Linear and Deep Neural Network-based Receivers for Massive MIMO Systems with One-Bit ADCs

Ly V. Nguyen, *Student Member, IEEE*, A. Lee Swindlehurst, *Fellow, IEEE*, and Duy H. N. Nguyen, *Senior Member, IEEE*

***Abstract*—The use of one-bit analog-to-digital converters (ADCs) is a practical solution for reducing cost and power consumption in massive Multiple-Input-Multiple-Output (MIMO) systems. However, the distortion caused by one-bit ADCs makes the data detection task much more challenging. In this paper, we propose a two-stage detection method for massive MIMO systems with one-bit ADCs. In the first stage, we present several linear receivers based on the Bussgang decomposition that show significant performance gains over conventional linear receivers. Next, we reformulate the maximum-likelihood (ML) detection problem to address its non-robustness. Based on the reformulated ML detection problem, we propose a model-driven deep neural network-based detector, namely OBMNet, whose performance is comparable with an existing support vector machine-based receiver, albeit with a much lower computational complexity. A nearest-neighbor search method is then proposed for the second stage to refine the first stage solution. Unlike existing search methods that typically perform the search over a large candidate set, the proposed search method generates a limited number of most likely candidates and thus limits the search complexity. Numerical results confirm the low complexity, efficiency, and robustness of the proposed two-stage detection method.**

***Index Terms*—Massive MIMO, one-bit ADCs, linear receivers, deep neural networks, machine learning, data detection.**

I. Introduction

Massive multiple-input multiple-output (MIMO) systems, possessing the capability of boosting the throughput and energy efficiency by several orders of magnitude over conventional MIMO systems [1], [2], are considered to be a disruptive solution for 5G-and-beyond networks [3], [4]. However, a massive MIMO system requires a large number of radio-frequency (RF) chains, which significantly increases the power consumption and hardware complexity. Among the components of an RF chain, high-resolution analog-to-digital converters (ADCs) are power-hungry devices whose power consumption increases exponentially with the number of bits per sample and linearly with the sampling rate [5]. A promising solution for reducing the power consumption and hardware complexity is to use low-resolution ADCs. The simplest architecture involving one-bit ADCs requires only one comparator and does not require an automatic gain control (AGC). Therefore, the use of one-bit ADCs can significantly reduce both the power consumption and hardware complexity. However, the severe nonlinearity of one-bit ADCs causes significant distortions in the received signals, since only the *sign* of the real and imaginary parts of the received signals is retained.

Due to the severe nonlinearity, data detection in one-bit massive MIMO systems becomes much more challenging. Numerous efforts have been made to address this problem, e.g., [6]–[14]. A one-bit maximum-likelihood (ML) detector was derived in [6]. For large-scale systems where ML detection is impractical, the authors of [6] proposed a so-called near-ML (nML) data detection method. The ML and nML methods are however non-robust at high signal-to-noise ratios (SNRs) when the channel state information (CSI) is not perfectly known. A one-bit sphere decoding (OSD) technique was proposed in [7]. However, the OSD technique requires a preprocessing stage whose computational complexity is exponentially proportional to both the number of receive and transmit antennas. The exponential computational complexity of OSD makes it difficult to implement in large-scale MIMO systems. Generalized approximate message passing (GAMP) and Bayes inference are exploited in [8], but the resulting method is sophisticated and expensive to implement. In [9], an iterative detection method based on the alternating direction method of multipliers (ADMM) algorithm is proposed that takes hardware impairments into account. The work in [10] exploits the forward-backward splitting (FBS) framework to design an iterative algorithm for one-bit massive MIMO-OFDM systems. Several other data detection approaches have also been proposed in [11]–[14], but they are only applicable in systems where either a cyclic redundancy check (CRC) [11]–[13] or an error correcting code such as a low-density parity-check (LDPC) code [14] is available. In this paper, we propose a two-stage detection method for massive MIMO systems with one-bit ADCs. The proposed method is efficient and robust with low complexity, and also applicable to large-scale systems without the need for CRC or error correcting codes.

In the first stage, we first focus on a class of linear receivers. Conventional receiver structures in this class has taken one of the following two strategies: (i) using standard linear receivers designed for systems with infinite-resolution ADCs, e.g., [6], [15], [16]; or (ii) using an approximate model for the one-bit ADC to construct other linear receiver designs, e.g., [17], [18]. As in our preliminary work discussed in [19], we exploit the Bussgang decomposition [20] in this

This work was supported in part by University Grants Program (UGP) from San Diego State University, and in part by the U.S. National Science Foundation under Grants CCF-1703635 and ECCS-1824565.

Ly V. Nguyen is with the Computational Science Research Center, San Diego State University, San Diego, CA, USA 92182 (e-mail: vnguyen6@sdsu.edu).

A. Lee Swindlehurst is with the Center for Pervasive Communications and Computing, Henry Samueli School of Engineering, University of California, Irvine, CA, USA 92697 (e-mail: swindle@uci.edu).

Duy H. N. Nguyen is with the Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA, USA 92182 (e-mail: duy.nguyen@sdsu.edu).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2021.3082844, IEEE Transactions on Wireless Communications

paper to examine various linear receiver architectures. Compared to other detection methods such as ML [6], OSD [7], GAMP [8] or iterative detection algorithms such as nML [6], or those based on ADMM [9], or FBS [10], the Bussgang-based linear receivers are easier to implement since they have simple structures and low complexity. Then, we study a deep learning-based detector for one-bit massive MIMO systems. There has been considerable recent interest in learning-based methods for MIMO data detection [21]–[28]. While the deep learning-based detectors in [21]–[24] are designed for MIMO systems with full-resolution ADCs, the learning-based detectors in [25]–[27] are dedicated to systems with low-resolution ADCs and are "blind" in the sense that channel state information (CSI) is not required. However, these blind detection methods are restricted to MIMO systems with a small number of transmit antennas and only low-dimensional constellations. More recently, in [28] a support vector machine (SVM) was exploited for one-bit MIMO data detection, and the SVM approach was shown to achieve better performance than the above linear and learning-based receivers. In this paper, we propose a new and efficient One-Bit massive MIMO data detection Network (OBMNet), which is based on the deep neural network (DNN) architecture.

The contributions of this first stage are as follows: First, we summarize existing linear receiver structures including conventional and Bussgang-based approaches. Then we observe a somewhat suprising result in the numerical examples indicating that conventional linear receivers with estimated CSI outperform those with perfect CSI in the presence of one-bit observations. An explanation for this observation, which is closely related to the structure of the Bussgang-based linear receivers, is also provided. Next, we reformulate the ML detection problem by approximating the cumulative distribution function of a Gaussian random variable with a Sigmoid function. We show that the reformulated problem addresses the non-robustness issue of conventional ML detection. We then propose a model-driven OBMNet for data detection in one-bit massive MIMO systems. Unlike the structure of conventional DNNs where each layer contains a fixed weight matrix and a fixed bias vector, each layer of the proposed OBMNet has two adaptive weight matrices and no bias vector. Numerical results show that OBMNet outperforms the linear receivers and its performance is also comparable with that of the SVM-based method in [28]. However, the proposed OBMNet has much lower computational complexity than the SVM-based method.

In the second stage, we propose a nearest-neighbor (NN) search method to refine the solution of stage 1. The idea of using two-stage detection methods has been studied previously in [6], [28]. However, the search metric used by the second stage of [6] is susceptible to CSI errors. This issue was addressed in [28] thanks to a more robust search metric. Although the second stage in [28] is robust, its complexity can be very high since the dimension of the search space over the entire candidate set can be very large. The contribution of the proposed NN search method is that it generates searches over a limited number of candidates that are nearest to the solution of stage 1 and thus helps contain the search complexity. The main challenge is to obtain the set of nearest candidates efficiently and quickly. To overcome this challenge, we propose a recursive strategy that can obtain this candidate set quickly so that the proposed NN search method can be implemented in an efficient manner.

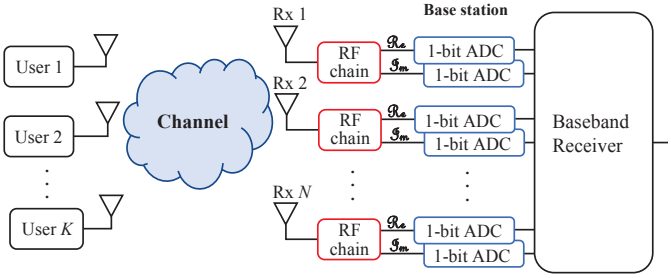


Fig. 1: Block diagram of a massive MIMO system with K single-antenna users and an N-antenna base station equipped with $2N$ one-bit ADCs.

The rest of this paper is organized as follows: Section II introduces the assumed system model and presents the conventional as well as the Bussgang-based linear receivers. The reformulated robust ML detection problem and OBMNet are proposed in Section III. Section IV presents the proposed NN search method. A computational complexity analysis and numerical results are given in Section V and Section VI concludes the paper.

Notation: Upper-case and lower-case boldface letters denote matrices and column vectors, respectively. $\mathbb{E}[\cdot]$ represents expectation. The operator $|\cdot|$ denotes the absolute value of a number. $\|\cdot\|$ denotes the ℓ_2-norm of a vector. The transpose and conjugate transpose are denoted by $[\cdot]^T$ and $[\cdot]^H$, respectively. The notation $\Re\{\cdot\}$ and $\Im\{\cdot\}$ respectively denotes the real and imaginary parts of the complex argument. $\mathrm{diag}(\cdot)$ denotes a diagonal matrix. $\mathbb{R}$ and $\mathbb{C}$ denote the set of real and complex numbers, respectively, and j is the unit imaginary number satisfying $j^2 = -1$. $\mathcal{CN}(0, \sigma^2)$ denotes a zero-mean circularly symmetric Gaussian random variable with variance σ^2, $\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{\tau^2}{2}} d\tau$ is the cumulative distribution function of the standard Gaussian random variable and $\sigma(t) = 1/(1 + e^{-t})$ is the Sigmoid function. If $\Re\{\cdot\}$, $\Im\{\cdot\}$, $\Phi(\cdot)$, and $\sigma(\cdot)$ are applied to a matrix or vector, they are applied separately to every element of that matrix or vector.

II. Linear Receivers for First-Stage Detection

This section introduces different types of linear receivers for massive MIMO systems with one-bit ADCs. We first present conventional linear receivers and then use the Bussgang decomposition to introduce three Bussgang-based linear receivers including Bussgang-based maximal ratio combining (BMRC), Bussgang-based zero-forcing (BZF), and Bussgang-based minimum mean squared error (BMMSE).

A. System Model

We consider an uplink massive MIMO system as illustrated in Fig. 1 with K single-antenna users and an N-antenna base station, where it is assumed that $N \geq K$. Let $\bar{\mathbf{x}} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_K]^T \in \mathbb{C}^K$ denote the transmitted signal vector, where $\bar{x}_k$ is the signal transmitted from the k^{th} user

under the power constraint $\mathbb{E}[|\bar{x}_k|^2] = 1$. The signal $\bar{x}_k$ is drawn from a constellation $\bar{\mathcal{M}}$, e.g, QPSK or 16-QAM. Let $\bar{\mathbf{H}} \in \mathbb{C}^{N\times K}$ denote the channel, which is assumed to be block flat fading with elements that are assumed to be independent and identically distributed (i.i.d.) as $\mathcal{CN}(0,1)$. Let $\bar{\mathbf{r}} = [\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_N]^T \in \mathbb{C}^N$ be the unquantized received signal vector at the base station, which is given as

$$\bar{\mathbf{r}} = \bar{\mathbf{H}}\bar{\mathbf{x}} + \bar{\mathbf{z}}, \tag{1}$$

where $\bar{\mathbf{z}} = [\bar{z}_1, \bar{z}_2, \ldots, \bar{z}_N]^T \in \mathbb{C}^N$ is a noise vector whose elements are assumed to be i.i.d. as $\mathcal{CN}(0, N_0)$, and N_0 is the noise power. Each analog received signal is then quantized by a pair of one-bit ADCs. Hence, we have the received signal

$$\bar{\mathbf{y}} = \operatorname{sign}\left(\Re\{\bar{\mathbf{r}}\}\right) + j \operatorname{sign}\left(\Im\{\bar{\mathbf{r}}\}\right) \tag{2}$$

where $\operatorname{sign}(\cdot)$ represents the one-bit ADC with $\operatorname{sign}(r) = +1$ if $r \geq 0$ and $\operatorname{sign}(r) = -1$ if $r < 0$. The operator $\operatorname{sign}(\cdot)$ of a matrix or vector is applied separately to every element of that matrix or vector. The SNR is defined as $\rho = 1/N_0$.

