Real-Valued Sparse Bayesian Learning Approach for Massive MIMO Channel Estimation

Lei Zhou, Zheng Cao, and Jisheng Dai, *Member, IEEE*

***Abstract*—This letter describes a real-valued sparse Bayesian learning (SBL) approach for massive multiple-input multipleoutput (MIMO) downlink channel estimation. The main idea of the approach is to introduce a certain unitary transformation into pilots, so as to convert complex-valued channel recovery problems into real ones. Due to exploiting the real-valued structure of the data matrices, the new approach brings a significant decrease in computational complexity, as well as a good noise suppression. Simulation results demonstrate that the new method can reduce the computation load and improve the channel estimation performance simultaneously.**

***Index Terms*—Channel estimation, multiple-input multiple-output (MIMO), real-valued transformation, sparse Bayesian learning (SBL), off-grid refinement.**

I. Introduction

MASSIVE multiple-input multiple-output (MIMO) has become a key enabling technology for next generation wireless communication systems. To reap the benefit of excessive base station (BS) antennas, it is essential to acquire channel state information at the transmitter (CSIT). The CSIT in time division duplex (TDD) systems can be obtained by leveraging the channel reciprocity, but the reciprocity does not hold in frequency division duplex (FDD) systems. Therefore, the FDD downlink channel estimation seems to be an extremely challenging task, as the training and feedback overhead is proportional to the number of antennas that can be quite large in massive MIMO systems [1].

Many works have shown that the effective dimension of a massive MIMO channel is actually much smaller than the original dimension due to the limited local scattering effect in the propagation environment [2], [3]. Particularly, when a uniform linear array (ULA) is located at the BS, the massive MIMO channel can be sparsely represented under the discrete Fourier transform (DFT) basis [3]. In recent years, many downlink channel estimation methods have been proposed by exploiting such hidden sparsity [2], [3]. But, the DFT-based channel estimation methods often have a performance loss owing to the leakage of energy caused by direction mismatch. In order to solve this problem, Dai *et al.* provided an off-grid sparse Bayesian learning (SBL) method for the downlink channel estimation [4]. Since the off-grid SBL method requires to calculate inverse of a large complex matrix in each iteration, its computational complexity can be very high in practical implementations. The additional grid refining procedure in [4] will further increase the computation load. Hence, it motivates us to look for an efficient way to reduce the computational complexity.

We notice that a considerable amount of computations can be saved if we transform the complex-valued problem into a real one [5], [6]. Therefore, we try to propose an efficient real-valued off-grid SBL approach for the massive MIMO downlink channel estimation in this letter. The main novelty of our approach is to introduce a certain unitary transformation into pilots, so as to convert complex-valued channel recovery problems into real ones. Note that performing the real-valued transformation with a certain unitary transformation is not new, which has been widely used for real-valued DOA estimation problems [5], [6]. However, to the best of our knowledge, no real-valved transformation method has been reported (or even noticed) for the massive MIMO channel estimation in the literature, because it could be impossible to apply the real-valued transformation to the massive MIMO channel estimation directly in the absence of the aforementioned design for pilots. Due to exploiting the real-valued structure of the data matrices, our new method can bring a significant computational load reduction, as well as a good noise suppression. Simulation results reveal that the proposed method can simultaneously achieve lower complexity and better channel estimation performance than the state-of-the-art methods.

II. Data Model

Consider a massive MIMO system supporting several mobile users (MUs) with N_r antennas, and there is only one BS with $N_t(\gg N_r)$ antennas. The downlink channel matrix from the BS to a MU can be represented as [7], [8]:

$$\mathbf{H}=\sum_{c=1}^{N_c}\sum_{s=1}^{N_s}\xi_{c,s}\mathbf{a}(\theta_{c,s})\mathbf{b}^H(\vartheta_{c,s}), \tag{1}$$

where N_c denotes the number of scattering clusters, N_s denotes the number of sub-paths per scattering cluster, $\xi_{c,s}$, $\theta_{c,s}$ and $\vartheta_{c,s}$ denote the path gain, angles-of-departure (AoDs) and angles-of-arrival (AoAs), respectively. Note that $\mathbf{a}(\theta_{c,s})$ and $\mathbf{b}(\vartheta_{c,s})$ are the array response vectors of the BS and MU, respectively, whose expressions will be given later. For ease of notation, we define $\{\theta_l\}_{l=1}^{L}$ with $\theta_{(c-1)N_s+s}=\theta_{c,s}$ and $L=N_cN_s$, so are $\{\vartheta_l\}_{l=1}^{L}$ and $\{\xi_l\}_{l=1}^{L}$. Then, $\mathbf{H}$ can be represented as:

$$\mathbf{H}=\mathbf{A}\mathbf{\Lambda}\mathbf{B}^H, \tag{2}$$

where $\mathbf{A}=[\mathbf{a}(\theta_1),\mathbf{a}(\theta_2),\ldots,\mathbf{a}(\theta_L)]\in\mathbb{C}^{N_t\times L}$, $\mathbf{B}=[\mathbf{b}(\vartheta_1),\mathbf{b}(\vartheta_2),\ldots,\mathbf{b}(\vartheta_L)]\in\mathbb{C}^{N_r\times L}$, and $\mathbf{\Lambda}=\mathrm{diag}(\boldsymbol{\xi})$ with $\mathrm{diag}(\cdot)$ denoting diagonal operator. Let the BS broadcast a sequence of T training pilot symbols, denoted by $\mathbf{X}\in\mathbb{C}^{T\times N_t}$,

