Article:
Zhao, W, You, Y, Zhang, L orcid.org/0000-0002-4535-3200 et al. (2 more authors) (2023) OMPL-SBL Algorithm for Intelligent Reflecting Surface-Aided mmWave Channel Estimation. IEEE Transactions on Vehicular Technology. ISSN 0018-9545

https://doi.org/10.1109/TVT.2023.3287400

OMPL-SBL Algorithm for Intelligent Reflecting Surface-Aided mmWave Channel Estimation

Wuqiong Zhao, You You, *Member, IEEE*, Li Zhang, *Senior Member, IEEE*, Xiaohu You, *Fellow, IEEE*, and Chuan Zhang, *Senior Member, IEEE*

***Abstract*—Channel estimation (CE) is critical for intelligent reflecting surface (IRS) aided millimeter wave (mmWave) multiple input multiple output (MIMO) systems. In this paper, we propose the orthogonal matching pursuit list-sparse Bayesian learning (OMPL-SBL) algorithm which divides the cascaded channel estimation into two stages. The first stage calculates the prior for sparse Bayesian learning (SBL) using orthogonal matching pursuit list (OMPL) exploiting the grid sparsity of the cascaded channel, and the second stage employs the prior to obtain the accurate estimation using SBL. The proposed algorithm is able to achieve high estimation accuracy with low computational complexity compared to ℓ_1-minimization and Bayesian algorithms. In simulation, we show the proposed algorithm not only cuts down time complexity by more than 95% of the SBL algorithm, but also achieves a higher estimation accuracy.**

***Index Terms*—Sparse Bayesian learning (SBL), intelligent reflecting surface (IRS), channel estimation (CE), compressed sensing (CS).**

I. Introduction

RECENTLY the scheme of intelligent reflecting surface (IRS) has attracted more attention with the development of 6G. In millimeter wave (mmWave) communications, the IRS is capable of reflecting signal from the transmitter (Tx) to the receiver (Rx) whose direct connection to the Tx is blocked. The IRS scheme stands out even more because of its passive reflecting nature, which is easy to implement and maintain in practice as detailed in [1].

Channel estimation plays an indispensable role in wireless communication as the channel state information (CSI) is essential in precoding and beamforming. But this has been especially difficult for the IRS due to the large number of reflecting units as well as antennas in multiple input multiple output (MIMO) systems. Many compressed sensing (CS) algorithms can be employed to perform IRS channel estimation efficiently by exploiting the sparsity in the angular domain. The widely adopted orthogonal matching pursuit (OMP) algorithms [2], [3] can estimate the cascaded channel (Tx–IRS–Rx) with low computational complexity. However, in many scenarios, its accuracy is far from satisfactory. The sparse Bayesian learning (SBL) algorithm in [4] offers significantly better estimation accuracy especially in high signal-to-noise ratio (SNR) range. But the conventional SBL algorithm has one shortcoming that hinders it from real implementation. The complexity of the SBL algorithm is unacceptable which can be thousands of that of the OMP algorithm. Attempts have been made to solve the problem in literature [5]–[7]. The Laplace prior is given in [5] which results in improvement in both accuracy and efficiency. The block sparsity in mmWave MIMO systems for quasi-static and time-selective channels are considered in [6] to improve CE performance. Support knowledge can also be employed in the prior as is shown in [7]. However, these methods still have high complexity, and cannot be applied to exploit the unique channel structure exhibited in IRS-aided mmWave MIMO systems.

This paper proposes a novel two-stage cascaded channel estimation scheme for IRS-aided mmWave MIMO systems based on compressed sensing. The main contributions are summarized as followings:

1) Extending greedy algorithm OMP, the proposed OMPL provides efficient prior for SBL by considering the grid-sparsity channel structure that is unique in IRS-aided mmWave MIMO systems;
2) The proposed algorithm is able to achieve higher accuracy than ℓ_1-minimization based algorithms and Bayesian algorithms, and the computational complexity is remarkably reduced (by more than 95%) in comparison with the traditional SBL algorithm;
3) Parameters in OMPL-SBL can be adjusted for different system requirements as a tradeoff between estimation accuracy and computational complexity.