Instead of assuming 1-bit ADCs with ± 1 outputs, in some situations the outputs are scaled to obtain

$$\bar{\mathbf{y}}_{\mathcal{Q}} = \delta\bar{\mathbf{y}} = \delta\big(\operatorname{sign}\left(\Re\{\bar{\mathbf{r}}\}\right) + j \operatorname{sign}\left(\Im\{\bar{\mathbf{r}}\}\right)\big), \tag{3}$$

where δ is chosen to minimize the variance of the quantization error. Due to the scaling assumed in our system model, this results in $\delta = \sqrt{(K+N_0)/\pi}$. Hence, while each real/imaginary element in $\bar{\mathbf{y}}$ belongs to the set $\{\pm 1\}$, each real/imaginary element in $\bar{\mathbf{y}}_{\mathcal{Q}}$ belongs to the set $\{\pm\delta\}$.

Given a received signal vector $\bar{\mathbf{y}}$ (or $\bar{\mathbf{y}}_{\mathcal{Q}}$) and a linear receiver represented by a combining matrix $\mathbf{W} \in \mathbb{C}^{K\times N}$, the demultiplexing task is performed as $\acute{\mathbf{x}} = \mathbf{W}\bar{\mathbf{y}}$ (or $\acute{\mathbf{x}} = \mathbf{W}\bar{\mathbf{y}}_{\mathcal{Q}}$). The signal $\acute{\mathbf{x}}$ is then equalized before symbol-by-symbol detection is performed. In the following, we present different structures for the combining matrix $\mathbf{W}$. The discussion in the following sections assumes that the channel $\bar{\mathbf{H}}$ is available at the base station, but in practice an estimate of $\bar{\mathbf{H}}$ would be used instead.

B. Conventional Linear Receivers

Here, we consider the output signal in (3). A straightforward strategy to obtain linear receivers for one-bit massive MIMO systems is to simply ignore the non-linear effect of the one-bit ADCs and use the conventional linear receivers designed for massive MIMO systems with infinite-resolution ADCs, as follows:

- MRC receiver

$$\mathbf{W}_{\texttt{MRC}} = \operatorname{diag}\left(\bar{\mathbf{H}}^H\bar{\mathbf{H}}\right)^{-1}\bar{\mathbf{H}}^H,$$

- ZF receiver

$$\mathbf{W}_{\texttt{ZF}} = \left(\bar{\mathbf{H}}^H\bar{\mathbf{H}}\right)^{-1}\bar{\mathbf{H}}^H,$$

- MMSE receiver

$$\mathbf{W}_{\texttt{MMSE}} = \left(\bar{\mathbf{H}}^H\bar{\mathbf{H}} + N_0\mathbf{I}_K\right)^{-1}\bar{\mathbf{H}}^H.$$

In another strategy, the nonlinear effect of the one-bit ADCs can be linearized by the Additive Quantization Noise Model (AQNM) [29], [30] as

$$\bar{\mathbf{y}}_{\mathcal{Q}} = \eta\bar{\mathbf{r}} + \bar{\mathbf{d}} = \eta\bar{\mathbf{H}}\bar{\mathbf{x}} + \eta\bar{\mathbf{z}} + \bar{\mathbf{d}}, \tag{4}$$

where $\eta = 1 - \lambda$ and λ is the inverse of the signal-to quantization-noise ratio, which is given by $\lambda = 1 - 2/\pi$ for one-bit ADCs [30]. The quantization distortion $\bar{\mathbf{d}}$ is treated as additive Gaussian noise $\bar{\mathbf{d}} \sim \mathcal{CN}(\mathbf{0}, \mathbf{\Sigma}_{\bar{d}})$ that is uncorrelated with $\bar{\mathbf{r}}$, where $\mathbf{\Sigma}_{\bar{d}} = \lambda\eta \operatorname{diag}(\bar{\mathbf{H}}\bar{\mathbf{H}}^H + N_0\mathbf{I}_N)$. The MMSE receiver for the model in (4) is given as [17]

$$\mathbf{W}_{\texttt{AQNM-MMSE}} = \frac{1}{\eta}\bar{\mathbf{H}}^H\left(\bar{\mathbf{H}}\bar{\mathbf{H}}^H + \frac{1}{\eta^2}\mathbf{\Sigma}_{\bar{d}} + N_0\mathbf{I}_N\right)^{-1}. \tag{5}$$

Another approximate MMSE receiver for quantized MIMO systems, referred to as the "Wiener Filter on Quantized data" (WFQ), is proposed in [18] as

$$\mathbf{W}_{\texttt{WFQ}} = \bar{\mathbf{H}}^H\left(\eta\mathbf{\Sigma}_{\bar{r}} + \lambda \operatorname{diag}(\mathbf{\Sigma}_{\bar{r}})\right)^{-1}, \tag{6}$$

where $\mathbf{\Sigma}_{\bar{r}} = \bar{\mathbf{H}}\bar{\mathbf{H}}^H + N_0\mathbf{I}_N$ is the covariance matrix of $\bar{\mathbf{r}}$. It is interesting to note that the receivers in (5) and (6) are in fact the same, and will be shown later to yield identical performance in Section V.

Once a combining matrix $\mathbf{W}$ has been computed, the demultiplexing task can be performed as $\acute{\mathbf{x}} = \mathbf{W}\bar{\mathbf{y}}_{\mathcal{Q}}$. Since $\|\acute{\mathbf{x}}\|^2$ may not equal K, the signal $\acute{\mathbf{x}}$ should be rescaled as [6]

$$\dot{\mathbf{x}} = [\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_K]^T = \sqrt{K}\frac{\acute{\mathbf{x}}}{\|\acute{\mathbf{x}}\|_2}. \tag{7}$$

Finally, $\dot{\mathbf{x}}$ can be used for symbol-by-symbol detection as

$$\hat{x}_k = \arg\max_{\bar{x}\in\bar{\mathcal{M}}} |\bar{x} - \dot{x}_k|. \tag{8}$$

C. Bussgang-Based Linear Receivers

Here, we exploit the Bussgang decomposition to linearize the system model in (2) and then use the linearized model to derive BMRC, BZF, and BMMSE receiver structures. Following the Bussgang decomposition, the system model in (2) can be rewritten as $\bar{\mathbf{y}} = \bar{\mathbf{V}}\bar{\mathbf{r}} + \bar{\mathbf{e}}$ [31] where $\bar{\mathbf{e}}$ is the quantization distortion, which is uncorrelated with $\bar{\mathbf{r}}$, i.e., $\mathbb{E}\big[\bar{\mathbf{r}}\bar{\mathbf{e}}^H\big] = \mathbb{E}\big[\bar{\mathbf{r}}\big]\mathbb{E}\big[\bar{\mathbf{e}}^H\big]$, and

$$\bar{\mathbf{V}} = \sqrt{\frac{2}{\pi}} \operatorname{diag}(\mathbf{\Sigma}_{\bar{r}})^{-\frac{1}{2}}. \tag{9}$$

Let $\bar{\mathbf{A}} = \bar{\mathbf{V}}\bar{\mathbf{H}}$ and $\bar{\mathbf{n}} = \bar{\mathbf{V}}\bar{\mathbf{z}} + \bar{\mathbf{e}}$, so the system model becomes

$$\bar{\mathbf{y}} = \bar{\mathbf{A}}\bar{\mathbf{x}} + \bar{\mathbf{n}}, \tag{10}$$

where $\bar{\mathbf{A}} = \sqrt{2/\pi} \operatorname{diag}(\mathbf{\Sigma}_{\bar{r}})^{-\frac{1}{2}}\bar{\mathbf{H}}$ is the effective channel and $\bar{\mathbf{n}}$ is the effective noise, which is modeled as Gaussian with zero mean and covariance matrix [31]:

$$\mathbf{\Sigma}_{\bar{n}} = \frac{2}{\pi}\Big[\arcsin\Big(\operatorname{diag}(\mathbf{\Sigma}_{\bar{r}})^{-\frac{1}{2}}\mathbf{\Sigma}_{\bar{r}}\operatorname{diag}(\mathbf{\Sigma}_{\bar{r}})^{-\frac{1}{2}}\Big) - \operatorname{diag}(\mathbf{\Sigma}_{\bar{r}})^{-\frac{1}{2}}\mathbf{\Sigma}_{\bar{r}}\operatorname{diag}(\mathbf{\Sigma}_{\bar{r}})^{-\frac{1}{2}} + N_0\operatorname{diag}(\mathbf{\Sigma}_{\bar{r}})^{-1}\Big]. \tag{11}$$

Note that $\arcsin(\mathbf{C}) = \arcsin(\Re\{\mathbf{C}\}) + j\arcsin(\Im\{\mathbf{C}\})$ for any complex matrix $\mathbf{C}$, and the operation $\arcsin(\cdot)$ of a real matrix is applied separately on each element of that matrix.

Based on the effective channel $\bar{\mathbf{A}}$, we can derive a BMRC receiver as

$$\mathbf{W}_{\texttt{BMRC}} = \operatorname{diag}\left(\bar{\mathbf{A}}^H\bar{\mathbf{A}}\right)^{-1}\bar{\mathbf{A}}^H, \tag{12}$$

and a BZF receiver as [19], [32]

$$\mathbf{W}_{\texttt{BZF}} = \left(\bar{\mathbf{A}}^H\bar{\mathbf{A}}\right)^{-1}\bar{\mathbf{A}}^H. \tag{13}$$

We now derive the BMMSE receiver for this Bussgang-based system model. The BMMSE receiver can be obtained by solving the following optimization problem:

$$\underset{\{\mathbf{W}\}}{\text{minimize}} \quad \mathbb{E}\big[\|\bar{\mathbf{x}} - \mathbf{W}\bar{\mathbf{y}}\|_2^2\big], \tag{14}$$

whose solution is given in closed form as follows:

$$\mathbf{W}_{\texttt{BMMSE}} = \mathbb{E}\big[\bar{\mathbf{x}}\bar{\mathbf{y}}^H\big]\big(\mathbb{E}\big[\bar{\mathbf{y}}\bar{\mathbf{y}}^H\big]\big)^{-1}. \tag{15}$$

We can expand $\mathbb{E}\big[\bar{\mathbf{x}}\bar{\mathbf{y}}^H\big] = \mathbb{E}\left[\bar{\mathbf{x}}\bar{\mathbf{x}}^H\bar{\mathbf{A}}^H\right] + \mathbb{E}\left[\bar{\mathbf{x}}\mathbf{n}^H\right] = \bar{\mathbf{A}}^H$ due to $\mathbb{E}\big[\bar{\mathbf{x}}\bar{\mathbf{x}}^H\big] = \mathbf{I}_K$ and $\mathbb{E}\left[\bar{\mathbf{x}}\mathbf{n}^H\right] = \mathbf{0}$. We have $\mathbb{E}\big[\bar{\mathbf{x}}\bar{\mathbf{n}}^H\big] = \mathbf{0}$ since

$$\mathbb{E}\big[\bar{\mathbf{x}}\bar{\mathbf{n}}^H\big] = \mathbb{E}\big[\bar{\mathbf{x}}(\bar{\mathbf{V}}\bar{\mathbf{z}} + \bar{\mathbf{e}})^H\big] = \mathbb{E}\big[\bar{\mathbf{x}}\bar{\mathbf{z}}^H\big]\bar{\mathbf{V}}^H + \mathbb{E}\big[\bar{\mathbf{x}}\bar{\mathbf{e}}^H\big],$$

where $\mathbb{E}\big[\bar{\mathbf{x}}\bar{\mathbf{z}}^H\big] = \mathbb{E}\big[\bar{\mathbf{x}}\big]\mathbb{E}\big[\bar{\mathbf{z}}^H\big] = \mathbf{0}$ and $\mathbb{E}\big[\bar{\mathbf{x}}\bar{\mathbf{e}}^H\big] = \mathbf{0}$. The result $\mathbb{E}\big[\bar{\mathbf{x}}\bar{\mathbf{e}}^H\big] = \mathbf{0}$ holds because

$$\mathbb{E}\big[\bar{\mathbf{r}}\bar{\mathbf{e}}^H\big] = \bar{\mathbf{H}}\mathbb{E}\big[\bar{\mathbf{x}}\bar{\mathbf{e}}^H\big] + \mathbb{E}\big[\bar{\mathbf{z}}\bar{\mathbf{e}}^H\big], \tag{16}$$

where the left hand side of (16) is $\mathbb{E}\big[\bar{\mathbf{r}}\bar{\mathbf{e}}^H\big] = \mathbb{E}\big[\bar{\mathbf{r}}\big]\mathbb{E}\big[\bar{\mathbf{e}}^H\big] = \mathbf{0}$ and the second term on the right hand side of (16) is also zero (i.e., $\mathbb{E}\big[\bar{\mathbf{z}}\bar{\mathbf{e}}^H\big] = \mathbf{0}$). Therefore, the first term on the right hand side of (16) must also be zero, which implies $\mathbb{E}\big[\bar{\mathbf{x}}\bar{\mathbf{e}}^H\big] = \mathbf{0}$.