Manuscript received September 18, 2019; accepted November 9, 2019. Date of publication November 13, 2019; date of current version March 9, 2020. This work was supported by the National Natural Science Foundation of China under Project 61571211. The associate editor coordinating the review of this article and approving it for publication was W. Zhang. *(Corresponding author: Jisheng Dai.)*

The authors are with the Department of Electronic Engineering, Jiangsu University, Zhenjiang 212013, China (e-mail: zhoulei_zl@foxmail.com; 1520889492@qq.com; jsdai@ujs.edu.cn).

Digital Object Identifier 10.1109/LWC.2019.2953265

for each MU to estimate the downlink channel. For each MU, the downlink received signal $\mathbf{Y} \in \mathbb{C}^{T \times N_r}$ can be written as

$$\mathbf{Y} = \mathbf{X}\mathbf{H} + \mathbf{E}, \tag{3}$$

where $\mathbf{E} \in \mathbb{C}^{T \times N_r}$ stands for the additive complex Gaussian noise with each element being zero mean and variance σ^2.

Due to the limited local scattering effect at the BS side [2], [3], the number of N_c is usually small and the subpaths associated with each scattering cluster are likely to concentrate in a small range. Hence, only a few angles are occupied in the angular domain, which brings a sparse representation. Define $\hat{\boldsymbol{\theta}} = \{\hat{\theta}_l\}_{l=1}^{\hat{L}}$ as a fixed sampling grid that uniformly covers the angular domain $[-\frac{\pi}{2}, \frac{\pi}{2}]$, where $\hat{L}$ denotes the number of grid points. If the grid is fine enough, we can rewrite (3) as:

$$\mathbf{Y} = \mathbf{X}\hat{\mathbf{A}}\mathbf{S} + \mathbf{E}, \tag{4}$$

where $\hat{\mathbf{A}} = [\mathbf{a}(\hat{\theta}_1), \mathbf{a}(\hat{\theta}_2), \ldots, \mathbf{a}(\hat{\theta}_{\hat{L}})] \in \mathbb{C}^{N_t \times \hat{L}}$ and $\mathbf{S} \in \mathbb{C}^{\hat{L} \times N_r}$ is a row sparse matrix whose non-zero rows correspond to the true directions. For example, if the *m*-th row of $\mathbf{S}$ (denoted by $\mathbf{S}_{m,:}$) is nonzero and the corresponding true AoD is θ_n, $n \in \{1, 2, \ldots, L\}$, then we have $\hat{\theta}_m = \theta_n$ and $\mathbf{S}_{m,:} = \xi_n \mathbf{b}^H(\vartheta_n)$.

As mentioned in [4], it is usually invalid to assume that the real direction is on a predefined spatial grid in practical applications. In order to handle the direction mismatch, we adopt the off-grid model proposed in [4]. Specifically, if $\theta_l \notin \{\hat{\theta}_i\}_{i=1}^{\hat{L}}$ and $\hat{\theta}_{n_l}, n_l \in \{1, 2, \ldots, \hat{L}\}$, is the nearest grid point to θ_l, where $\theta_l = \hat{\theta}_{n_l} + \varphi_{n_l}$, φ_{n_l} stands for the off-grid gap, and

$$\varphi_{n_l} = \begin{cases} \theta_l - \hat{\theta}_{n_l}, & l = 1, 2, \ldots, L, \\ 0, & \text{otherwise}. \end{cases}$$

As a consequence, we have

$$\mathbf{Y} = \mathbf{X}\hat{\mathbf{A}}(\boldsymbol{\varphi})\mathbf{S} + \mathbf{E}, \tag{5}$$

where $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, \ldots, \varphi_{\hat{L}}]^T$, $\hat{\mathbf{A}}(\boldsymbol{\varphi}) = [\mathbf{a}(\hat{\theta}_1 + \varphi_1),\ \mathbf{a}(\hat{\theta}_2 + \varphi_2), \ldots, \mathbf{a}(\hat{\theta}_{\hat{L}} + \varphi_{\hat{L}})]$. With (5), an off-grid SBL-based method proposed in [4] can be applied to recover the sparse representation matrix $\mathbf{S}$. Note that the off-grid SBL method [4] requires to calculate inverse of a large complex matrix per iteration, which will result in a high computational load. Moreover, additional grid refining procedure in [4] will further increase the computational load. From [5], we know that a complex multiplication costs four times as much as that of a real multiplication, therefore a considerable amount of computations can be saved if we transform the complex-valued problem (5) into a real-valued one.