II. System Model

A. Cascaded Channel

We consider an IRS-assisted MIMO system, where the N_t-antenna Tx and the N_r-antenna Rx are equipped with the uniform linear array (ULA), and the IRS has a uniform planar array (UPA) with $M = M_x \times M_y$ reflecting units. Similar

 This work was supported in part by National Key R&D Program of China under Grant 2020YFB2205503, in part by CPSF under Grants 2022M722433, in part by NSFC under Grants 62122020 and 61871115, in part by the Jiangsu Provincial NSF under Grant BK20211512, in part by the Major Key Project of PCL under Grant PCL2021A01-2, in part by Jiangsu Excellent Postdoctoral Program, and in part by the Fundamental Research Funds for the Central Universities. *(Wuqiong Zhao and You You contribute equally to this work.) (Corresponding author: Chuan Zhang.)*

W. Zhao, Y. You, X. You, and C. Zhang are with the LEADS, the National Mobile Communications Research Laboratory, and the Frontiers Science Center for Mobile Information Communication and Security, Southeast University, Nanjing 211189, China; and also with the Purple Mountain Laboratories, Nanjing 211100, China. (email: chzhang@seu.edu.cn).

L. Zhang is with School of Electronic and Electrical Engineering, University of Leeds, Leeds, LS2 9JT, U.K..

to [3], [8], a narrowband geometric channel model is used to characterize the IRS–Rx channel $\mathbf{G}$ and Tx–IRS channel $\mathbf{R}$ as[1]

$$\begin{cases} \mathbf{G} = \sqrt{\dfrac{MN_r}{L_1}} \displaystyle\sum_{l_1=1}^{L_1} \alpha_{l_1} \mathbf{a}_r\left(\theta_{l_1}^R\right) \mathbf{a}_s\left(\vartheta_{l_1}^{S_1}, \varphi_{l_1}^{S_1}\right)^{\mathsf{H}}, \\ \mathbf{R} = \sqrt{\dfrac{MN_t}{L_2}} \displaystyle\sum_{l_2=1}^{L_2} \alpha_{l_2} \mathbf{a}_s\left(\vartheta_{l_2}^{S_2}, \varphi_{l_2}^{S_2}\right) \mathbf{a}_t\left(\theta_{l_2}^T\right)^{\mathsf{H}}, \end{cases} \tag{1}$$

where L_1 and L_2 are the numbers of IRS–Rx and Tx–IRS paths, respectively. α_{l_1} and α_{l_2} are the corresponding complex gain consisting of path loss. $\theta_{l_1}^R$ is the angle of arrival (AoA) at the Rx and $\theta_{l_2}^T$ is the angle of departure (AoD) at the Tx. $(\vartheta_{l_1}^{S_1}, \varphi_{l_1}^{S_1})$ and $(\vartheta_{l_2}^{S_2}, \varphi_{l_2}^{S_2})$ represent the (azimuth and elevation) AoD and AoA at the IRS, respectively. $\mathbf{a}_t(\theta_{l_2}^T) \in \mathbb{C}^{N_t \times 1}$, $\mathbf{a}_r(\theta_{l_1}^R) \in \mathbb{C}^{N_r \times 1}$ and $\mathbf{a}_s(\vartheta, \varphi) \in \mathbb{C}^{M \times 1}$ represent the normalized array steering vectors at the Tx, Rx and IRS, which can be formulated according to [8]. Therefore, $\mathbf{G}$ and $\mathbf{R}$ can be decomposed as

$$\begin{cases} \mathbf{G} = \mathbf{U}_{N_r} \mathbf{\Gamma} \mathbf{U}_M^{\mathsf{H}}, \\ \mathbf{R} = \mathbf{U}_M \mathbf{\Sigma} \mathbf{U}_{N_t}^{\mathsf{H}}, \end{cases} \tag{2}$$

