect to $\mathbf{W}$ is much larger than that with respect to $\mathbf{F}$ in magnitudes. We refer readers to [24] for more details on this.

3) Complexity Analysis: We herein present the complexity analysis of Algorithm 1. First, note that $\mathbf{V}$ and $\mathbf{V}_{\bar{k}}$ are unchanged over J inner iterations of updating $\mathbf{F}$, while $\mathbf{W}$ is of size $(M \times K)$ with $M, K \ll N$. Therefore, the computation of $\nabla_{\mathbf{F}} R$ and $\nabla_{\mathbf{W}} R$ requires most of the complexity of Algorithm 1. In (10), computing $\tilde{\mathbf{H}}_k \mathbf{F}$ requires a complexity of only $\mathcal{O}(NM)$ because $\tilde{\mathbf{H}}_k \mathbf{F} = \mathbf{h}_k \mathbf{h}_k^{\mathsf{H}} \mathbf{F}$, i.e., the term $\mathbf{h}_k^{\mathsf{H}} \mathbf{F}$ can be computed first before the right-multiplication with $\mathbf{h}_k$. Therefore, the complexity of computing $\tilde{\mathbf{H}}_k \mathbf{F} \mathbf{V}$ is $\mathcal{O}(NM^2)$. Furthermore, the complexity to compute $\mathrm{trace}\{\mathbf{F}\mathbf{V}\mathbf{F}^{\mathsf{H}} \tilde{\mathbf{H}}_k\}$ is only $\mathcal{O}(NM)$. This is because $\mathbf{V}\mathbf{F}^{\mathsf{H}} \tilde{\mathbf{H}}_k = (\tilde{\mathbf{H}}_k \mathbf{F} \mathbf{V})^{\mathsf{H}}$, and $\tilde{\mathbf{H}}_k \mathbf{F} \mathbf{V}$ has already been computed. Thus, the complexity for computing the first term in (10) is $\mathcal{O}(NM^2K)$. The total complexity in to obtain (10) is still $\mathcal{O}(NM^2K)$ because the two terms in (10) have the same complexity. Thus, the overall complexity to obtain $\mathbf{F}$ is $\mathcal{O}(IJ \max(NM^2K, N^2K))$. Similarly, we can obtain the complexity of determining $\mathbf{W}$ as $\mathcal{O}(I \max(M^2K^2, N^2K))$. Because $N \geq K$ and $NM^2K \geq M^2K^2$, the complexity required for $\mathbf{F}$ dominates that for $\mathbf{W}$. Thus, the overall computational load of Algorithm 1 is $\mathcal{O}(IJ \max(NM^2K, N^2K))$.