III. The Proposed Real-Valued Method

A. Real-Valued Transformation

Without loss of generality, let N_t be even. If the geometrical center of the ULA is chosen as the reference point of steering vectors, $\mathbf{a}(\theta)$ is in the form of [5], [6]

$$\mathbf{a}(\theta) = \Big[e^{-j\frac{(N_t-1)\phi(\theta)}{2}}, e^{-j\frac{(N_t-3)\phi(\theta)}{2}}, \ldots, e^{-j\frac{\phi(\theta)}{2}}, e^{j\frac{\phi(\theta)}{2}}, \ldots, e^{j\frac{(N_t-1)\phi(\theta)}{2}}\Big]^T, \tag{6}$$

where $\phi(\theta) = (-2\pi d/\lambda)\sin(\theta)$, λ is the wavelength of the downlink propagation, and d stands for the distance between adjacent sensors. To convert the complex-valued manifold $\mathbf{a}(\theta)$ into a real one, we define the transformation matrix $\mathbf{Q}_{N_t}$ as:

$$\mathbf{Q}_{N_t} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_{\frac{N_t}{2}} & \mathbf{J}_{\frac{N_t}{2}} \\ j\mathbf{J}_{\frac{N_t}{2}} & -j\mathbf{I}_{\frac{N_t}{2}} \end{bmatrix}, \tag{7}$$

where $\mathbf{I}_{\frac{N_t}{2}}$ denotes $\frac{N_t}{2} \times \frac{N_t}{2}$ identity matrix and $\mathbf{J}_{\frac{N_t}{2}}$ denotes an $\frac{N_t}{2} \times \frac{N_t}{2}$ exchange matrix with all ones on its anti-diagonal positions and zeros elsewhere. It is easy to check that $\underline{\mathbf{a}}(\theta) \triangleq \mathbf{Q}_{N_t}\mathbf{a}(\theta)$ is real-valued, which can be rewritten as [5], [6]:

$$\underline{\mathbf{a}}(\theta) = \sqrt{2}\Big[\cos(\frac{N_t - 1}{2}\phi(\theta)), \cos(\frac{N_t - 3}{2}\phi(\theta)), \ldots, \cos(\frac{1}{2}\phi(\theta)), \sin(\frac{1}{2}\phi(\theta)), \ldots, \sin(\frac{N_t - 1}{2}\phi(\theta))\Big]^T. \tag{8}$$

Note that this real-valued transformation applies ULAs only.

To formulate (5) as a real-valued sparse representation problem, we introduce $\mathbf{Q}_{N_t}$ into the pilot matrix, i.e., $\mathbf{X} = \mathbf{G}\mathbf{Q}_{N_t}$, where the elements of $\mathbf{G} \in \mathbb{R}^{T \times N_t}$ can be randomly chosen with an i.i.d. real Gaussian distribution. Substituting $\mathbf{X} = \mathbf{G}\mathbf{Q}_{N_t}$ into (5) leads to

$$\mathbf{Y} = \boldsymbol{\Phi}(\boldsymbol{\varphi})\mathbf{S} + \mathbf{E}, \tag{9}$$

where $\boldsymbol{\Phi}(\boldsymbol{\varphi}) = \mathbf{G}\underline{\mathbf{A}}(\boldsymbol{\varphi}) \in \mathbb{R}^{T \times \hat{L}}$ and $\underline{\mathbf{A}}(\boldsymbol{\varphi}) = \mathbf{Q}_{N_t}\hat{\mathbf{A}}(\boldsymbol{\varphi}) = [\underline{\mathbf{a}}(\hat{\theta}_1 + \varphi_1), \underline{\mathbf{a}}(\hat{\theta}_2 + \varphi_2), \ldots, \underline{\mathbf{a}}(\hat{\theta}_{\hat{L}} + \varphi_{\hat{L}})] \in \mathbb{R}^{N_t \times \hat{L}}$. If we define $\hat{\mathbf{Y}} = [\mathrm{Re}(\mathbf{Y}), \mathrm{Im}(\mathbf{Y})]$ with $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ standing for the real part and imaginary part, respectively, we can obtain a real-valued sparse representation problem:

$$\hat{\mathbf{Y}} = \boldsymbol{\Phi}(\boldsymbol{\varphi})\hat{\mathbf{S}} + \hat{\mathbf{E}}, \tag{10}$$

where $\hat{\mathbf{S}} = [\mathrm{Re}(\mathbf{S}), \mathrm{Im}(\mathbf{S})]$ and $\hat{\mathbf{E}} = [\mathrm{Re}(\mathbf{E}), \mathrm{Im}(\mathbf{E})]$. Let the SVD of $\hat{\mathbf{Y}}$ be