In addition, $\mathbb{E}\big[\bar{\mathbf{y}}\bar{\mathbf{y}}^H\big]$ is given by [31]

$$\mathbb{E}\big[\bar{\mathbf{y}}\bar{\mathbf{y}}^H\big] = \frac{2}{\pi}\arcsin\Big(\operatorname{diag}(\boldsymbol{\Sigma}_{\bar{r}})^{-\frac{1}{2}}\boldsymbol{\Sigma}_{\bar{r}}\operatorname{diag}(\boldsymbol{\Sigma}_{\bar{r}})^{-\frac{1}{2}}\Big).$$

Hence, the resulting BMMSE receiver is given as [19], [32]

$$\begin{aligned}\mathbf{W}_{\texttt{BMMSE}} &= \bar{\mathbf{A}}^H\left[\frac{2}{\pi}\arcsin\Big(\operatorname{diag}(\boldsymbol{\Sigma}_{\bar{r}})^{-\frac{1}{2}}\boldsymbol{\Sigma}_{\bar{r}}\operatorname{diag}(\boldsymbol{\Sigma}_{\bar{r}})^{-\frac{1}{2}}\Big)\right]^{-1}\\ &= \bar{\mathbf{A}}^H\left(\bar{\mathbf{A}}\bar{\mathbf{A}}^H + \boldsymbol{\Sigma}_{\bar{n}}\right)^{-1},\end{aligned} \tag{17}$$

where the second equality comes from the equivalent model in (10) and the expression for $\boldsymbol{\Sigma}_{\bar{n}}$ in (11). It can be seen that the structure of the BMMSE receiver is similar to the that of the MMSE receiver, except that the BMMSE receiver applies a new effective channel and a new effective noise covariance. These differences come as the result of linearizing the system model with the Bussgang decomposition.

Since the Bussgang-based linear receivers are derived for the 1-bit ADCs whose output is ± 1, the demultiplexing task here is performed as $\acute{\mathbf{x}} = \mathbf{W}\bar{\mathbf{y}}$. The rescaling step and symbol-by-symbol detection are the same as in (7) and (8).

III. OBMNet for First-Stage Detection

In this section, we first reformulate the conventional ML rule for one-bit MIMO systems, which is then exploited to devise OBMNet. We consider the same system model as presented in Section II, but for convenience in later derivations, we convert (1) and (2) into the real domain as follows:

$$\mathbf{y} = \operatorname{sign}\left(\mathbf{H}\mathbf{x} + \mathbf{z}\right), \tag{18}$$

where

$$\begin{aligned}\mathbf{y} &= \begin{bmatrix}\Re\{\bar{\mathbf{y}}\}\\ \Im\{\bar{\mathbf{y}}\}\end{bmatrix} \in \mathbb{R}^{2N},\ \mathbf{x} = \begin{bmatrix}\Re\{\bar{\mathbf{x}}\}\\ \Im\{\bar{\mathbf{x}}\}\end{bmatrix} \in \mathbb{R}^{2K},\\ \mathbf{z} &= \begin{bmatrix}\Re\{\bar{\mathbf{z}}\}\\ \Im\{\bar{\mathbf{z}}\}\end{bmatrix} \in \mathbb{R}^{2N},\ \text{and}\\ \mathbf{H} &= \begin{bmatrix}\Re\{\bar{\mathbf{H}}\} & -\Im\{\bar{\mathbf{H}}\}\\ \Im\{\bar{\mathbf{H}}\} & \Re\{\bar{\mathbf{H}}\}\end{bmatrix} \in \mathbb{R}^{2N\times 2K}.\end{aligned}$$

We also denote $\mathbf{y} = [y_1, \ldots, y_{2N}]^T$, $\mathbf{x} = [x_1, \ldots, x_{2K}]^T$, $\mathbf{z} = [z_1, \ldots, z_{2N}]^T$, and $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_{2N}]^T$.

The conventional ML detection problem [6] for one-bit ADCs is given as

$$\hat{\mathbf{x}}_{\texttt{ML}} = \underset{\bar{\mathbf{x}}\in\bar{\mathcal{M}}^K}{\arg\max} \prod_{n=1}^{2N}\Phi(\sqrt{2\rho}y_n\hat{\mathbf{h}}_n^T\mathbf{x}), \tag{19}$$

which can also be written as

$$\hat{\mathbf{x}}_{\texttt{ML}} = \underset{\bar{\mathbf{x}}\in\bar{\mathcal{M}}^K}{\arg\max} \sum_{n=1}^{2N}\log\Phi(\sqrt{2\rho}y_n\hat{\mathbf{h}}_n^T\mathbf{x}), \tag{20}$$

where $\hat{\mathbf{h}}_n$ is an estimate of $\mathbf{h}_n$ for $n \in \{1, \ldots, 2N\}$. The ML detection formulations in (19) and (20) are however non-robust at high SNRs when $\hat{\mathbf{h}}_n \neq \mathbf{h}_n$, or in other words, when the CSI is imperfectly known. To see this, assume that $\mathbf{x}^\star$ is the transmitted data vector, and note that it is quite possible that $\operatorname{sign}(\hat{\mathbf{h}}_n^T\mathbf{x}^\star) \neq y_n$ for some n, which would make $y_n\hat{\mathbf{h}}_n^T\mathbf{x}^\star$ negative. Since the function $\Phi(\cdot)$ approaches 0 exponentially fast, the term $\log\Phi(\sqrt{2\rho}y_n\hat{\mathbf{h}}_n^T\mathbf{x}^\star)$ will tend to $-\infty$ in such cases even for a moderate SNR value such as 20 dB. If there exists another data vector $\mathbf{x} \neq \mathbf{x}^\star$ that satisfies $y_n\hat{\mathbf{h}}_n^T\mathbf{x} > 0, \forall n$, a detection error will surely occur. If this is not the case, the objective function value in (20) tends to $-\infty$ for all possible data vectors including the transmitted data vector. A detection error will almost surely occur since any data vector can be chosen as a solution to problem (20) . This non-robustness under imperfect CSI has been numerically reported in [25], [26] and a detailed explanation of this issue can be found in [28, Appendix A].

To address the non-robustness of the above ML formulation, we exploit a result in [33], which shows that the function $\Phi(t)$ can be accurately approximated by the Sigmoid function $\sigma(t)$, which is a widely-used activation function in machine learning research. The approximation of $\Phi(t)$ is given as

$$\Phi(t) \approx \sigma(ct) = \frac{1}{1+e^{-ct}}, \tag{21}$$

where $c = 1.702$ is a constant. It was shown in [33] that $|\Phi(t) - \sigma(ct)| \leq 0.0095$, $\forall t \in \mathbb{R}$. Thus, maximizing $\log\Phi(t)$ is approximately equivalent to minimizing $\log(1+e^{-ct})$.

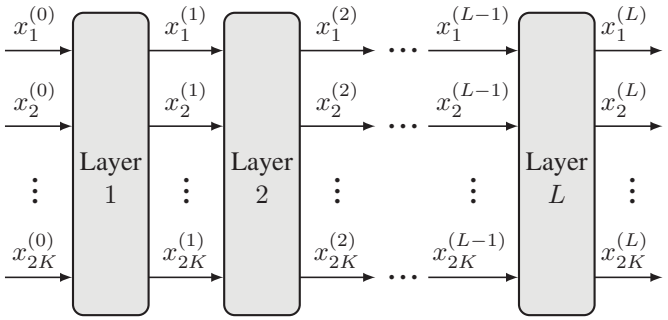


Fig. 2: Overall structure of the proposed OBMNet.

Applying the approximation in (21) to (20), we obtain the following ML detection problem:

$$\hat{\mathbf{x}}_{\mathtt{ML}}^{\text{robust}} = \arg\min_{\bar{\mathbf{x}}\in\bar{\mathcal{M}}^K} \sum_{n=1}^{2N} \log\left(1+e^{-c\sqrt{2\rho}y_n\hat{\mathbf{h}}_n^T\bar{\mathbf{x}}}\right). \tag{22}$$

The reformulated ML detection problem (22) does not share the non-robustness issue of (20), since if $\sqrt{2\rho}y_n\hat{\mathbf{h}}_n^T\mathbf{x}^\star$ is largely negative (due to $\text{sign}(\hat{\mathbf{h}}_n^T\mathbf{x}^\star) \neq y_n$ and large ρ), we have $\log(1+e^{-c\sqrt{2\rho}y_n\hat{\mathbf{h}}_n^T\mathbf{x}^\star}) \approx -c\sqrt{2\rho}y_n\hat{\mathbf{h}}_n^T\mathbf{x}^\star$. This approximation holds because $\log(1+e^t)\approx t$ for large t. Note that the value of $-c\sqrt{2\rho}y_n\hat{\mathbf{h}}_n^T\mathbf{x}^\star$ is finite for large ρ, and thus so is the objective function in (22) for all possible data vectors. Therefore, the reformulated ML detection problem is more robust and (22) is more likely to yield $\mathbf{x}^\star$ as the optimal solution, unlike problem (20). It is interesting to note that $\log(1+e^t)$ is referred to as the SoftPlus activation function in the machine learning literature. Hence, the proposed robust ML detection problem in (22) can be interpreted as a minimization problem whose objective is a sum of SoftPlus activation functions. Note that we have $\log(1+e^t)\approx t$ for large t. However, a sequential computation by first evaluating e^t then the log function may result in an infinite value since e^t grows very rapidly. Hence, one should use the approximation $\log(1+e^t)\approx t$ when t is large, e.g., $t>100$.

Now, we develop the OBMNet detector based on the proposed robust ML detection problem in (22). We relax the constraint $\bar{\mathbf{x}}\in\bar{\mathcal{M}}^K$ in (22) to $\bar{\mathbf{x}}\in\mathbb{C}^K$ and denote the channel estimate $\hat{\mathbf{H}}=[\hat{\mathbf{h}}_1,\ldots,\hat{\mathbf{h}}_{2N}]^T$. Let $\mathbf{G}=\text{diag}(y_1,\ldots,y_{2N})\hat{\mathbf{H}}$ and define the rows of $\mathbf{G}$ as $\mathbf{G}=[\mathbf{g}_1,\ldots,\mathbf{g}_{2N}]^T$. Then (22) can be rewritten as

$$\arg\min_{\bar{\mathbf{x}}\in\mathbb{C}^K} \underbrace{\sum_{n=1}^{2N} \log\left(1+e^{-c\sqrt{2\rho}\mathbf{g}_n^T\bar{\mathbf{x}}}\right)}_{\mathcal{P}(\bar{\mathbf{x}})}. \tag{23}$$

The gradient of $\mathcal{P}(\bar{\mathbf{x}})$ is

$$\begin{aligned}\nabla\mathcal{P}(\bar{\mathbf{x}}) &= \sum_{n=1}^{2N} \frac{-c\sqrt{2\rho}\,\mathbf{g}_n}{1+e^{c\sqrt{2\rho}\,\mathbf{g}_n^T\bar{\mathbf{x}}}} \\ &= -c\sqrt{2\rho}\mathbf{G}^T\sigma\big(-c\sqrt{2\rho}\mathbf{G}\bar{\mathbf{x}}\big).\end{aligned} \tag{24}$$

Hence, an iterative gradient descent method can be used to solve (23) as follows:

$$\bar{\mathbf{x}}^{(\ell)} = \bar{\mathbf{x}}^{(\ell-1)} + \alpha_\ell c\sqrt{2\rho}\,\mathbf{G}^T\sigma\left(-c\sqrt{2\rho}\mathbf{G}\bar{\mathbf{x}}^{(\ell-1)}\right) \tag{25}$$

where ℓ is the iteration index and α_ℓ is the step size.

In order to optimize the step sizes $\{\alpha_\ell\}$, we use the *deep unfolding* technique [34] to unfold each iteration in (25) as a layer of a deep neural network. The overall structure of the proposed OBMNet is illustrated in Fig. 2, where there are L layers and each layer takes a vector of $2K$ elements as the input and generates an output vector of the same size. The specific structure for each layer ℓ is illustrated in Fig. 3.

It can be seen that the proposed layer structure in Fig. 3 is different from that of conventional DNNs, since it exploits the specific structure of the ML detection problem. In particular, each layer of a conventional DNN often contains a weight matrix and a bias vector to be trained. However, due to the structure of the ML detection problem, the proposed OBMNet contains only $L+1$ trainable parameters including L step sizes $\{\alpha_1,\ldots,\alpha_L\}$ and a scaling parameter β inside the Sigmoid function. The proposed layer structure has two weight matrices $-\mathbf{G}$ and $\mathbf{G}^T$ and no bias vector, and the weight matrices are defined by the channel estimate and the received signal.

Since $\mathbf{G}\in\mathbb{R}^{2N\times 2K}$, the learning process of each layer can be interpreted as first up-converting the signal from dimension $2K$ to dimension $2N$ using the weight matrix $-\mathbf{G}$, then applying nonlinear activation functions before down-converting the signal back to dimension $2K$ using the weight matrix $\mathbf{G}^T$. The activation function in OBMNet is the Sigmoid function, which is also widely used in conventional DNNs. Note that the use of the Sigmoid activation function in OBMNet is not arbitrary but results from the use of the approximation in (21) and the structure of the ML detection problem.