where beamspace channels $\mathbf{\Gamma}$ and $\mathbf{\Sigma}$ are sparse matrices, $\mathbf{U}_{N_r} \in \mathbb{C}^{N_r \times N_r^G}$, $\mathbf{U}_M \in \mathbb{C}^{M \times M^G}$ and $\mathbf{U}_{N_t} \in \mathbb{C}^{N_t \times N_t^G}$ are dictionary matrices consisting of N_r^G, $M^G = M_x^G \times M_y^G$ and N_t^G steering vectors of predetermined grids at the Rx, IRS and Tx, respectively. For simplicity, we assume that all spatial angles are on the uniform grid from -1 to 1. Therefore, the cascaded channel is defined as

$$\mathbf{H} \triangleq \mathbf{G} \mathrm{diag}(\mathbf{\Psi}) \mathbf{R}, \tag{3}$$

where $\mathbf{\Psi} \in \mathbb{C}^{M \times 1}$ is the phase shift vector at the IRS.

B. Compressed Sensing Formulation

During channel estimation, T pilot blocks are transmitted. For the t-th pilot block ($t = 1, 2, \cdots, T$), the IRS has reflection vector $\mathbf{\Psi}_t$. N_t^B and N_r^B beams are formed at the Tx and Rx, respectively ($N_t^B < N_t$, $N_r^B < N_r$), so that N_t^B pilots are transmitted in one pilot block. Therefore, the received signal $\mathbf{y}_{t,p}$ considering the p-th ($p = 1, 2, \cdots, N_t^B$) transmitted beam with precoding vector $\mathbf{f}_{t,p}$ can be written as

$$\mathbf{y}_{t,p} = \mathbf{W}_t^{\mathsf{H}} \mathbf{H}_t \mathbf{f}_{t,p} s_{t,p} + \mathbf{W}_t^{\mathsf{H}} \mathbf{n}_{t,p}, \tag{4}$$

where $\mathbf{W}_t$ is the combining matrix at the Rx, $s_{t,p}$ is the transmitted symbol satisfying $|s_{t,p}| = 1$, $\mathbf{n}_{t,p} \sim \mathcal{CN}(0, \sigma_n^2 \mathbf{I}_{N_r^B})$ is the additive white Gaussian noise (AWGN), and $\mathbf{H}_t$ is the channel with IRS phase shift vector $\mathbf{\Psi}_t$. Collecting all N_t^B transmitted symbols, the received signal can be formulated in a matrix form as

$$\mathbf{Y}_t = \mathbf{W}_t^{\mathsf{H}} \mathbf{H}_t \mathbf{F}_t + \mathbf{N}_t, \tag{5}$$

where $\mathbf{Y}_t \triangleq [\mathbf{y}_{t,1}, \cdots, \mathbf{y}_{t,N_t^B}]$, $\mathbf{F}_t \triangleq [\mathbf{f}_{t,1}, \cdots, \mathbf{f}_{t,N_t^B}]$, and the noise matrix $\mathbf{N}_t \triangleq [\mathbf{W}_t^{\mathsf{H}} \mathbf{n}_{t,1}, \cdots, \mathbf{W}_t^{\mathsf{H}} \mathbf{n}_{t,N_t^B}]$. Therefore, Eq. (5) can be vectorized as

$$\begin{aligned} \mathbf{y}_t &\overset{(a)}{=} \mathrm{vec}(\mathbf{W}_t^{\mathsf{H}} \mathbf{G} \mathrm{diag}(\mathbf{\Psi}_t) \mathbf{R} \mathbf{F}_t) + \mathbf{n}_t \\ &\overset{(b)}{=} (\mathbf{F}_t^{\mathsf{T}} \mathbf{R}^{\mathsf{T}} \circledcirc \mathbf{W}_t^{\mathsf{H}} \mathbf{G}) \mathbf{\Psi}_t + \mathbf{n}_t \\ &\overset{(c)}{=} (\mathbf{F}_t^{\mathsf{T}} \otimes \mathbf{W}^{\mathsf{H}})(\mathbf{R}^{\mathsf{T}} \circledcirc \mathbf{G}) \mathbf{\Psi}_t + \mathbf{n}_t \\ &\overset{(d)}{=} (\mathbf{F}_t^{\mathsf{T}} \otimes \mathbf{W}^{\mathsf{H}})(\mathbf{U}_{N_t}^* \otimes \mathbf{U}_{N_r}) \mathbf{J} \mathbf{D} \mathbf{\Psi}_t + \mathbf{n}_t \end{aligned} \tag{6}$$