$$\hat{\mathbf{Y}} = \mathbf{U}_s\boldsymbol{\Sigma}_s\mathbf{V}_s^T + \mathbf{U}_n\boldsymbol{\Sigma}_n\mathbf{V}_n^T, \tag{11}$$

where $\boldsymbol{\Sigma}_s$ and $\boldsymbol{\Sigma}_n$ contain *M* significant and $\min\{T, 2N_r\} - M$ non-significant singular values, respectively. Here we use $\eta\%$ (e.g., $\eta = 80$) of the max singular value as the threshold to determine whether the singular value is significant or not. In order to retain useful information while reduce computational load, we define a reduced dimensional matrix $\underline{\mathbf{Y}}$ of size $T \times M$:

$$\underline{\mathbf{Y}} = \hat{\mathbf{Y}}\mathbf{V}_s = \boldsymbol{\Phi}(\boldsymbol{\varphi})\underline{\mathbf{S}} + \underline{\mathbf{E}}, \tag{12}$$

where $\underline{\mathbf{S}} = \hat{\mathbf{S}}\mathbf{V}_s$ and $\underline{\mathbf{E}} = \hat{\mathbf{E}}\mathbf{V}_s$. Recall that the element of the matrix $\hat{\mathbf{E}}$ in (10) is i.i.d. real Gaussian with a common variance $\sigma^2/2$. According to the orthogonal invariance property of Gaussian random matrix and the definition of $\underline{\mathbf{E}}$, each element of $\underline{\mathbf{E}}$ is approximately i.i.d real Gaussian with a common variance $\sigma^2/2$. The reason why this holds only approximately is that $\mathbf{V}_s$ may depend on $\hat{\mathbf{E}}$. However, for moderate to high signal-to-noise ratio (SNR), the term $\boldsymbol{\Phi}(\boldsymbol{\varphi})\hat{\mathbf{S}}$ will dominate $\mathbf{V}_s$ and

$$\mathrm{span}(\mathbf{V}_s) = \mathrm{span}\Big(\hat{\mathbf{S}}^T\boldsymbol{\Phi}^T(\boldsymbol{\varphi})\Big), \tag{13}$$

where $\mathrm{span}(\cdot)$ denotes the subspace spanned by columns of a matrix. Then, we have the following claim.

Claim 1: The real-valued formulation (12) can bring a noise suppression effect.

Proof: According to (13), we have

$$\boldsymbol{\Phi}(\boldsymbol{\varphi})\hat{\mathbf{S}}\mathbf{V}_s\mathbf{V}_s^T = \boldsymbol{\Phi}(\boldsymbol{\varphi})\hat{\mathbf{S}}. \tag{14}$$

Then, the SNR in (12) can be calculated as

$$\frac{\mathrm{tr}(\boldsymbol{\Phi}(\boldsymbol{\varphi})\underline{\mathbf{S}}\underline{\mathbf{S}}^H\boldsymbol{\Phi}(\boldsymbol{\varphi})^H)}{\mathrm{tr}(\underline{\mathbf{E}}\underline{\mathbf{E}}^H)} = \frac{\mathrm{tr}(\boldsymbol{\Phi}(\boldsymbol{\varphi})\hat{\mathbf{S}}\hat{\mathbf{S}}^H\boldsymbol{\Phi}(\boldsymbol{\varphi})^H)}{\mathrm{tr}(\underline{\mathbf{E}}\underline{\mathbf{E}}^H)} \tag{15}$$

$$= \frac{\mathrm{tr}(\boldsymbol{\Phi}(\boldsymbol{\varphi})\hat{\mathbf{S}}\hat{\mathbf{S}}^H\boldsymbol{\Phi}(\boldsymbol{\varphi})^H)}{TM\sigma^2/2}, \tag{16}$$

where (15) follows (14) directly. The SNR in (5) is equal to the one in (10), which can be calculated as

$$\frac{\mathrm{tr}(\boldsymbol{\Phi}(\boldsymbol{\varphi})\hat{\mathbf{S}}\hat{\mathbf{S}}^H\boldsymbol{\Phi}(\boldsymbol{\varphi})^H)}{\mathrm{tr}(\hat{\mathbf{E}}\hat{\mathbf{E}}^H)} = \frac{\mathrm{tr}(\boldsymbol{\Phi}(\boldsymbol{\varphi})\hat{\mathbf{S}}\hat{\mathbf{S}}^H\boldsymbol{\Phi}(\boldsymbol{\varphi})^H)}{TN_r\sigma^2}. \tag{17}$$

Since $M \leq \min\{T, 2N_r\}$, (16) will give a larger value than (17). Therefore, (12) can bring a better noise suppression. Note that (16) could give the maximum value if $M = 1$. However, this case would not happen in practice unless for a degenerated channel. ■