The objective function to be minimized during the training phase is $\|\tilde{\mathbf{x}}-\mathbf{x}\|^2$, where

$$\tilde{\mathbf{x}} = \frac{\sqrt{K}}{\|\mathbf{x}^{(L)}\|}\mathbf{x}^{(L)} \tag{26}$$

and $\mathbf{x}$ is the target signal, i.e., the transmitted signal. It should also be noted that the layered structure in Fig. 3 does not contain the coefficient $c\sqrt{2\rho}$. We omit this coefficient because it is a constant throughout the layers of OBMNet, and the output of the last layer $\mathbf{x}^{(L)}$ needs to be normalized as in (26). We found by experiments that this omission not only helps improve the detection performance but also helps the training process to stably converge.

The training process is accomplished offline. A training sample can be obtained by randomly generating a channel matrix $\mathbf{H}$, a transmitted signal $\mathbf{x}$, and a noise vector $\mathbf{z}$. The received signal $\mathbf{y}$ and the channel $\mathbf{H}$ are used to build the weight matrices and the transmitted signal $\mathbf{x}$ is used as the target. After the offline training processing, the trained step sizes $\{\alpha_\ell\}$ and the trained scaling parameter β are ready to be used for the online detection phase. Similar to DetNet for unquantized MIMO detection [21], OBMNet for one-bit MIMO detection does not need to be retrained for a new channel realization $\mathbf{H}$.

IV. Nearest-Neighbor Search for Second-Stage Detection

Given a received signal, as discussed above we can either use a linear receiver or OBMNet to obtain an estimate $\tilde{\mathbf{x}}$ of

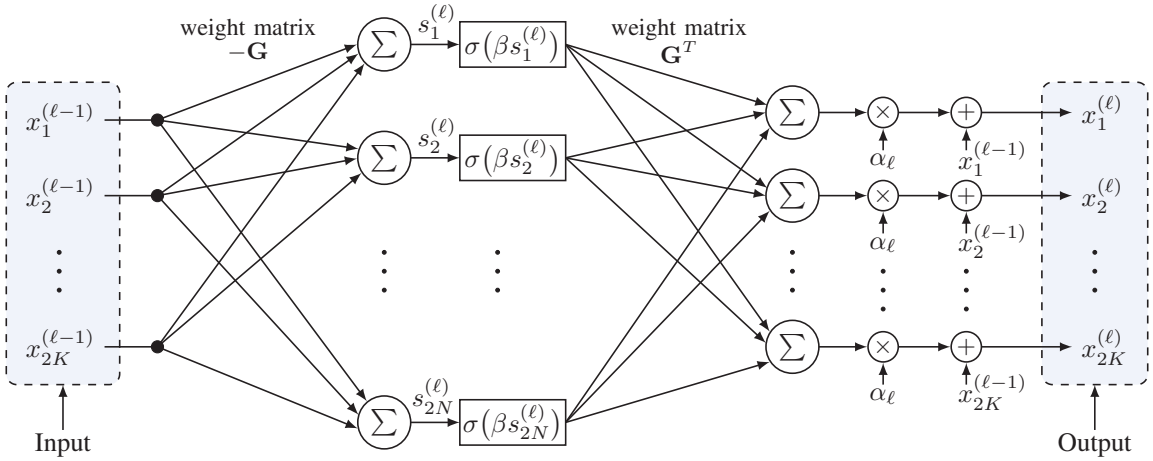


Fig. 3: Specific structure of layer ℓ of OBMNet. The weight matrices are defined by the channel and the received signal. There is no bias vector.

the transmitted signal $\mathbf{x}$. However, these receivers all ignore the constraint that the transmitted signal $\mathbf{x}$ belongs to a known discrete set of constellation points. Ignoring this constraint can result in elements of the estimate $\tilde{\mathbf{x}}$ that are well removed from the constellation points, and thus detection errors are likely to occur once symbol-by-symbol detection is applied. This motivates us to propose here an NN search method as a second detection stage in order to fine-tune the solution of stage 1.

The proposed NN search method first finds a limited set of symbol vectors that are nearest to $\tilde{\mathbf{x}}$ and then searches over that set for the most likely symbol vector as the final detection solution. As mentioned in the Introduction section, this idea has already been used in [6] and [28]. However, the search space for the methods in [6] and [28] is very large when the number of users is large, and so they are not efficient in terms of computational complexity. The contribution of the proposed NN search method is that it generates searches over a limited number of symbol vectors that are nearest to the estimate $\tilde{\mathbf{x}}$, and thus significantly reduces the computational load.

We denote $\mathcal{M}$ as the constellation in the real domain; for example, $\mathcal{M} = \left\{\pm\frac{1}{\sqrt{2}}\right\}$ for QPSK and $\mathcal{M} = \left\{\pm\frac{1}{\sqrt{10}}, \pm\frac{3}{\sqrt{10}}\right\}$ for 16-QAM. Let $\mathcal{B}$ be the set of decision boundary points; *i.e.,* $\mathcal{B} = \{0\}$ for QPSK and $\mathcal{B} = \left\{0, \pm\frac{2}{\sqrt{10}}\right\}$ for 16-QAM. Denote $\tilde{\mathbf{x}} = [\tilde{x}_1, \ldots, \tilde{x}_{2K}]^T$ and $\mathbf{b} = [b_1, \ldots, b_{2K}]^T$, where b_i is the decision boundary point that is nearest to $\tilde{x}_i$, as follows:

$$b_i = \arg\min_{b\in\mathcal{B}} |b - \tilde{x}_i|, \quad i \in \{1, 2, \ldots, 2K\}. \tag{27}$$

An illustrative example for the relative difference between $\tilde{x}_i$ and the constellation points is given in Fig. 4. This example illustrates the problem that occurs when $\tilde{x}_i$ is close to a decision boundary point, where symbol-by-symbol detection may not be reliable. Here, we use a threshold $\gamma > 0$ to classify whether symbol-by-symbol detection is used or not. More specifically, if the distance from $\tilde{x}_i$ to its nearest decision boundary point b_i is greater than γ, i.e., $|\tilde{x}_i - b_i| > \gamma$, then we can use symbol-by-symbol detection for $\tilde{x}_i$. When $|\tilde{x}_i - b_i| \le \gamma$, symbol-by-symbol detection is not reliable, and

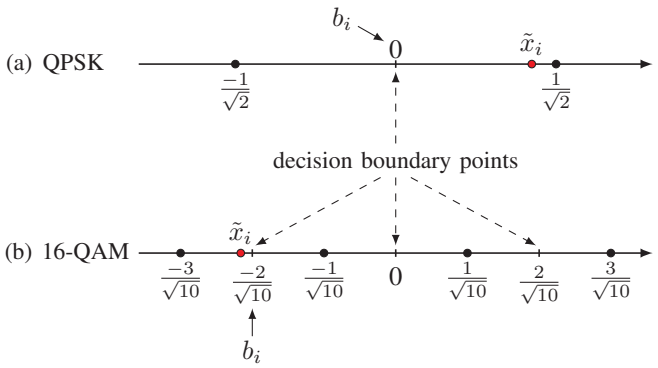


Fig. 4: An example for the relative difference between $\tilde{x}_i$ and the constellation points: (a) the estimate $\tilde{x}_i$ is far from $b_i = 0$ and close to the constellation point $1/\sqrt{2}$, which means there is a high probability that the transmitted signal x_i is $1/\sqrt{2}$; (b) the estimate $\tilde{x}_i$ is close to the boundary point $b_i = -2/\sqrt{10}$, thus it is difficult to say if $-3/\sqrt{10}$ or $-1/\sqrt{10}$ was transmitted.

so we list the two nearest constellation points to $\tilde{x}_i$ as the candidates for the transmitted signal x_i.

Let $\mathcal{A}_i$ denote the set of candidates for the transmitted signal x_i. When $|\tilde{x}_i - b_i| > \gamma$, we apply symbol-by-symbol detection and so

$$\mathcal{A}_i = \left\{\arg\min_{x\in\mathcal{M}} |x - \tilde{x}_i|\right\}.$$

When $|\tilde{x}_i - b_i| \le \gamma$, we have $\mathcal{A}_i = \left\{b_i \pm \frac{1}{\sqrt{2}}\right\} = \left\{\pm\frac{1}{\sqrt{2}}\right\}$ for QPSK and $\mathcal{A}_i = \left\{b_i \pm \frac{1}{\sqrt{10}}\right\}$ for 16-QAM. Hence, $\mathcal{A}_i$ contains only one or two elements. The following example illustrates the formation of $\mathcal{A}_i$.

Example 1. *Suppose that* $\tilde{\mathbf{x}} = [0.1, -0.5, -0.3, 0.8]^T$ *and QPSK modulation is used with* $\gamma = \frac{1}{2\sqrt{2}} \approx 0.35$. *Note here that* $b_1 = b_2 = b_3 = b_4 = 0$. *We have*

- $\mathcal{A}_1 = \mathcal{A}_3 = \left\{\pm\frac{1}{\sqrt{2}}\right\}$ *because* $|\tilde{x}_1 - b_1| = 0.1 < \gamma$ *and* $|\tilde{x}_3 - b_3| = 0.3 < \gamma$,
- $\mathcal{A}_2 = \left\{\frac{-1}{\sqrt{2}}\right\}$ *because* $|\tilde{x}_2 - b_2| = 0.5 > \gamma$ *and* $\tilde{x}_2$ *is closer to* $\frac{-1}{\sqrt{2}}$ *than* $\frac{1}{\sqrt{2}}$, *i.e.,* $\left|\tilde{x}_2 - \frac{-1}{\sqrt{2}}\right| < \left|\tilde{x}_2 - \frac{1}{\sqrt{2}}\right|$,
- $\mathcal{A}_4 = \left\{\frac{1}{\sqrt{2}}\right\}$ *because* $|\tilde{x}_4 - b_4| = 0.8 > \gamma$ *and* $\tilde{x}_4$ *is closer to* $\frac{1}{\sqrt{2}}$ *than* $\frac{-1}{\sqrt{2}}$, *i.e.,* $\left|\tilde{x}_4 - \frac{1}{\sqrt{2}}\right| < \left|\tilde{x}_4 - \frac{-1}{\sqrt{2}}\right|$.

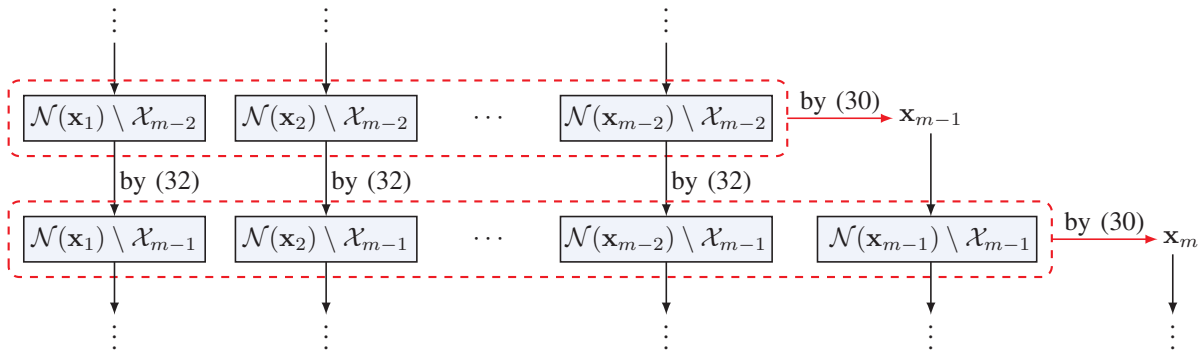


Fig. 5: Flowchart of the proposed nearest-neighbor search method. A recursive formation of sets is exploited to reduce the computational complexity. A subset $\mathcal{N}(\mathbf{x}_p)\setminus\mathcal{X}_{m-1}$ with $p \in \{1,\ldots,m-2\}$ is obtained by removing $\mathbf{x}_{m-1}$ from the subset $\mathcal{N}(\mathbf{x}_p)\setminus\mathcal{X}_{m-2}$ as given in (32). The last subset $\mathcal{N}(\mathbf{x}_{m-1})\setminus\mathcal{X}_{m-1}$ is obtained by using $\mathbf{x}_{m-1}$ and other nearest symbol vectors. The m^{th} nearest symbol vector $\mathbf{x}_m$ is then obtained by searching over the $m-1$ subsets.

Hence, in this example, $\mathcal{A}_1$ and $\mathcal{A}_3$ have two elements while $\mathcal{A}_2$ and $\mathcal{A}_4$ have only one element.