where in (a) the cascaded channel in (3) is substituted and $\mathbf{y}_t \triangleq \mathrm{vec}(\mathbf{Y}_t) \in \mathbb{C}^{N_t^B N_r^B \times 1}$, $\mathbf{n}_t \triangleq \mathrm{vec}(\mathbf{N}_t)$. In (b), the property of Khatri-Rao product is exploited as in [9]. In (c) and (d), $\mathbf{G}$ and $\mathbf{R}$ channels are decomposed according to Eq. (2), the property $(\mathbf{AB} \circledcirc \mathbf{CD}) = (\mathbf{A} \otimes \mathbf{C})(\mathbf{B} \circledcirc \mathbf{D})$ is applied, and $\mathbf{J} \triangleq \mathbf{\Sigma}^{\mathsf{T}} \otimes \mathbf{\Gamma}$ and $\mathbf{D} \triangleq \mathbf{U}_M^{\mathsf{T}} \circledcirc \mathbf{U}_M^{\mathsf{H}}$. Let $\widetilde{\mathbf{D}} = \mathbf{D}(1 : M^G, :)$ be the first M^G rows of $\mathbf{D}$ and $\mathbf{\Lambda}$ is a merged version of $\mathbf{J}$ as shown in [8]:

$$\mathbf{\Lambda}(:, i) = \sum_{n \in \mathcal{S}_i} \mathbf{J}(:, n), \tag{7}$$

where $\mathcal{S}_i$ is the set of indices in $\mathbf{D}$ that have the identical row as the i-th row. Therefore, the received signal in Eq. (6) can be simplified as

$$\begin{aligned} \mathbf{y}_t &= (\mathbf{F}_t^{\mathsf{T}} \otimes \mathbf{W}_t^{\mathsf{H}})(\mathbf{U}_{N_t}^* \otimes \mathbf{U}_{N_r}) \mathbf{\Lambda} \widetilde{\mathbf{D}} \mathbf{\Psi}_t + \mathbf{n}_t \\ &= \mathbf{Q}_t \boldsymbol{\lambda} + \mathbf{n}_t, \end{aligned} \tag{8}$$

where the mixed-product property of Kronecker product is employed with $\mathbf{Q}_t \triangleq (\widetilde{\mathbf{D}} \mathbf{\Psi}_t)^{\mathsf{T}} \otimes \left((\mathbf{F}_t^{\mathsf{T}} \otimes \mathbf{W}_t^{\mathsf{H}})(\mathbf{U}_{N_t}^* \otimes \mathbf{U}_{N_r})\right)$. After T pilot blocks, we obtain

$$\mathbf{y} = \mathbf{Q} \boldsymbol{\lambda} + \mathbf{n}, \tag{9}$$

where $\mathbf{y} \triangleq [\mathbf{y}_1^{\mathsf{T}}, \cdots, \mathbf{y}_T^{\mathsf{T}}]$, $\mathbf{Q} \triangleq [\mathbf{Q}_1^{\mathsf{T}}, \cdots, \mathbf{Q}_T^{\mathsf{T}}]$. The dimension of $\mathbf{y}$ is the number of measurement $m \triangleq T N_t^B N_r^B$. Considering the sparsity of $\boldsymbol{\lambda}$, CS based algorithms can be employed. For an arbitrary reflection vector $\mathbf{\Psi}$ used in data transformation, the cascaded channel $\mathbf{H}$ can be calculated as

$$\mathbf{H} = \mathrm{vec}_{N_r, N_t}^{-1}\left((\mathbf{U}_{N_t}^* \otimes \mathbf{U}_{N_r}) \mathbf{\Lambda} \widetilde{\mathbf{D}} \mathbf{\Psi}\right). \tag{10}$$