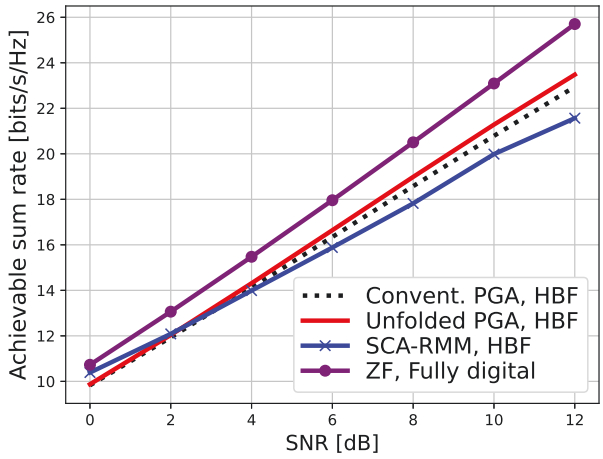
IV. Numerical Results

In the simulations, we assume $K = M = 4$ and $N = 32$. The channels for training and testing are generated based on (2) with $P = 10$, $\alpha_{pk} \sim \mathcal{CN}(0,1)$, and $\theta_{pk} \sim \mathcal{U}(0, 2\pi)$ [7]. We implement and train the deep unfolded PGA model with Python and the Pytorch library. For the model training, the decaying learning rate and initial learning rate of 0.97 and 0.001, respectively, and the Adam optimizer are used. The model is trained with a dataset of 500 channels over 20 epochs. The step sizes are initialized as $\mu_{(0,0)} = 0.01$ and $\lambda_{(0,0)} = 0.0001$, which are also used as the fixed step sizes for the conventional PGA algorithm without unfolding. These are set based on empirical observations. Specifically, we found that $\lambda_{(0,0)} \ll \mu_{(0,0)}$ should be used to ensure the convergence of the conventional PGA method.

In Figs. 2(a) and 2(b), we demonstrate the convergence and sum rate performance of the proposed unfolded PGA scheme. For comparison, we consider HBF designs using the conventional PGA procedure with $I = 30$ iterations and the design combining successive convex approximation (SCA) and Riemannian manifold minimization (RMM), which is termed



(a) R vs. I, SNR $= 10$ dB.



(b) R vs. SNRs.

Fig. 2. Convergence and sum rate performance of the proposed deep unfolding HBF design with $N = 32$ and $K = M = 4$.

as "SCA-RMM, HBF" in the figures. Specifically, in this combined SCA-RMM scheme, the effective precoder is first found via SCA [30] and then factored into analog and digital components with the RMM method [6]. We also include the results of the fully digital zero-forcing (ZF) beamformer.

It is seen in Fig. 2(a) that the sum rate of the PGA algorithms (with and without unfolding) can converge to a higher value than the SCA-RMM approach. The significant performance loss of the SCA-RMM scheme is due to the sub-optimality while solving the effective precoder with SCA and the factorization in two steps with RMM. In particular, the sum rate objective of the unfolded PGA model increases quickly and reaches a high value with only $I = 10$ layers. The conventional PGA converges slowly and is still far from convergence even after $I = 30$ iterations. It is clear that with the learned step sizes, the unfolded PGA scheme requires only three layers to achieve the same objective value of the conventional PGA with 30 iterations. In Fig. 2(b), it is seen that the unfolded PGA scheme outperforms the compared traditional HBF designs, especially at moderate and high SNRs, and performs close to the fully-digital ZF scheme.

V. Conclusion

We have proposed an unfolded PGA model for HBF design in large multiuser MIMO systems. In the proposed scheme, the learning capability of DNNs is leveraged to efficiently tune the step sizes of the PGA procedure to improve and accelerate its convergence. As a result, the proposed unfolded framework follows the optimization principle of the conventional PGA method with better performance while requiring much lower complexity. Our simulation results validated the fast convergence as well as superior performance of the proposed algorithm compared with other traditional optimization approaches. Our future work will consider more practical sub-connected HBF architectures in wideband multicarrier scenarios.

Aknowledgement

This work was supported in part by the Research Council of Finland through 6G Flagship under Grant 346208 and through project DIRECTION under Grant 354901, Nokia Foundation's Jorma Ollila Grant, EERA Project under Grant 332362, Infotech Oulu via EEWA Project, Business Finland, Keysight, MediaTek, Siemens, Ekahau, and Verkotan via project 6GLearn, and U.S. National Science Foundation grant CCF-2225575.