The complete set of candidates for the transmitted signal vector is given by the Cartesian product

$$\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_{2K},$$

and so the size of $\mathcal{A}$ is $|\mathcal{A}| = \prod_{i=1}^{2K} |\mathcal{A}_i| = 2^A$, where A is the number of sets $\mathcal{A}_i$ having two elements. The existing search methods in [6] and [28] always search over the entire set $\mathcal{A}$. However, it can be seen that the size of $\mathcal{A}$ grows exponentially with A. In addition, A also grows as the number of users K increases. Thus, searching over the entire list $\mathcal{A}$ as in [6] and [28] can be prohibitively complex when the number of users is large.

On the other hand, the proposed NN search method finds a set of M symbol vectors in $\mathcal{A}$ that are nearest to $\tilde{\mathbf{x}}$, then searches over that smaller set for the final solution. In this way, the NN search method can limit the computational complexity. Note that a symbol vector in this context is any element of $\mathcal{A}$. Let $\mathcal{X}_M = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ denote the set of the M nearest symbol vectors to $\tilde{\mathbf{x}}$. The larger M is, the higher the probability that the set $\mathcal{X}_M$ contains the true symbol vector. However, a large value of M will result in more computation for the search. Therefore, M should be chosen to achieve a good trade-off between detection accuracy and computational complexity. The value of M can be chosen by empirical evaluations. The main challenge here is how to find the M nearest symbol vectors to $\tilde{\mathbf{x}}$ quickly and efficiently. To address this problem, we employ the following notation and definitions.

For any two symbol vectors $\mathbf{x} \in \mathcal{A}$ and $\mathbf{x}' \in \mathcal{A}$, let $d(\mathbf{x}, \mathbf{x}')$ denote the number of position indices at which the elements of $\mathbf{x}$ are different from the corresponding elements of $\mathbf{x}'$. Since each element of $\mathbf{x}$ and $\mathbf{x}'$ belongs to a finite set of just one or two elements, $d(\mathbf{x}, \mathbf{x}')$ is actually the Hamming distance between $\mathbf{x}$ and $\mathbf{x}'$.

Definition 1 (Neighbor of a symbol vector)**.** *A symbol vector $\mathbf{x}$ is called a neighbor of another symbol vector $\mathbf{x}'$, or vice versa, when the Hamming distance between them is one, i.e., $d(\mathbf{x}, \mathbf{x}') = 1$.*

Definition 2 (Neighbor of a set)**.** *Given a set of symbol vectors $\mathcal{S}$ and another symbol vector $\mathbf{x} \notin \mathcal{S}$, let*

$$d_{\min}(\mathbf{x}, \mathcal{S}) = \min_{\mathbf{x}' \in \mathcal{S}} d(\mathbf{x}, \mathbf{x}'). \tag{28}$$

The symbol vector $\mathbf{x}$ is called a neighbor of $\mathcal{S}$ if and only if $d_{\min}(\mathbf{x}, \mathcal{S}) = 1$, or in other words, if and only if $\mathbf{x}$ is the neighbor of at least one member of $\mathcal{S}$.

Let $\mathcal{N}(\mathbf{x})$ and $\mathcal{N}(\mathcal{S})$ denote the set of neighbors of symbol vector $\mathbf{x}$ and set $\mathcal{S}$, respectively. Let $\mathcal{X}_M = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ with $\mathbf{x}_m \in \mathcal{A}$ and $m \in \{1, 2, \ldots, M\}$ denote the set of the M nearest symbol vectors to $\tilde{\mathbf{x}}$ satisfying

$$\|\mathbf{x}_1 - \tilde{\mathbf{x}}\|^2 < \|\mathbf{x}_2 - \tilde{\mathbf{x}}\|^2 < \ldots < \|\mathbf{x}_M - \tilde{\mathbf{x}}\|^2 < \|\mathbf{x}_{\text{out}} - \tilde{\mathbf{x}}\|^2 \tag{29}$$

where $\mathbf{x}_{\text{out}}$ is any symbol vector in $\mathcal{A}$, but not in $\mathcal{X}_M$. Hence, $\mathbf{x}_m$ is the m^{th} nearest symbol vector to $\tilde{\mathbf{x}}$. Clearly, the nearest symbol vector $\mathbf{x}_1$ is obtained by applying symbol-by-symbol detection to $\tilde{\mathbf{x}}$. The problem now is how to efficiently find $\mathbf{x}_2$, …, $\mathbf{x}_M$. The following proposition can be exploited to solve this problem.

Proposition 1. *The m^{th} nearest symbol vector $\mathbf{x}_m$ must be a neighbor of the set $\mathcal{X}_{m-1} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{m-1}\}$, i.e.,*

$$\mathbf{x}_m \in \mathcal{N}(\mathcal{X}_{m-1}).$$

Proof: Please refer to Appendix A ■

Proposition 1 indicates that we can find the m^{th} nearest symbol vector $\mathbf{x}_m$ from the neighbor set of $\mathcal{X}_{m-1}$, i.e.,

$$\mathbf{x}_m = \underset{\mathbf{x} \in \mathcal{N}(\mathcal{X}_{m-1})}{\arg\min} \|\mathbf{x} - \tilde{\mathbf{x}}\|^2 \tag{30}$$

where $\mathcal{N}(\mathcal{X}_{m-1})$ is the neighbor set of $\mathcal{X}_{m-1}$ and is given as

$$\begin{aligned}\mathcal{N}(\mathcal{X}_{m-1}) &= \left(\bigcup_{p=1}^{m-1} \mathcal{N}(\mathbf{x}_p)\right) \setminus \mathcal{X}_{m-1} \\ &= \bigcup_{p=1}^{m-1} \Big(\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-1}\Big).\end{aligned} \tag{31}$$

Hence, in order to find $\mathbf{x}_m$, we need to accomplish two tasks: (i) find $m-1$ subsets $\{\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-1}\}_{p=1,\ldots,m-1}$ and (ii) search for $\mathbf{x}_m$ within the subsets. The method of directly

Algorithm 1: Proposed Nearest-Neighbor Search.

Input: $\tilde{\mathbf{x}}$, γ, M.
Output: $\hat{\mathbf{x}}$.
1 Find $\mathbf{b}$ and $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{2K}$ based on $\mathbf{b}$;
2 Let $|\mathcal{A}| = \prod_{i=1}^{2K} |\mathcal{A}_i|$;
3 **if** $|\mathcal{A}| \leq M$ **then**
4 Let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_{2K}$;
5 $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{A}} \mathcal{P}(\mathbf{x})$;
6 **else**
7 Find $\mathbf{x}_1$ via symbol-by-symbol detection;
8 Let $\mathcal{C}_1 = \text{sort}\left(\mathcal{N}(\mathbf{x}_1)\right)$;
9 **for** $m = 2$ **to** M **do**
10 Let $\mathcal{S}_m = \{\mathcal{C}_1[1], \mathcal{C}_2[1], \ldots, \mathcal{C}_{m-1}[1]\}$;
11 $\mathbf{x}_m = \arg\min_{\mathbf{x}\in\mathcal{S}_m} \|\mathbf{x} - \tilde{\mathbf{x}}\|^2$;
12 **if** $m < M$ **then**
13 **for** $p = 1$ **to** $m - 1$ **do**
14 **if** $\mathcal{C}_p[1] = \mathbf{x}_m$ **then**
15 Remove $\mathcal{C}_p[1]$ from $\mathcal{C}_p$;
16 **end**
17 **end**
18 Let $\mathcal{C}_m = \text{sort}\left(\mathcal{N}\left(\mathbf{x}_m\right)\right)$;
19 **for** $p = 1$ **to** $m - 1$ **do**
20 **if** $\mathcal{C}_m[1] = \mathbf{x}_p$ **then**
21 Remove $\mathcal{C}_m[1]$ from $\mathcal{C}_m$;
22 **end**
23 **end**
24 **end**
25 **end**
26 $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{X}_M} \mathcal{P}(\mathbf{x})$;
27 **end**
28 **return** $\hat{\mathbf{x}}$;

finding the $m-1$ subsets and then searching them for $\mathbf{x}_m$ is not efficient. In the following, we present a recursive strategy to obtain $\mathbf{x}_m$ quickly and efficiently.

Note that the inner term on the right-hand side of (31) can be written as follows:

$$\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-1} = \Big(\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-2}\Big) \setminus \{\mathbf{x}_{m-1}\}. \tag{32}$$

Therefore, we can exploit (32) to obtain the first $m-2$ subsets $\{\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-1}\}_{p=1,\ldots,m-2}$ by removing $\mathbf{x}_{m-1}$ from $m-2$ other subsets $\{\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-2}\}_{p=1,\ldots,m-2}$, which were already obtained previously when we found $\mathbf{x}_{m-1}$. The last subset $\mathcal{N}(\mathbf{x}_{m-1}) \setminus \mathcal{X}_{m-1}$ is obtained by using $\mathbf{x}_{m-1}$ and the other nearest symbol vectors. A flowchart illustrating this recursive strategy is given in Fig. 5.

Remark 1: If the elements of $\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-2}$ are already sorted in ascending order of distance to $\tilde{\mathbf{x}}$, then $\mathbf{x}_{m-1}$ can be removed from $\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-2}$ by simply checking the first element of $\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-2}$. The reason for this is that $\mathbf{x}_{m-1}$ is the $(m-1)^{\text{th}}$ nearest symbol vector, which means the distance from $\mathbf{x}_{m-1}$ to $\tilde{\mathbf{x}}$ cannot be greater than the distance from any element of $\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-2}$ to $\tilde{\mathbf{x}}$. In addition, the elements of $\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-2}$ are distinct and already sorted, and so if $\mathbf{x}_{m-1}$ exists in $\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-2}$, it must be the first element of $\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-2}$.

Remark 2: If the elements of each subset $\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-1}$ are already sorted in ascending order of distance to $\tilde{\mathbf{x}}$, then the search over the $m-1$ subsets for $\mathbf{x}_m$ can be done by simply searching over a list of $m-1$ candidates, where each candidate is the first element of a subset $\mathcal{N}(\mathbf{x}_p) \setminus \mathcal{X}_{m-1}$.

TABLE I: Computational Complexity Comparison: T_{d} is the data block length, N_{iter} is the number of iterations, $\kappa(N)$ is a super-linear function of N, and $GN_{\text{s}} = 2N$.

Method	Preprocessing	Stage 1
OBMNet	–	$\mathcal{O}(KNLT_{\text{d}})$
BMRC	$\mathcal{O}(KN)$	$\mathcal{O}(KNT_{\text{d}})$
BZF [32]	$\mathcal{O}(K^2N)$	
BMMSE [32]	$\mathcal{O}(\max\{KN^2, N^{2.373}\})$	
ADMM [9]	$\mathcal{O}(\max\{KN^2, N^{2.373}\})$	$\mathcal{O}(N^2 N_{\text{iter}} T_{\text{d}})$
SVM-based [28]	–	$\mathcal{O}(KN\kappa(N)T_{\text{d}})$
OSD [7]	$\mathcal{O}(4^{N/G}KN\lvert\bar{\mathcal{M}}\rvert^K)$	$\mathcal{O}\big((N/N_{\text{s}})KNT_{\text{d}}\big)$

Based on the observations in Remarks 1 and 2, we propose the nearest-neighbor search method described in Algorithm 1. The key idea is to use the recursive strategy depicted in Fig. 5 and to implement the observations made in Remarks 1 and 2. Whenever forming a set $\mathcal{N}(\mathbf{x}_m)$, we sort its elements in ascending order of distance to $\tilde{\mathbf{x}}$ as described in lines 8 and 18 of Algorithm 1. In this way, we only need to sort $M-1$ times, and the remainder of the proposed algorithm only involves comparisons based on checking the first elements of the subsets. We denote $\mathcal{C}_1, \ldots, \mathcal{C}_{M-1}$ as the subsets corresponding to $\mathbf{x}_1, \ldots, \mathbf{x}_{M-1}$, respectively, and $\mathcal{C}_m[1]$ denotes the first element of the subset $\mathcal{C}_m$. Lines 10 and 11 implement Remark 2 to obtain $\mathbf{x}_m$. Remark 1 is implemented in lines 13-17. The last subset is obtained in lines 18-23. Finally, line 26 gives the final solution by searching for the highest-likelihood symbol vector among the M nearest symbol vectors.