The SBL framework aims to find the most likely cascaded channel whose probability is defined as

$$p(\boldsymbol{\lambda}; \mathbf{\Gamma}) = \prod_{i=1}^{G} \frac{1}{\pi \gamma_i} \exp\left(-\frac{\|\boldsymbol{\lambda}(i)\|_2^2}{\gamma_i}\right), \tag{11}$$

where $G = N_t^G N_r^G M^G$, hyperparameter $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \cdots, \gamma_G]$ and $\mathbf{\Gamma} = \mathrm{diag}(\boldsymbol{\gamma})$. Different approaches such as expectation maximization (EM) used in [4] can be employed to update hyperparameter and estimate the sparse vector $\boldsymbol{\lambda}$. However, these approaches have extremely high complexity and thus not practical in channel estimation.

[1]*Notations:* Lower-case and upper-case boldface letter $\mathbf{x}$ and $\mathbf{X}$ denote a vector and a matrix respectively. $\mathbf{X}^{\mathsf{T}}$, $\mathbf{X}^{\mathsf{H}}$, $\mathbf{X}^*$ and $\mathbf{X}^\dagger$ denote the transpose, the conjugate transpose, the conjugate and the pseudoinverse. $\|\mathbf{x}\|_2$ and $\|\mathbf{X}\|_F$ denote the ℓ_2-norm of vector $\mathbf{x}$ and the Frobenius norm of matrix $\mathbf{X}$. $\mathrm{diag}(\mathbf{x})$ denotes the diagonal matrix with vector $\mathbf{x}$ on its diagonal. $\mathrm{vec}(\mathbf{X})$ denotes the vectorization of matrix $\mathbf{X}$. $\mathrm{vec}_{M,N}^{-1}(\mathbf{x})$ reshapes the vector $\mathbf{x}$ into an $M \times N$ matrix. $\mathrm{sum}(\mathbf{X})$ calculates the sum of all elements in matrix $\mathbf{X}$. $\mathrm{abs}(\mathbf{x})$ returns a vector with all elements being its absolute value. $\mathbf{A} \otimes \mathbf{B}$ denotes the Kronecker product and $\mathbf{A} \circledcirc \mathbf{B}$ denotes the Khatri-Rao product. $\mathbf{a} \oslash \mathbf{b}$ denotes the element-wise division. $\mathbf{I}_M$ is the identity matrix of size $M \times M$ and $\mathbf{O}_{M \times N}$ is an $M \times N$ matrix with all elements being zero. $\mathbf{1}_N$ is a vector of size N with all elements being one. $\mathbb{E}[a]$ is the expected value of a. $|\mathcal{S}|$ stands for the number of elements in set $\mathcal{S}$. Finally, $\mathcal{CN}(\mu, \sigma^2)$ is the complex-valued Gaussian distribution with mean μ and variance σ^2.

III. Proposed OMPL-SBL Algorithm

We propose the OMPL-SBL algorithm based on the Bayesian framework with an optimized prior for hyperparameter. The proposed algorithm consists of two stages: the OMPL stage that generates a prior considering the grid-sparsity channel property, and the SBL stage that estimates the channel.

A. The OMPL Stage

Alg. 1 shows the process of the OMPL stage. The maximum iteration number $k_{\max}$ is usually selected to be larger than the number of non-zero elements in $\boldsymbol{\lambda}$ which is $L \triangleq L_1 L_2$.

1) List Update (Steps 2~15): In Alg. 1 from steps 2 to 15, different from the traditional OMP algorithm which is a greedy algorithm choosing only one column support in one iteration, the OMPL algorithm chooses a list of n column supports. A column support is one column of the sensing matrix $\mathbf{Q}$ that corresponds to an element in the sparse vector $\boldsymbol{\lambda}$. The list update in the OMPL stage can be performed as iterations repeating BRANCH and MERGE updating column supports indices (indices of columns of sensing matrix $\mathbf{Q}$). Let $\boldsymbol{\Theta}^{(k)}$ denote the list of n different sets of selected column supports indices in the k-th iteration. It can be represented as

$$\boldsymbol{\Theta}^{(k)} = \left\{ \boldsymbol{\Xi}_1^{(k)}, \boldsymbol{\Xi}_2^{(k)}, \cdots, \boldsymbol{\Xi}_n^{(k)} \right\}, \tag{12}$$

where $\boldsymbol{\Xi}_i^{(k)}$ for $i = 1, 2, \cdots, n$ has k distinct column supports indices selected during the k iterations.