B. Sparse Bayesian Learning Formulation and Inference

Following the commonly used SBL model [9], we assign a non-stationary Gaussian prior distribution with a distinct precision (e.g., δ_l) for each row of $\underline{\mathbf{S}}$. Let $\boldsymbol{\delta} = [\delta_1, \delta_2, \ldots, \delta_{\hat{L}}]^T$, and then we have

$$p(\underline{\mathbf{S}}|\boldsymbol{\delta}) = \prod_{m=1}^{M} \mathcal{N}\Big(\underline{\mathbf{s}}_m|\mathbf{0}, (\mathrm{diag}(\boldsymbol{\delta}))^{-1}\Big), \tag{18}$$

where $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes Gaussian distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$, and $\underline{\mathbf{s}}_m$ denotes the m-th column of $\underline{\mathbf{S}}$. The hypoerparameter $\boldsymbol{\delta}$ is further assigned as independent Gamma distribution

$$p(\boldsymbol{\delta}) = \prod_{l=1}^{\hat{L}} \Gamma(\delta_l; 1 + a, b), \tag{19}$$

where $a, b \rightarrow 0$ [9].

Under the assumption of Gaussian noise, we have

$$p(\underline{\mathbf{Y}}|\underline{\mathbf{S}}, \alpha, \boldsymbol{\varphi}) = \prod_{m=1}^{M} \mathcal{N}(\underline{\mathbf{y}}_m|\boldsymbol{\Phi}(\boldsymbol{\varphi})\underline{\mathbf{s}}_m, \alpha^{-1}\mathbf{I}), \tag{20}$$

where $\underline{\mathbf{y}}_m$ denotes the m-th column of $\underline{\mathbf{Y}}$ and $\alpha = \sigma^{-2}$ corresponds to the noise precision. Similarly, α is assigned as a Gamma distribution with $p(\alpha) = \Gamma(\alpha; 1 + a, b)$. We model $\boldsymbol{\varphi}$ as a noninformative uniform prior: $\boldsymbol{\varphi} \sim U([-\frac{\pi}{2}, \frac{\pi}{2}]^{\hat{L}})$ [4].

To find the optimal α, $\boldsymbol{\delta}$ and $\boldsymbol{\varphi}$, an EM algorithm will be utilized to perform the Bayesian inference, which will repeatedly construct a lower-bound on the evidence function $\ln p(\underline{\mathbf{Y}}, \alpha, \boldsymbol{\delta}, \boldsymbol{\varphi})$, and then optimize the lower-bound w.r.t. $\alpha, \boldsymbol{\delta}, \boldsymbol{\varphi}$. The lower-bound on $\ln p(\underline{\mathbf{Y}}, \alpha, \boldsymbol{\delta}, \boldsymbol{\varphi})$ is given by [9]

$$\langle \ln(p(\underline{\mathbf{Y}}|\underline{\mathbf{S}}, \alpha, \boldsymbol{\varphi})p(\underline{\mathbf{S}}|\boldsymbol{\delta})p(\alpha)p(\boldsymbol{\delta}))\rangle_{p(\underline{\mathbf{S}}|\underline{\mathbf{Y}},\alpha,\boldsymbol{\delta},\boldsymbol{\varphi})}, \tag{21}$$

where $\langle\cdot\rangle_{p(x)}$ denotes the expectation operator w.r.t. $p(x)$, and

$$p(\underline{\mathbf{S}}|\underline{\mathbf{Y}}, \alpha, \boldsymbol{\delta}, \boldsymbol{\varphi}) = \prod_{m=1}^{M} \mathcal{N}(\underline{\mathbf{s}}_m|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}), \tag{22}$$

with

$$\boldsymbol{\mu}_m = \alpha\boldsymbol{\Sigma}\boldsymbol{\Phi}^H(\boldsymbol{\varphi})\underline{\mathbf{y}}_m, \tag{23}$$

$$\boldsymbol{\Sigma} = (\alpha\boldsymbol{\Phi}^H(\boldsymbol{\varphi})\boldsymbol{\Phi}(\boldsymbol{\varphi}) + \mathrm{diag}(\boldsymbol{\delta}))^{-1}. \tag{24}$$

Note that the following updates of α and $\boldsymbol{\delta}$ can be obtained easily as in [4], [9]:

$$\alpha^{new} = \frac{TM + 2a}{2b + \sum_{m=1}^{M} \|\underline{\mathbf{y}}_m - \boldsymbol{\Phi}(\boldsymbol{\varphi})\boldsymbol{\mu}_m\|_2^2 + M\mathrm{tr}(\boldsymbol{\Theta})}, \tag{25}$$

$$\delta_l^{new} = \frac{2a + M}{2b + \sum_{m=1}^{M} [\boldsymbol{\Xi}_m]_{ll}}, \ \forall l, \tag{26}$$