Appendix A
Proof of Theorem 1

We first rewrite (4) as

$$\begin{aligned} R &= \sum_{k=1}^{K} \log_2 \left(\frac{\sum_{k=1}^{K} |\mathbf{h}_k^{\mathsf{H}} \mathbf{F} \mathbf{w}_k|^2 + \sigma_{\mathrm{n}}^2}{\sum_{\ell \in \mathcal{K} \setminus k} |\mathbf{h}_k^{\mathsf{H}} \mathbf{F} \mathbf{w}_\ell|^2 + \sigma_{\mathrm{n}}^2} \right) \\ &= \sum_{k=1}^{K} \log_2 \left(\frac{\mathrm{tr}(\mathbf{F}\mathbf{W}\mathbf{W}^{\mathsf{H}}\mathbf{F}^{\mathsf{H}}\mathbf{h}_k\mathbf{h}_k^{\mathsf{H}}) + \sigma_{\mathrm{n}}^2}{\mathrm{tr}(\mathbf{F}\mathbf{W}_{\bar{k}}\mathbf{W}_{\bar{k}}^{\mathsf{H}}\mathbf{F}^{\mathsf{H}}\mathbf{h}_k\mathbf{h}_k^{\mathsf{H}}) + \sigma_{\mathrm{n}}^2} \right) \qquad (14) \\ &= \sum_{k=1}^{K} \log_2 \left(\mathrm{tr}(\mathbf{F}\mathbf{V}\mathbf{F}^{\mathsf{H}}\tilde{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2 \right) \\ &\quad - \sum_{k=1}^{K} \log_2 \left(\mathrm{tr}(\mathbf{F}\mathbf{V}_{\bar{k}}\mathbf{F}^{\mathsf{H}}\tilde{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2 \right), \qquad (15) \end{aligned}$$

where $\mathbf{V} \triangleq \mathbf{W}\mathbf{W}^{\mathsf{H}}$, $\mathbf{V}_{\bar{k}} \triangleq \mathbf{W}_{\bar{k}}\mathbf{W}_{\bar{k}}^{H}$, $\tilde{\mathbf{H}}_k \triangleq \mathbf{h}_k\mathbf{h}_k^{\mathsf{H}}$, and $\mathbf{W}_{\bar{k}} \in \mathbb{C}^{M \times K}$ is obtained by replacing the k-th column of $\mathbf{W}$ with all zeros. As a result, $\nabla_{\mathbf{F}} R$ can be computed as

$$\begin{aligned} \nabla_{\mathbf{F}} R = &\sum_{k=1}^{K} \underbrace{\frac{\partial}{\partial \mathbf{F}^*} \log_2 \left(\mathrm{tr}(\mathbf{F}\mathbf{V}\mathbf{F}^{\mathsf{H}}\tilde{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2 \right)}_{\triangleq \partial_{k1}} \\ &- \sum_{k=1}^{K} \underbrace{\frac{\partial}{\partial \mathbf{F}^*} \log_2 \left(\mathrm{tr}(\mathbf{F}^{\mathsf{H}}\mathbf{V}_{\bar{k}}\mathbf{F}^{\mathsf{H}}\tilde{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2 \right)}_{\triangleq \partial_{k2}}. \qquad (16) \end{aligned}$$

By leveraging the result $\partial \mathrm{tr}(\mathbf{Z}\mathbf{A}_0\mathbf{Z}^{\mathsf{H}}\mathbf{A}_1)/\partial \mathbf{Z}^* = \mathbf{A}_1\mathbf{Z}\mathbf{A}_0$ [31], we obtain ∂_{k1} and ∂_{k2} as

$$\partial_{k1} = \frac{\frac{\partial}{\partial \mathbf{F}^*}\left(\mathrm{tr}(\mathbf{F}\mathbf{V}\mathbf{F}^{\mathsf{H}}\tilde{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2\right)}{\ln 2(\mathrm{tr}(\mathbf{F}\mathbf{V}\mathbf{F}^{\mathsf{H}}\tilde{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2)} = \frac{\tilde{\mathbf{H}}_k\mathbf{F}\mathbf{V}}{\ln 2(\mathrm{tr}(\mathbf{F}\mathbf{V}\mathbf{F}^{\mathsf{H}}\tilde{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2)}, \tag{17}$$

$$\partial_{k2} = \frac{\tilde{\mathbf{H}}_k\mathbf{F}\mathbf{V}_{\bar{k}}}{\ln 2(\mathrm{tr}(\mathbf{F}\mathbf{V}_{\bar{k}}\mathbf{F}^{\mathsf{H}}\tilde{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2)}. \tag{18}$$