V. Computational Complexity Analysis and Numerical Results

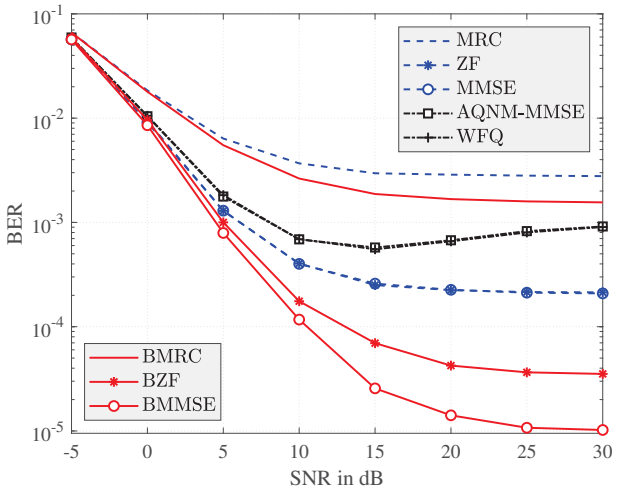
A. Computational Complexity Analysis

A computational complexity comparison in terms of big-$\mathcal{O}$ notation is provided in Table I. It can be seen that the linear receivers have the lowest complexity, while the OSD method in [7] has the highest complexity, which grows exponentially with K and N. Note that the complexity of the SVM-based method [28] is due to the decomposition techniques used to solve the SVM problem, e.g., [35]–[37]. The term $\kappa(N)$ is empirically reported to be a super-linear function of N. The complexity of the proposed OBMNet detector is only $\mathcal{O}(KNLT_{\text{d}})$, which is lower than that of both the ADMM-based and the SVM-based methods.

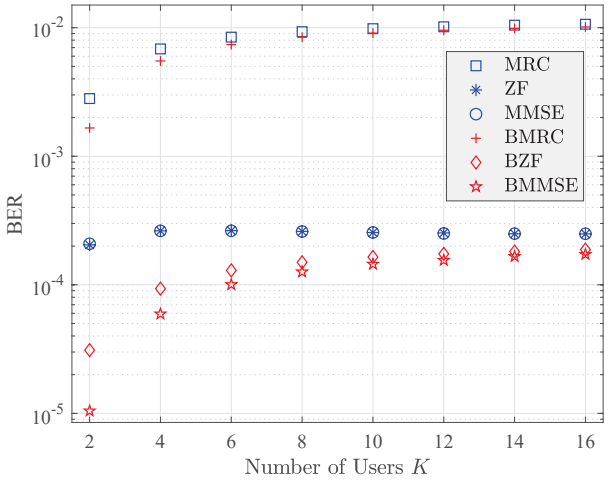
The computational complexity of the proposed NN search method is $\mathcal{O}(MK\max\{M, N\}T_{\text{d}})$ in the worst case. This complexity is mainly due to the detection step for $\hat{\mathbf{x}}$ and the **for** loops as described in Algorithm 1. The complexity of the full $\mathcal{A}$-space search method is $\mathcal{O}(|\mathcal{A}|KNT_{\text{d}})$ where $|\mathcal{A}|$ can grow exponentially with K.

B. Numerical Results

This section presents numerical results to show the performance of the proposed two-stage detection method. The channel elements are assumed to be i.i.d. and each channel element is generated from the normal distribution $\mathcal{CN}(0, 1)$.



(a) $K = 2$, $N = 16$.



(b) $N = 8K$ and $\rho = 30$ dB.

Fig. 6: First stage performance comparison between the conventional and Bussgang-based linear receivers with QPSK signaling.

First, we evaluate the performance of the conventional and Bussgang-based linear receivers assuming perfect CSI is available (examples with estimated CSI will be given next). Fig. 6 presents BER comparisons for QPSK signaling. Among the conventional receivers, we see that the ZF and MMSE receivers obtain the same performance (blue curves with symbols), as do the AQNM-MMSE [17] and WFQ receivers [18] (black curves with symbols). The Bussgang-based linear receivers significantly outperform their conventional counterparts. The high-SNR error floors of the Bussgang-based linear receivers are much lower than those of the conventional approaches. These performance improvements are achieved thanks to the exact linear input-output relationship of massive MIMO systems with one-bit ADCs obtained by the Bussgang decomposition. In Fig. 6b, we evaluate the performance as the number of users K increases. Here, we omit AQNM-MMSE and WFQ since they are outperformed by ZF and MMSE. It is observed that the Bussgang-based linear receivers always yield lower BERs than the standard methods, and the performance improvement is best seen when the number of users K is not too large. As K increases, the gap between the error floors tend to diminish. This is due to the fact that for large K, we have $\bar{\mathbf{H}}\bar{\mathbf{H}}^H \approx K\mathbf{I}_N$, which yields $\mathbf{\Sigma}_{\bar{r}} \approx (K+N_0)\mathbf{I}_N$, $\bar{\mathbf{A}} \approx \sqrt{\mu}\bar{\mathbf{H}}$ and $\mathbf{\Sigma}_{\bar{n}} \approx \big(1-\mu K\big)\mathbf{I}_N$, where $\mu = 2/(\pi(K+N_0))$. These approximations result in Bussgang-based linear receivers that are equivalent to the conventional approaches with a scaling factor:

$$\mathbf{W}_{\text{BMRC}} \approx \sqrt{\frac{1}{\mu}}\,\text{diag}(\bar{\mathbf{H}}^H\bar{\mathbf{H}})^{-1}\bar{\mathbf{H}}^H,$$

$$\mathbf{W}_{\text{BZF}} \approx \sqrt{\frac{1}{\mu}}\Big(\bar{\mathbf{H}}^H\bar{\mathbf{H}}\Big)^{-1}\bar{\mathbf{H}}^H,$$

$$\mathbf{W}_{\text{BMMSE}} \approx \sqrt{\frac{1}{\mu}}\bar{\mathbf{H}}^H\left(\bar{\mathbf{H}}\bar{\mathbf{H}}^H + \frac{1-\mu K}{\mu}\mathbf{I}_N\right)^{-1}.$$

In Fig. 7, we provide BER comparisons between the ZF, MMSE, BZF, and BMMSE linear receivers with estimated CSI for a case with $K = 2$ users and $N = 16$ antennas. Figure 7(a) shows results for the Bussgang-based channel estimator in [15], while Fig. 7(b) employs the SVM-based channel estimator of [28]. It can be seen that the BMMSE receiver always outperforms the others. An interesting observation here is that ZF and MMSE with estimated CSI outperform ZF and MMSE with perfect CSI. There is a reason for this. Recall that Bussgang-based linear receivers BZF and BMMSE use the effective channel

$$\bar{\mathbf{A}} = \sqrt{\frac{2}{\pi}}\,\text{diag}(\bar{\mathbf{H}}\bar{\mathbf{H}}^H + N_0\mathbf{I}_N)^{-1/2}\bar{\mathbf{H}}. \tag{33}$$

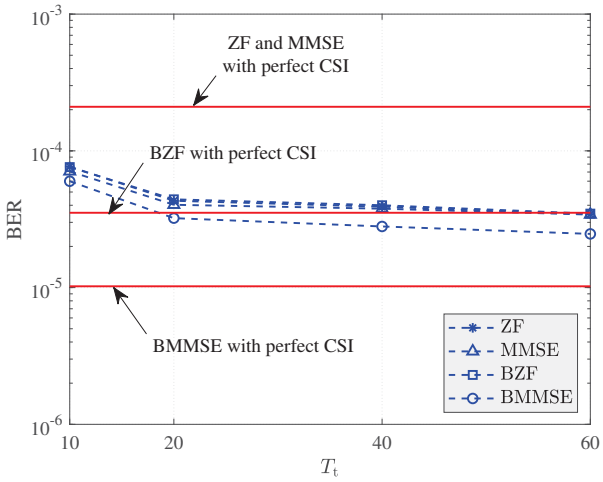
Let $\bar{\mathbf{a}}_i^T$ and $\bar{\mathbf{h}}_i^T$ denote the i^{th} row of $\bar{\mathbf{A}}$ and $\bar{\mathbf{H}}$, respectively, then we have

$$\bar{\mathbf{a}}_i = \sqrt{\frac{2}{\pi}}\frac{\bar{\mathbf{h}}_i}{\sqrt{\|\bar{\mathbf{h}}_i\|^2 + N_0}}, \quad i = 1, 2, \ldots, N. \tag{34}$$

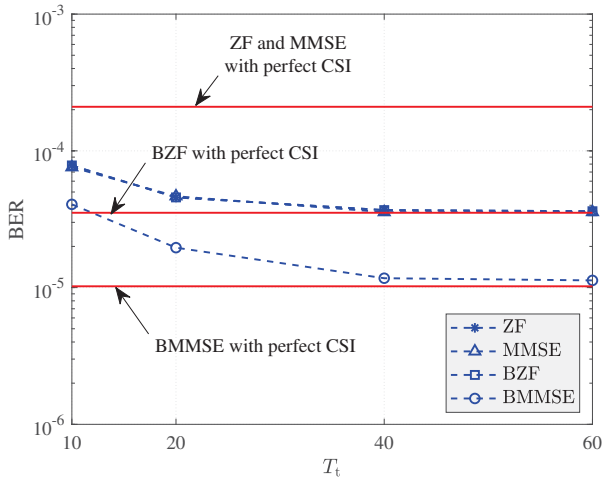
This indicates that the effective channel $\bar{\mathbf{a}}_i$ is a normalized version of the true channel. Note that the instantaneous magnitude of $\bar{\mathbf{h}}_i$ is not identifiable in 1-bit quantized MIMO systems [38], and consequently the SVM-based [28] and BMMSE [15] channel estimators provide estimates whose magnitudes are normalized. Therefore, when using a channel estimator such as [15], [28], ZF with estimated CSI will give the same performance as BZF with estimated CSI. ZF with estimated CSI outperforms ZF with perfect CSI since the channel estimate takes into account the inherent scaling ambiguity in the observed data. For the same reason, MMSE and BMMSE with estimated CSI also outperform MMSE with perfect CSI, but MMSE performs worse than BMMSE because MMSE still applies the noise covariance matrix $N_0\mathbf{I}$, while BMMSE uses the covariance matrix $\mathbf{\Sigma}_{\bar{n}}$ that includes information about the quantization noise.

For the first stage, we proposed the OBMNet, which is devised from a reformulated robust ML detection problem. In Fig. 8, we verify the robustness of the reformulated ML detection problem in (22) when implemented with estimated CSI. We carried out simulations using the BMMSE channel estimator [15] with different training lengths T_{t}. It can be seen from Fig. 8 that when the CSI is perfectly known, both the conventional and the proposed ML detection algorithms

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2021.3082844, IEEE Transactions on Wireless Communications



(a) BMMSE estimated CSI [15].



(b) SVM-based estimated CSI [28].

Fig. 7: BER comparison between ZF, MMSE, BZF, and BMMSE linear receivers with estimated CSI. The setting is $K = 2$ users, $N = 16$ receive antennas, QPSK signaling, and $\mathrm{SNR} = 30\mathrm{dB}$. T_{t} is the training length.

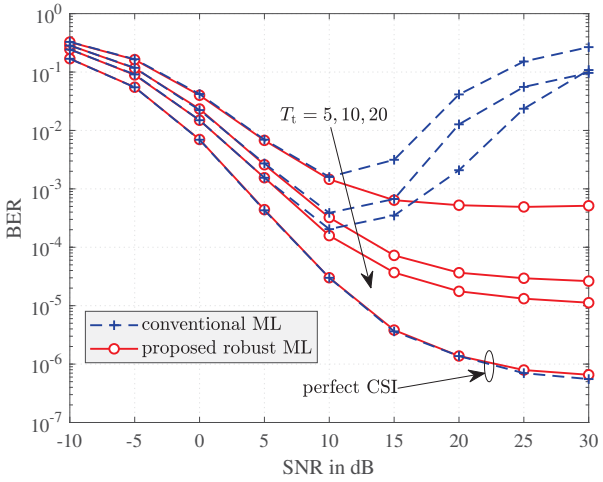


Fig. 8: Performance comparison between the conventional and the proposed ML detection problems with $K = 2$, $N = 16$, and QPSK signaling. The BMMSE channel estimator [15] is used with different training lengths T_{t}.