where $\|\cdot\|_p$ denotes p-norm, $\mathrm{tr}(\cdot)$ denotes trace operator, $[\cdot]_{ll}$ denotes the (l, l)-th element, $\boldsymbol{\Theta} = \boldsymbol{\Phi}(\boldsymbol{\varphi})\boldsymbol{\Sigma}\boldsymbol{\Phi}^H(\boldsymbol{\varphi})$ and $\boldsymbol{\Xi}_m = \boldsymbol{\mu}_m{\boldsymbol{\mu}_m}^H + \boldsymbol{\Sigma}$. The update of $\boldsymbol{\varphi}$ is slightly different from [4], because all the matrices in our method are real-valued. Ignoring the independent terms, (21) becomes:

$$\langle \ln p(\underline{\mathbf{Y}}|\underline{\mathbf{S}}, \alpha, \boldsymbol{\varphi})\rangle_{p(\underline{\mathbf{S}}|\underline{\mathbf{Y}},\alpha,\boldsymbol{\delta},\boldsymbol{\varphi})}$$
$$= -\frac{\alpha}{2}(\sum_{m=1}^{M} \|\underline{\mathbf{y}}_m - \boldsymbol{\Phi}(\boldsymbol{\varphi})\boldsymbol{\mu}_m\|_2^2 + M\mathrm{tr}(\boldsymbol{\Phi}(\boldsymbol{\varphi})\boldsymbol{\Sigma}\boldsymbol{\Phi}^H(\boldsymbol{\varphi}))). \tag{27}$$

The derivative of (27) can be calculated by

$$\boldsymbol{\zeta}_{\boldsymbol{\varphi}} = [\zeta(\varphi_1), \zeta(\varphi_2), \ldots, \zeta(\varphi_{\hat{L}})]^T, \tag{28}$$

with

$$\boldsymbol{\zeta}(\varphi_l) = \underline{\mathbf{a}}'(\hat{\theta}_l + \varphi_l)^T\mathbf{G}^T\mathbf{G}\underline{\mathbf{a}}(\hat{\theta}_l + \varphi_l)c_1 + \underline{\mathbf{a}}'(\hat{\theta}_l + \varphi_l)^T\mathbf{G}^T\mathbf{c}_2,$$

where $\underline{\mathbf{a}}'(\hat{\theta}_l + \varphi_l) = d\underline{\mathbf{a}}(\hat{\theta}_l + \varphi_l)/d\varphi_l$, $c_1 = -\alpha(M\chi_{ll} + \sum_{m=1}^{M}(\mu_{ml})^2)$, $\mathbf{c}_2 = -\alpha(M\mathbf{G}\sum_{j\neq l}\chi_{jl}\underline{\mathbf{a}}(\hat{\theta}_j + \varphi_j) - \sum_{m=1}^{M}\mu_{ml}\underline{\mathbf{y}}_m^{-l})$, $\underline{\mathbf{y}}_m^{-l} = \underline{\mathbf{y}}_m - \mathbf{G}\sum_{j\neq l}\mu_{mj}\underline{\mathbf{a}}(\hat{\theta}_j + \varphi_j)$, μ_{ml} and χ_{jl} denote the l-th element and the (j, l)-th element of $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}$, respectively. Here, a fixed stepsize is used to update $\boldsymbol{\varphi}$ as in [4] to reduce the computational complexity:

$$\boldsymbol{\varphi}^{new} = \boldsymbol{\varphi} + \frac{\pi}{100\hat{L}} \cdot \mathrm{sign}(\boldsymbol{\zeta}_{\boldsymbol{\varphi}}), \tag{29}$$

where $\mathrm{sign}(\cdot)$ denotes the signum function. Actually, we do not need to refine every φ_l in each iteration, because any φ_ls corresponding to the rows of S that have non-significant elements can be safely removed. Therefore, we can similarly set a threshold to choose some proper active points as in [10].

The main computational complexity of our method is given as follows.

- The complexities in calculating $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are $\mathcal{O}(M\hat{L}^2)$ and $\mathcal{O}(T\hat{L}^2)$ per iteration, respectively.
- The complexities in calculating α, $\boldsymbol{\delta}$ and $\boldsymbol{\varphi}$ are $\mathcal{O}(T\hat{L}^2)$, $\mathcal{O}(M\hat{L})$ and $\mathcal{O}(TN\hat{L})$ per iteration, respectively.

Therefore, the total computational complexity of our method is $\mathcal{O}(T\hat{L}^2)$ per iteration, which is the same as that for the original complex-valued SBL method. However, it is worth noting that all the matrices in our method are real-valued and solving such problem involves real computations only. As a result, our method can save a considerable amount of computations (by a factor of four) compared with the original complex-valued SBL method, because complex multiplication costs four times as much as that of real multiplication [5].