From (16), (17), and (18), we obtain 1. To compute $\nabla_{\mathbf{W}} R$, we follow a similar method for deriving $\nabla_{\mathbf{F}} R$ and rewrite R in (14) as

$$\begin{aligned} R &= \sum_{k=1}^{K} \log_2\left(\frac{\mathrm{tr}(\mathbf{W}\mathbf{W}^{\mathsf{H}}\mathbf{F}^{\mathsf{H}}\mathbf{h}_k\mathbf{h}_k^{\mathsf{H}}\mathbf{F}) + \sigma_{\mathrm{n}}^2}{\mathrm{tr}(\mathbf{W}_{\bar{k}}\mathbf{W}_{\bar{k}}^{\mathsf{H}}\mathbf{F}^{\mathsf{H}}\mathbf{h}_k\mathbf{h}_k^{\mathsf{H}}\mathbf{F}) + \sigma_{\mathrm{n}}^2}\right) \\ &= \sum_{k=1}^{K} \log_2\left(\mathrm{tr}(\mathbf{W}\mathbf{W}^{\mathsf{H}}\bar{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2\right) \\ &\quad - \sum_{k=1}^{K} \log_2\left(\mathrm{tr}(\mathbf{W}_{\bar{k}}\mathbf{W}_{\bar{k}}^{\mathsf{H}}\bar{\mathbf{H}}_k) + \sigma_{\mathrm{n}}^2\right), \end{aligned} \tag{19}$$

with $\bar{\mathbf{H}}_k \triangleq \mathbf{F}^{\mathsf{H}}\tilde{\mathbf{H}}_k\mathbf{F}$. Following similar derivations as in (16)–(18), we obtain (11), and the proof is completed.

REFERENCES

[1] N. T. Nguyen and K. Lee, "Coverage and cell-edge sum-rate analysis of mmwave massive MIMO systems with ORP schemes and MMSE receivers," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5349–5363, 2018.

[2] X. Gao, L. Dai, and A. M. Sayeed, "Low RF-complexity technologies to enable millimeter-wave MIMO with large antenna array for 5G wireless communications," *IEEE COMMAG*, vol. 56, no. 4, pp. 211–217, 2018.

[3] N. T. Nguyen and K. Lee, "Unequally sub-connected architecture for hybrid beamforming in massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1127–1140, 2019.

[4] G. M. Gadiel, N. T. Nguyen, and K. Lee, "Dynamic unequally sub-connected hybrid beamforming architecture for massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3469–3478, 2021.

[5] N. T. Nguyen, N. Shlezinger, Y. C. Eldar, and M. Juntti, "Multiuser MIMO wideband joint communications and sensing system with subcarrier allocation," *IEEE Trans. Signal Process.*, vol. 71, pp. 2997–3013, 2023.

[6] X. Yu, J.-C. Shen, J. Zhang, and K. B. Letaief, "Alternating minimization algorithms for hybrid precoding in millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 485–500, 2016.

[7] F. Sohrabi and W. Yu, "Hybrid digital and analog beamforming design for large-scale antenna arrays," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, 2016.

[8] N. T. Nguyen, M. Ma, N. Shlezinger, Y. C. Eldar, A. Swindlehurst, and M. Juntti, "Deep unfolding-enabled hybrid beamforming design for mmwave massive MIMO systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2023.

[9] N. T. Nguyen, M. Ma, O. Lavi, N. Shlezinger, Y. C. Eldar, A. L. Swindlehurst, and M. Juntti, "Deep unfolding hybrid beamforming designs for THz massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 71, pp. 3788–3804, 2023.

[10] Q.-V. Pham *et al.*, "Intelligent radio signal processing: A survey," *IEEE Access*, vol. 9, pp. 83 818–83 850, 2021.

[11] A. Jagannath, J. Jagannath, and T. Melodia, "Redefining wireless communication for 6G: Signal processing meets deep learning with deep unfolding," *IEEE Trans. Artificial Intelligence*, vol. 2, no. 6, pp. 528–536, 2021.