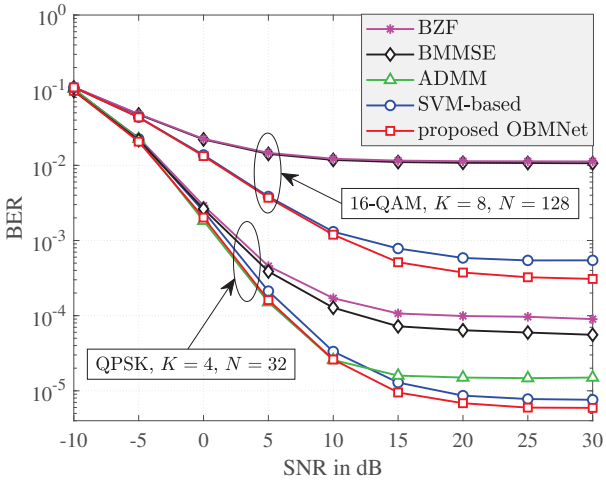
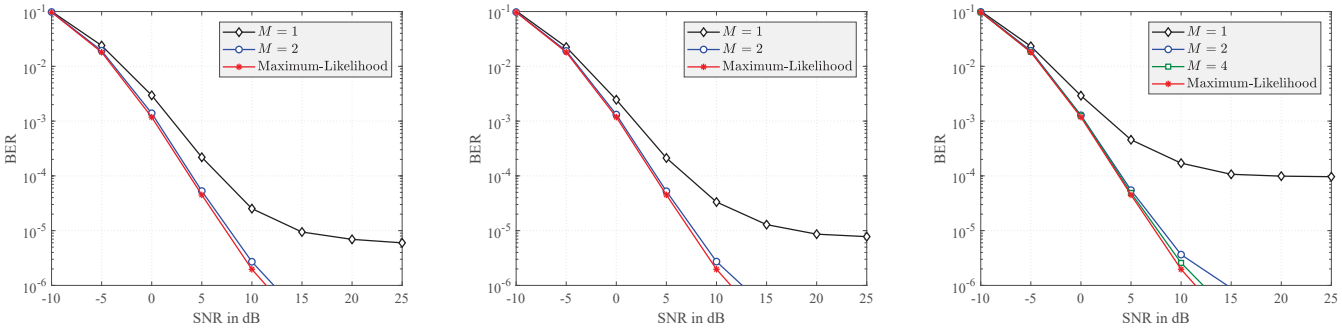


Fig. 9: First stage performance comparison between the proposed OBMNet detector and existing detection methods with perfect CSI.

yield almost identical performance. However, when the CSI is imperfectly known, the performance of conventional ML detection is significantly degraded at high SNR, while the proposed robust ML detection algorithm remains stable. This verifies our analysis in Section III.

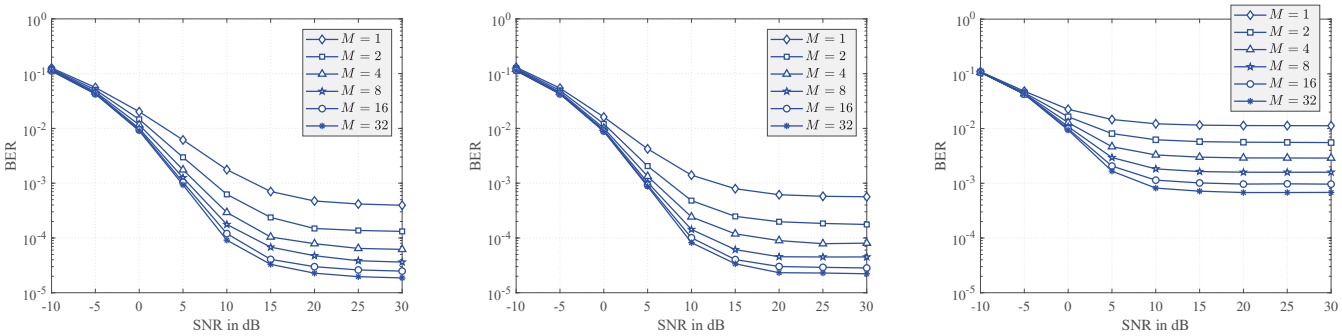
Fig. 9 provides a performance comparison between the proposed OBMNet and several existing receivers including BMMSE [32], BZF [32], ADMM [9], and SVM-based [28]. The performance of OSD is comparable to that of the SVM-based method but with much higher computational complexity. Since the SVM-based method also outperforms other prior methods, we use it as a comparative benchmark in this paper. To implement the SVM-based receiver, we use the Scikit-learn machine learning library [39]. For training OBMNet, we use TensorFlow [40] and the Adam optimizer [41] with a learning rate of 10^{-2}. The size of each training set is set to 1000. The input of the first layer $\mathbf{x}_0$ is set to a zero vector. For the case of QPSK, $K = 4$, and $N = 32$, OBMNet has 10 layers; and for the case of 16-QAM, $K = 8$, and $N = 128$, OBMNet has 15 layers. During the detection phase, the trained OBMNet is employed to perform batch detection. Note that batch detection is an advantage of DNN since it can take a batch of multiple symbol vectors as its input, which speeds up the detection process [21]. The effect of batch size on run time can be seen in Table II. The results in Fig. 9 show that the proposed OBMNet and the SVM-based method outperform the Bussgang-based linear receivers as well as the ADMM-based method. At high SNRs, the BER floor of the OBMNet detector is slightly lower than that of the SVM-based method. Note that the ADMM-based method is designed specifically for QPSK signaling, so Fig. 9 does not show a result for this method with 16-QAM.

To evaluate the computational complexity of the receivers



(a) Proposed OBMNet. (b) SVM-based [28]. (c) BZF [32].

Fig. 10: Second stage performance comparison between different receivers with $K = 4$, $N = 32$, QPSK signaling, and perfect CSI.



(a) Proposed OBMNet. (b) SVM-based [28]. (c) BZF [32].

Fig. 11: Second stage performance comparison between different receivers with $K = 8$, $N = 128$, 16-QAM signaling, and perfect CSI.

TABLE II: First stage average run time.

QPSK, $K = 4$, $N = 32$				
batch size	**proposed OBMNet**	**BZF [32]**	**BMMSE [32]**	**SVM-based [28]**
1	2.2×10^{-4}	1.3×10^{-5}	1.5×10^{-5}	$[3.1, 3.8]\times10^{-4}$
10	5.8×10^{-5}	1.1×10^{-5}	1.1×10^{-5}	$[3.1, 3.8]\times10^{-4}$
100	4.2×10^{-5}	1.0×10^{-5}	1.0×10^{-5}	$[3.1, 3.8]\times10^{-4}$
250	3.6×10^{-5}	1.0×10^{-5}	1.0×10^{-5}	$[3.1, 3.8]\times10^{-4}$
16-QAM, $K = 8$, $N = 128$				
batch size	**proposed OBMNet**	**BZF [32]**	**BMMSE [32]**	**SVM-based [28]**
1	5.2×10^{-4}	2.8×10^{-5}	3.5×10^{-5}	$[6.4, 9.6]\times10^{-4}$
5	3.1×10^{-4}	2.5×10^{-5}	3.3×10^{-5}	$[6.4, 9.6]\times10^{-4}$
10	2.8×10^{-4}	2.4×10^{-5}	3.2×10^{-5}	$[6.4, 9.6]\times10^{-4}$
25	2.6×10^{-4}	2.4×10^{-5}	3.2×10^{-5}	$[6.4, 9.6]\times10^{-4}$

used in Fig. 9, average run time is reported in Table II. Since the run time is largely affected by implementation details and the associated hardware/platform, to ensure fairness, we implemented all the receivers using the same simulation hardware with Python 3.7 and the Numpy package. Note that the run time of the SVM-based method depends on the SNR, and so we report the resulting range of run times. It can be seen from Table II that the Bussgang-based linear receivers have lower complexity than OBMNet and the SVM-based receiver. This is obvious since the linear receivers only require a matrix-vector multiplication for detecting each received signal. The run time of the BZF receiver is smaller than that of BMMSE because the combining matrix $\mathbf{W}_{\texttt{BZF}}$ only involves the inversion of a $K \times K$ matrix while $\mathbf{W}_{\texttt{BMMSE}}$ requires the inverse of an $N \times N$ matrix. OBMNet is more computationally expensive than the linear receivers but its complexity is still much less than that of the SVM-based method. It can also be seen that the run time of OBMNet can be significantly reduced by increasing the batch size. In situations where the added latency is not an issue, the use of batch detection is an advantage for DNN since it can speed up the detection process [21]. Note that the run time of the SVM-based method does not depend on the batch size since it processes different received signals separately and each time slot requires the SVM-based method to solve a new optimization problem.

For the second stage, performance comparisons are given in Fig. 10 for the case of QPSK with $K = 4$ and $N = 32$, and Fig. 11 for the case of 16-QAM with $K = 8$ and $N = 128$. We set $\gamma = \frac{1}{2\sqrt{2}}$ for QPSK and $\gamma = \frac{1}{2\sqrt{10}}$ for 16-QAM. Here, we compare the BZF, OBMNet, and SVM-based receivers and omit BMMSE since the performance of BZF and BMMSE are comparable, and the complexity of BZF is lower than that of BMMSE. The case of $M = 1$ is equivalent to the use of symbol-by-symbol detection in the first stage. In this case, OBMNet provides the best performance, i.e., it yields the best initial detection results. When increasing M, the proposed NN search method in the second stage significantly improves the performance compared to the first stage. In Fig. 10, the BERs obtained with a small M, e.g., $M = 2$, are already

TABLE III: Second stage average run time.

QPSK, $K = 4$, $N = 32$, batch size $= 250$			
M	**proposed OBMNet**	**SVM-based [28]**	**BZF [32]**
2	$[0.4, 1.0] \times 10^{-4}$	$[0.6, 1.2] \times 10^{-4}$	$[0.5, 1.0] \times 10^{-4}$
16-QAM, $K = 8$, $N = 128$, batch size $= 25$			
M	**proposed OBMNet**	**SVM-based [28]**	**BZF [32]**
2	$[1.6, 2.5] \times 10^{-4}$	$[1.9, 3.2] \times 10^{-4}$	$[2.0, 2.5] \times 10^{-4}$
4	$[1.8, 3.7] \times 10^{-4}$	$[2.1, 5.2] \times 10^{-4}$	$[2.8, 3.5] \times 10^{-4}$
8	$[2.0, 6.6] \times 10^{-4}$	$[2.4, 9.6] \times 10^{-4}$	$[3.9, 6.2] \times 10^{-4}$
16	$[2.3, 14.7] \times 10^{-4}$	$[3.3, 21.7] \times 10^{-4}$	$[5.4, 13.1] \times 10^{-4}$
32	$[3.0, 34.1] \times 10^{-4}$	$[4.3, 46.5] \times 10^{-4}$	$[8.1, 30.0] \times 10^{-4}$

close to the BER of the ML detection approach. The results in Fig. 11 clearly show that the performance can be improved by increasing M, but this requires more computation resources as seen in Table III. Thus, one should choose M to balance the detection accuracy and computational complexity. It should be noted that $|\mathcal{A}|$ is always a power of two, but M can be any positive integer number.

VI. Conclusion

In this paper, we have summarized the literature of linear receivers for one-bit massive MIMO systems and proposed a two-stage detection method for massive MIMO systems with one-bit ADCs. In particular, for the first stage, we proposed a novel model-driven OBMNet detector, which is constructed based on a reformulated robust ML detection problem. The layered structure of OBMNet is simple, unique, and adaptive to the CSI and received signals. This OBMNet detector outperforms existing approaches and also has low complexity. For the second stage, an NN search method was proposed to further improve the performance of the first stage. This NN search method allows one to limit the search complexity as desired.

Appendix A
Proof of Proposition 1

Since $\mathbf{x}_m$ is the m^{th} nearest symbol vector, we have the following condition:

$$\|\mathbf{x}_1 - \tilde{\mathbf{x}}\|^2 < \ldots < \|\mathbf{x}_{m-1} - \tilde{\mathbf{x}}\|^2 < \|\mathbf{x}_m - \tilde{\mathbf{x}}\|^2 < \|\mathbf{x} - \tilde{\mathbf{x}}\|^2 \tag{35}$$

for any $\mathbf{x} \notin \mathcal{X}_m$.

We prove the proposition by contradiction. Suppose that $\mathbf{x}_m$ is not a neighbor of $\mathcal{X}_{m-1}$, i.e., $\mathbf{x}_m \notin \mathcal{N}(\mathcal{X}_{m-1})$ or $d_{\min}(\mathbf{x}_m, \mathcal{X}_{m-1}) > 1$. For the sake of simplicity, we consider the case where $d_{\min}(\mathbf{x}_m, \mathcal{X}_{m-1}) = 2$. Proof for the other cases where $d_{\min}(\mathbf{x}_m, \mathcal{X}_{m-1}) > 2$ can be accomplished similarly.

Let $\mathbf{x}_p \in \mathcal{X}_{m-1}$ with $p \in \{1, 2, \ldots, m-1\}$ be a symbol vector such that $d(\mathbf{x}_p, \mathbf{x}_m) = 2$. Without loss of generality, we can always assume that the two position indices at which the differences occur are 1 and 2, i.e.,

$$\begin{cases} x_{m,1} \neq x_{p,1} \\ x_{m,2} \neq x_{p,2} \\ x_{m,i} = x_{p,i} \ \forall i \in \{3, \ldots, 2K\}. \end{cases} \tag{36}$$

Now, we consider two other symbol vectors $\mathbf{x}' = [x'_1, \ldots, x'_{2K}]^T$ and $\mathbf{x}'' = [x''_1, \ldots, x''_{2K}]^T$ such that

$$\begin{cases} x'_1 = x_{m,1} \neq x_{p,1} = x''_1 \\ x'_2 = x_{p,2} \neq x_{m,2} = x''_2 \\ x'_i = x''_i = x_{p,i} = x_{m,i} \ \forall i \in \{3, \ldots, 2K\}. \end{cases} \tag{37}$$

Hence, $\mathbf{x}'$ and $\mathbf{x}''$ are the two symbol vectors satisfying $d(\mathbf{x}', \mathbf{x}_m) = d(\mathbf{x}'', \mathbf{x}_m) = 1$. In other words, both $\mathbf{x}'$ and $\mathbf{x}''$ are neighbors of $\mathbf{x}_m$.