Finally, we discuss the difference between our method and the traditional real-valued method (TRM) in the literature [11]. According to the commonly used real-valued transformation, TRM rewrites (9) as

$$\begin{bmatrix}\mathrm{Re}(\mathbf{Y})\\ \mathrm{Im}(\mathbf{Y})\end{bmatrix} = \begin{bmatrix}\mathrm{Re}(\boldsymbol{\Phi}(\boldsymbol{\varphi})) & -\mathrm{Im}(\boldsymbol{\Phi}(\boldsymbol{\varphi}))\\ \mathrm{Im}(\boldsymbol{\Phi}(\boldsymbol{\varphi})) & \mathrm{Re}(\boldsymbol{\Phi}(\boldsymbol{\varphi}))\end{bmatrix}\begin{bmatrix}\mathrm{Re}(\mathbf{S})\\ \mathrm{Im}(\mathbf{S})\end{bmatrix} + \begin{bmatrix}\mathrm{Re}(\mathbf{E})\\ \mathrm{Im}(\mathbf{E})\end{bmatrix}, \tag{30}$$

and then recover the sparse matrix $[\mathrm{Re}(\mathbf{S})^T, \mathrm{Im}(\mathbf{S})^T]^T$ directly from (30). Such transformation is applicable to any measurement matrix $\boldsymbol{\Phi}(\boldsymbol{\varphi})$, but it can neither reduce the computational burden [because the size of (30) is twice that of (9)], nor achieve a better noise suppression [because the SVD technique may not be applied to (30) due to the small number of measurement in (30)]. Actually, the advantages of our method come from exploiting the real-valued structure of $\boldsymbol{\Phi}(\boldsymbol{\varphi})$. When the imaginary part of $\boldsymbol{\Phi}(\boldsymbol{\varphi})$ is zero, (30) becomes:

$$\begin{bmatrix}\mathrm{Re}(\mathbf{Y})\\ \mathrm{Im}(\mathbf{Y})\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Phi}(\boldsymbol{\varphi}) & \\ & \boldsymbol{\Phi}(\boldsymbol{\varphi})\end{bmatrix}\begin{bmatrix}\mathrm{Re}(\mathbf{S})\\ \mathrm{Im}(\mathbf{S})\end{bmatrix} + \begin{bmatrix}\mathrm{Re}(\mathbf{E})\\ \mathrm{Im}(\mathbf{E})\end{bmatrix}, \tag{31}$$

which can be further rewritten as (10). It should be noted that the real-valued structure of $\boldsymbol{\Phi}(\boldsymbol{\varphi})$ is guaranteed by introducing the unitary transformation $\mathbf{Q}_{N_t}$ into the pilot matrix [see (9)]. To the best of our knowledge, such idea for the real-valued channel estimation has not been reported in the literature.

IV. Simulation Results

In this section, we will present several simulation results to illustrate the performance of our proposed method. We will compare our method with SBL [9], off-grid SBL [4], DFT, overcomplete DFT and TRM [11]. Both DFT and overcomplete DFT recover $\mathbf{H}$ using the l_1-norm minimization algorithm [12]. To verify that directly applying the SVD technique to the complex-valued problem cannot bring a significant performance improvement, we also include an SVD-based extension of off-grid SBL (named off-grid SVD) in simulations, where the measurement matrix is reduced to $T \times \min(N_r, M)$ by adopting the SVD technique. For fairness, TRM includes the same off-grid refining as ours. We use the 3GPP spatial channel model (SCM) [8] to generate the channel coefficients for an urban microcell. The downlink frequency is 2170 MHz, and the normalized mean square error (NMSE) is defined as

$$\frac{1}{M_{ct}} \sum_{m=1}^{M_{ct}} \frac{\|\mathbf{H}_m^{est} - \mathbf{H}_m\|_2^2}{\|\mathbf{H}_m\|_2^2}, \tag{32}$$

where $\mathbf{H}_m^{est}$ is the estimation of $\mathbf{H}_m$ at the m-th Monte Carlo trial and M_{ct} is the number of Monte Carlo trials. The experiments are carried out in MATLAB 8.4.0 on a PC with an AMD Ryzen 2700X CPU and 16 GB of RAM.

Simulation 1 shows the channel estimation performance with the number of training pilot symbols. Assume that the BS is composed of a ULA with 100 antennas and MU is equipped with 4 antennas. All the results are obtained based on 200 Monte Carlo trials. Every trial consists of $N_c = 2$ random scattering clusters ranging from $-90°$ to $90°$, and each cluster contains $N_s = 10$ sub-paths concentrated in a $20°$ angular spread. The SNR is fixed to 0 dB and the number of grid points is chosen as 200 except for DFT. From Fig. 1-a, it is shown that: 1) the off-grid methods (our method, off-grid SBL, off-grid SVD and TRM) always outperform others, because the grid refining procedure can fix the direction mismatch gap; 2) off-grid SVD almost achieves the same performance as off-grid SBL, which means that directly applying the SVD technique to the complex-valued problem could not reveal substantial performance gains; 3) compared with off-grid SBL, TRM has a little performance loss, because it expends the problem into real and imaginary parts, and considers the two parts are independent to each other; and 4) our method can improve the performance because of the noise suppression mentioned in Claim 1.