[12] X. Li and A. Alkhateeb, "Deep learning for direct hybrid precoding in millimeter wave massive MIMO systems," in *Proc. Annual Asilomar Conf. Signals, Syst., Comp.*, 2019, pp. 800–805.

[13] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, 2019.

[14] Q. Hu *et al.*, "Two-timescale end-to-end learning for channel acquisition and hybrid precoding," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 163–181, 2021.

[15] A. M. Elbir and K. V. Mishra, "Joint antenna selection and hybrid beamformer design using unquantized and quantized deep learning networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1677–1688, 2019.

[16] A. M. Elbir and A. K. Papazafeiropoulos, "Hybrid precoding for multiuser millimeter wave massive MIMO systems: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 552–563, 2019.

[17] T. Peken *et al.*, "Deep learning for SVD and hybrid beamforming," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6621–6642, 2020.

[18] H. Hojatian, J. Nadal, J.-F. Frigon, and F. Leduc-Primeau, "Unsupervised deep learning for massive MIMO hybrid beamforming," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7086–7099, 2021.

[19] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *arXiv*, 2022.

[20] N. Shlezinger, M. Ma, O. Lavi, N. T. Nguyen, Y. C. Eldar, and M. Juntti, "AI-empowered hybrid MIMO beamforming," *arXiv preprint arXiv:2303.01723*, 2023.

[21] L. V. Nguyen, N. T. Nguyen, N. H. Tran, M. Juntti, A. L. Swindlehurst, and D. H. Nguyen, "Leveraging deep neural networks for massive MIMO data detection," *IEEE Wireless Commun.*, vol. 30, no. 1, pp. 174–180, 2022.

[22] N. T. Nguyen, L. V. Nguyen, T. Huynh-The, D. H. Nguyen, A. L. Swindlehurst, and M. Juntti, "Machine learning-based reconfigurable intelligent surface-aided MIMO systems," in *Proc. IEEE Works. on Sign. Proc. Adv. in Wirel. Comms.*, 2021, pp. 101–105.

[23] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?" *IEEE Trans. Wireless Commun.*, vol. 67, no. 10, pp. 7331–7376, 2019.

[24] N. T. Nguyen, L. V. Nguyen, N. Shlezinger, Y. C. Eldar, A. L. Swindlehurst, and M. Juntti, "Joint communications and sensing hybrid beamforming design via deep unfolding," *arXiv preprint arXiv:2307.04376*, 2023.

[25] E. Balevi and J. G. Andrews, "Unfolded hybrid beamforming with GAN compressed ultra-low feedback overhead," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 8381–8392, 2021.

[26] O. Agiv and N. Shlezinger, "Learn to rapidly optimize hybrid precoding," in *Proc. IEEE SPAWC*, 2022, pp. 1–5.

[27] S. Shi, Y. Cai, Q. Hu, B. Champagne, and L. Hanzo, "Deep-unfolding neural-network aided hybrid beamforming based on symbol-error probability minimization," *IEEE Trans. Veh. Technol.*, 2022.

[28] K.-Y. Chen *et al.*, "Hybrid beamforming in mmwave MIMO-OFDM systems via deep unfolding," in *Proc. IEEE Veh. Technol. Conf.*, 2022, pp. 1–7.

[29] L. Liang, W. Xu, and X. Dong, "Low-complexity hybrid precoding in massive multiuser MIMO systems," *IEEE Commun. Lett.*, vol. 3, no. 6, pp. 653–656, 2014.

[30] L.-N. Tran, M. F. Hanif, A. Tolli, and M. Juntti, "Fast converging algorithm for weighted sum rate maximization in multicell MISO downlink," *IEEE Signal Process. Lett.*, vol. 19, no. 12, pp. 872–875, 2012.

[31] A. Hjorungnes and D. Gesbert, "Complex-valued matrix differentiation: Techniques and key results," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2740–2746, 2007.