If $\mathbf{x}' \in \mathcal{X}_{m-1}$ and/or $\mathbf{x}'' \in \mathcal{X}_{m-1}$, then $d_{\min}(\mathbf{x}_m, \mathcal{X}_{m-1}) = 1$ because $\mathbf{x}_m$ is a neighbor of both $\mathbf{x}'$ and $\mathbf{x}''$, which is contradicted by the assumption that $d_{\min}(\mathbf{x}_m, \mathcal{X}_{m-1}) = 2$. Thus, $\mathbf{x}_m$ is a neighbor of $\mathcal{X}_{m-1}$, i.e, $\mathbf{x}_m \in \mathcal{N}(\mathcal{X}_{m-1})$.

If $\mathbf{x}' \notin \mathcal{X}_{m-1}$ and $\mathbf{x}'' \notin \mathcal{X}_{m-1}$, we have

$$|x_{m,1} - \tilde{x}_1|^2 = |x'_1 - \tilde{x}_1|^2 > |x_{p,1} - \tilde{x}_1|^2. \tag{38}$$

Adding both sides of (38) with $|x_{m,2} - \tilde{x}_2|^2$ yields

$$|x_{m,1} - \tilde{x}_1|^2 + |x_{m,2} - \tilde{x}_2|^2 > |x_{p,1} - \tilde{x}_1|^2 + |x_{m,2} - \tilde{x}_2|^2,$$

which can be rewritten as

$$|x_{m,1} - \tilde{x}_1|^2 + |x_{m,2} - \tilde{x}_2|^2 > |x''_1 - \tilde{x}_1|^2 + |x''_2 - \tilde{x}_2|^2 \tag{39}$$

because $x_{p,1} = x''_1$ and $x_{m,2} = x''_2$. The inequality in (39) indicates that $\|\mathbf{x}_m - \tilde{\mathbf{x}}\|^2 > \|\mathbf{x}'' - \tilde{\mathbf{x}}\|^2$, which means $\mathbf{x}''$ is closer to $\tilde{\mathbf{x}}$ than $\mathbf{x}_m$, or in other words, $\mathbf{x}_m$ is not the m^{th} nearest symbol vector of $\tilde{\mathbf{x}}$. This is contradicted by (35).

References

[1] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.

[2] J. Hoydis, S. ten Brink, and M. Debbah, "Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?" *IEEE J. Select. Areas in Commun.*, vol. 31, no. 2, pp. 160–171, Feb. 2013.

[3] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[4] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Select. Areas in Commun.*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[5] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE J. Select. Areas in Commun.*, vol. 17, no. 4, pp. 539–550, Apr. 1999.

[6] J. Choi, J. Mo, and R. W. Heath, "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.

[7] Y. Jeon, N. Lee, S. Hong, and R. W. Heath, "One-bit sphere decoding for uplink massive MIMO systems with one-bit ADCs," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4509–4521, July 2018.

[8] C. K. Wen, C. J. Wang, S. Jin, K. K. Wong, and P. Ting, "Bayes-optimal joint channel-and-data estimation for massive MIMO with low-precision ADCs," *IEEE Trans. Signal Process.*, vol. 64, no. 10, pp. 2541–2556, May 2016.

[9] ÃČ. T. Demir and E. Björnson, "ADMM-based one-bit quantized signal detection for massive MIMO systems with hardware impairments," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, Barcelona, Spain, May 2020, pp. 9120–9124.

[10] S. H. Mirfarshbafan, M. Shabany, S. A. Nezamalhosseini, and C. Studer, "Algorithm and VLSI design for 1-bit data detection in massive MIMO-OFDM," *IEEE Open J. Circuits and Systems*, vol. 1, pp. 170–184, Oct. 2020.

[11] Y. Jeon, N. Lee, and H. V. Poor, "Robust data detection for MIMO systems with one-bit ADCs: A reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1663–1676, Mar. 2020.

[12] S. H. Song, S. Lim, G. Kwon, and H. Park, "CRC-aided soft-output detection for uplink multi-user MIMO systems with one-bit ADCs," in *Proc. IEEE Wireless Commun. and Networking Conf.*, Marrakesh, Morocco, Apr. 2019, pp. 1–5.

[13] Y. Cho and S. Hong, "One-bit Successive-cancellation Soft-output (OSS) detector for uplink MU-MIMO systems with one-bit ADCs," *IEEE Access*, vol. 7, pp. 27172–27182, Feb. 2019.

[14] Z. Shao, R. C. de Lamare, and L. T. N. Landau, "Iterative detection and decoding for large-scale multiple-antenna systems with 1-bit ADCs," *IEEE Wireless Commun. Letters*, vol. 7, no. 3, pp. 476–479, June 2018.

[15] 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.

[16] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, "Throughput analysis of massive MIMO uplink with low-resolution ADCs," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 4038–4051, June 2017.

[17] K. Liu, C. Tao, L. Liu, T. Zhou, and Y. Liu, "Asymptotic analysis for low-resolution massive MIMO systems with MMSE receiver," *China Commun.*, vol. 15, no. 9, pp. 189–199, Sep. 2018.

[18] A. Mezghani, M.-S. Khoufi, and J. A. Nossek, "A modified MMSE receiver for quantized MIMO systems," *Proc. ITG/IEEE WSA, Vienna, Austria*, 2007.

[19] L. V. Nguyen and D. H. N. Nguyen, "Linear receivers for massive MIMO systems with one-bit ADCs," *CoRR*, 2019. [Online]. Available: https://arXiv.org/abs/1907.06664.

[20] J. J. Bussgang, "Crosscorrelation functions of amplitude-distorted gaussian signals," MIT Research Lab. Electronics, Tech. Rep. 216, 1952.

[21] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, May 2019.

[22] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5635–5648, Aug. 2020.

[23] N. T. Nguyen and K. Lee, "Deep learning-aided Tabu search detection for large MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4262–4275, June 2020.

[24] G. Gao, C. Dong, and K. Niu, "Sparsely connected neural network for massive MIMO detection," in *Proc. IEEE Int. Conf. Computer and Commun.*, Chengdu, China, Dec. 2018, pp. 397–402.

[25] L. V. Nguyen, D. T. Ngo, N. H. Tran, A. L. Swindlehurst, and D. H. N. Nguyen, "Supervised and semi-supervised learning for MIMO blind detection with low-resolution ADCs," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2427–2442, Apr. 2019.

[26] Y. Jeon, S. Hong, and N. Lee, "Supervised-learning-aided communication framework for MIMO systems with low-resolution ADCs," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7299–7313, Aug. 2018.

[27] S. Kim, M. So, N. Lee, and S. Hong, "Semi-supervised learning detector for MU-MIMO systems with one-bit ADCs," in *Proc. IEEE Int. Conf. Commun. Workshops*, Shanghai, China, May 2019, pp. 1–6.

[28] L. V. Nguyen, A. L. Swindlehurst, and D. H. N. Nguyen, "SVM-based channel estimation and data detection for one-bit massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2086–2099, 2021.

[29] A. K. Fletcher, S. Rangan, V. K. Goyal, and K. Ramchandran, "Robust predictive quantization: Analysis and design via convex optimization," *IEEE J. Selec. Topics in Signal Process.*, vol. 1, no. 4, pp. 618–632, Dec. 2007.

[30] O. Orhan, E. Erkip, and S. Rangan, "Low power analog-to-digital conversion in millimeter wave systems: Impact of resolution and bandwidth on performance," in *Proc. Inform. Theory and Applications Workshop*, San Diego, CA, USA, Feb. 2015, pp. 191–198.

[31] A. Mezghani and J. A. Nossek, "Capacity lower bound of MIMO channels with output quantization and correlated noise," in *Proc. IEEE Int. Symp. Inform. Theory*, Cambridge, Massachusetts, USA, July 2012, pp. 1–5.

[32] A. S. Lan, M. Chiang, and C. Studer, "Linearized binary regression," in *Proc. Annual Conf. on Inform. Sciences and Systems*, Princeton, NJ, USA, Mar. 2018, pp. 1–6.

[33] S. R. Bowling, M. T. Khasawneh, S. Kaewkuekool, and B. R. Cho, "A logistic approximation to the cumulative normal distribution," *Journal of Industrial Engineering and Management*, vol. 2, no. 1, pp. 114–127, Mar. 2009.

[34] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *CoRR*, 2014. [Online]. Available: https://arXiv:1409.2574.

[35] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Microsoft Research, Tech. Rep. MSR-TR-98-14, 1999.

[36] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods - Support Vector Learning*, B. Scholkopf and A. Smola, Eds. MIT Press, 1998, pp. 44–56.

[37] C. W. Hsu and C. J. Lin, "A simple decomposition method for support vector machines," *Machine Learning*, vol. 46, pp. 291–314, 2002.

[38] S. Rao, A. Mezghani, and A. L. Swindlehurst, "Channel estimation in one-bit massive MIMO systems: Angular versus unstructured models," *IEEE J. Select. Topics in Signal Process.*, vol. 13, no. 5, pp. 1017–1031, Sep. 2019.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Oct. 2011.

[40] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May. 2015, pp. 1–41.

Ly V. Nguyen (Student Member, IEEE) received the B.Eng. degree in Electronics and Telecommunications from the University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam, in 2014, and the M.Sc. degree in Advanced Wireless Communications Systems from Centrale-Supélec, Paris-Saclay University, France, in 2016. He is currently pursuing the Ph.D. degree in a joint doctoral program in Computational Science with San Diego State University and University of California, Irvine, CA, USA. He received a Best Paper Award from the 2020 IEEE International Conference on Communications (ICC). His research interests include wireless communications, signal processing, and machine learning.

A. Lee Swindlehurst (Fellow, IEEE) received the B.S. (1985) and M.S. (1986) degrees in Electrical Engineering from Brigham Young University (BYU), and the PhD (1991) degree in Electrical Engineering from Stanford University. He was with the Department of Electrical and Computer Engineering at BYU from 1990-2007, where he served as Department Chair from 2003-06. During 1996-97, he held a joint appointment as a visiting scholar at Uppsala University and the Royal Institute of Technology in Sweden. From 2006-07, he was on leave working as Vice President of Research for ArrayComm LLC in San Jose, California. Since 2007 he has been a Professor in the Electrical Engineering and Computer Science Department at the University of California Irvine, where he served as Associate Dean for Research and Graduate Studies in the Samueli School of Engineering from 2013-16. During 2014-17 he was also a Hans Fischer Senior Fellow in the Institute for Advanced Studies at the Technical University of Munich. In 2016, he was elected as a Foreign Member of the Royal Swedish Academy of Engineering Sciences (IVA). His research focuses on array signal processing for radar, wireless communications, and biomedical applications, and he has over 300 publications in these areas. Dr. Swindlehurst is a Fellow of the IEEE and was the inaugural Editor-in-Chief of the IEEE Journal of Selected Topics in Signal Processing. He received the 2000 IEEE W. R. G. Baker Prize Paper Award, the 2006 IEEE Communications Society Stephen O. Rice Prize in the Field of Communication Theory, the 2006 and 2010 IEEE Signal Processing Society's Best Paper Awards, and the 2017 IEEE Signal Processing Society Donald G. Fink Overview Paper Award.

Duy H. N. Nguyen (Senior Member, IEEE) received the B.Eng. degree (Hons.) from the Swinburne University of Technology, Hawthorn, VIC, Australia, in 2005, the M.Sc. degree from the University of Saskatchewan, Saskatoon, SK, Canada, in 2009, and the Ph.D. degree from McGill University, Montréal, QC, Canada, in 2013, all in electrical engineering. From 2013 to 2015, he held a joint appointment as a Research Associate with McGill University and a Post-doctoral Research Fellow with the Institut National de la Recherche Scientifique, Université du Québec, Montréal, QC, Canada. He was a Research Assistant with the University of Houston, Houston, TX, USA, in 2015, and a Post-doctoral Research Fellow with the University of Texas at Austin, Austin, TX, USA, in 2016. Since 2016, he has been an Assistant Professor with the Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA, USA. His current research interests include resource allocation in wireless networks, signal processing for communications, convex optimization, game theory, and machine learning. Dr. Nguyen has been serving as a TPC member for a number of flagship IEEE conferences, including ICC, GLOBECOM, and INFOCOM. He was a recipient of the Australian Development Scholarship, the FRQNT Doctoral Fellowship and Post-doctoral Fellowship, and the NSERC Post-doctoral Fellowship. He received a Best Paper Award at the 2020 IEEE International Conference on Communications (ICC). He is currently an Associate Editor for the *EURASIP Journal on Wireless Communications and Networking.*