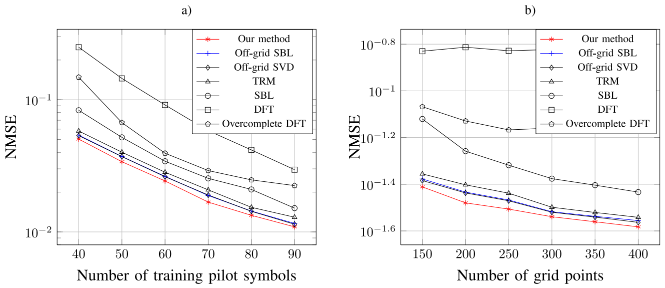

Fig. 1. NMSE of channel estimates versus the number of training pilot symbols and grid points, respectively. a) NMSE versus T; b) NMSE versus $\hat{L}$.

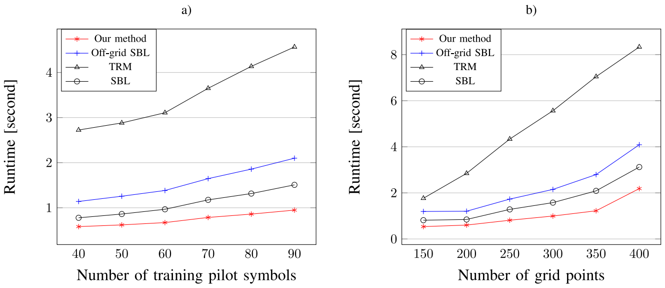

Fig. 2. Computational time versus the number of training pilot symbols and grid points, respectively. a) CPU Time versus T; b) CPU Time versus $\hat{L}$.

In Simulation 2, we study the impact of the number of grid points on the channel estimation performance. We consider the same scenario as in Simulation 1, except that the number of training pilot symbols is fixed to 50. Fig. 1-b shows the NMSE performance of the downlink channel estimate versus the number of grid points. It reconfirms that our method can achieve the best NMSE performance, no matter how many grid points are used.

Simulation 3 shows the runtime versus the number of training pilot symbols and grid points, respectively. As shown in Fig. 2, we observe that 1) the runtime required by all the methods increases as the number of training pilot symbols and grid points increase, respectively; 2) the runtime required by TRM is much more than others because it expends the recovery problem size twice; 3) off-grid SBL is slower than SBL since the additional grid refining increases the computation load; and 4) our method is much faster than others.

V. Conclusion

We have proposed an efficient real-valued SBL method that approaches the problem of downlink channel estimation. By introducing the real-valued transformation, our method converts the complex-valued problem into real-valued one, which brings a significant decrease in computational complexity, as well as a better performance. The numerical results presented verify the effectiveness of our method.

References

[1] J.-C. Shen, J. Zhang, E. Alsusa, and K. B. Letaief, "Compressed CSI acquisition in FDD massive MIMO: How much training is needed?" *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 4145–4156, Jun. 2016.

[2] Z. Gao, L. Dai, Z. Wang, and S. Chen, "Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6169–6183, Dec. 2015.

[3] X. Rao and V. K. N. Lau, "Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3261–3271, Jun. 2014.

[4] J. Dai, A. Liu, and V. K. N. Lau, "FDD massive MIMO channel estimation with arbitrary 2D-array geometry," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2584–2599, May 2018.

[5] K.-C. Huarng and C.-C. Yeh, "A unitary transformation method for angle-of-arrival estimation," *IEEE Trans. Signal Process.*, vol. 39, no. 4, pp. 975–977, Apr. 1991.

[6] J. Dai, X. Xu, and D. Zhao, "Direction-of-arrival estimation via real-valued sparse representation," *IEEE Antennas Wireless Propag. Lett.*, vol. 12, pp. 376–379, Mar. 2013.

[7] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[8] "Universal mobile telecommunications system (UMTS); spatial channel model for multiple input multiple output (MIMO) simulations, v11.0.0, release 11," 3GPP, Sophia Antipolis, France, Rep. TR 25.996, 2012.

[9] M. E. Tipping and A. Smola, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, no. 3, pp. 211–244, 2001.

[10] J. Dai, X. Bao, W. Xu, and C. Chang, "Root sparse Bayesian learning for off-grid DOA estimation," *IEEE Signal Process. Lett.*, vol. 24, no. 1, pp. 46–50, Jan. 2017.

[11] T.-H. Liu, "Comparisons of two real-valued MIMO signal models and their associated ZF-SIC detectors over the Rayleigh fading channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 12, pp. 6054–6066, Dec. 2013.

[12] D. L. Donoho and Y. Tsaig, "Fast solution of l_1-norm minimization problems when the solution may be sparse," *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 4789–4812, Nov. 2008.