The BRANCH operation is summarized as follows. In the first iteration (step 4), BRANCH calculates the G residuals the same way as the conventional OMP algorithm and chooses n column supports indices with the least residuals as $\zeta_1^{(1)}, \zeta_2^{(1)}, \cdots, \zeta_n^{(1)}$. Therefore, the list after the first iteration $\boldsymbol{\Theta}^{(1)} = \{\{\zeta_1^{(1)}\}, \{\zeta_2^{(1)}\}, \cdots, \{\zeta_n^{(1)}\}\}$. In the k-th ($k > 1$) iteration (step 6), with $\boldsymbol{\Theta}^{(k-1)}$ from the previous iteration, each column supports indices set $\boldsymbol{\Xi}_i^{(k-1)}$ branches into n sets. For the set $\boldsymbol{\Xi}_i^{(k-1)}$, residuals $\|\mathbf{r}_{i,j}\|_2$ are calculated for $j = 1, 2, \cdots, G-k+1$ from new sets by adding one more column support index $\xi_{i,j}^{(k)}$. The residual vector $\mathbf{r}_{i,j} = \mathbf{y} - \mathbf{D}_{i,j}^{(k)}((\mathbf{D}_{i,j}^{(k)})^{\dagger}\mathbf{y})$, where $\mathbf{D}_{i,j}^{(k)} \triangleq \mathbf{Q}(:, \boldsymbol{\Xi}_i^{(k-1)} \cup \{\xi_{i,j}^{(k)}\})$ for $1 \le i \le n, 1 \le j \le G-k+1$. The newly chosen column supports indices corresponding to the least n residuals are $\zeta_1^{(k)}, \zeta_2^{(k)}, \cdots, \zeta_n^{(k)}$. Therefore, the n sets branched from $\boldsymbol{\Xi}_i^{(k-1)}$ can be represented as

$$\widehat{\boldsymbol{\Theta}}_i^{(k)} = \left\{ \boldsymbol{\Xi}_i^{(k-1)} \cup \{\zeta_1^{(k)}\}, \cdots, \boldsymbol{\Xi}_i^{(k-1)} \cup \{\zeta_n^{(k)}\} \right\}. \tag{13}$$

The MERGE operation in step 7 merges the n branched results from Eq. (13) into a new list as

$$\widehat{\boldsymbol{\Theta}}^{(k)} = \widehat{\boldsymbol{\Theta}}_1^{(k)} \cup \widehat{\boldsymbol{\Theta}}_2^{(k)} \cup \cdots \cup \widehat{\boldsymbol{\Theta}}_n^{(k)}. \tag{14}$$

Since there may exist repetition among $\widehat{\boldsymbol{\Theta}}_1^{(k)}, \widehat{\boldsymbol{\Theta}}_2^{(k)}, \cdots, \widehat{\boldsymbol{\Theta}}_n^{(k)}$, the size of $\widehat{\boldsymbol{\Theta}}^{(k)}$ can be smaller than n^2. In step 8, the n sets with the least residuals in $\widehat{\boldsymbol{\Theta}}^{(k)}$ will form the updated list $\boldsymbol{\Theta}^{(k)}$. In step 10, with the knowledge of column supports indices $\widehat{\boldsymbol{\Theta}}^{(k)}$, estimations for each set of column supports can be calculated as $\mathbf{M}^{(k)}(:, i) = (\mathbf{D}_i^{(k)})^{\dagger}\mathbf{y}$ in the k-th iteration with the i-th set of column supports, where $i = 1, 2, \cdots, n$.

Algorithm 1 OMPL Stage

Input: $\mathbf{y}, \mathbf{Q}$.
Initialization: $\widehat{\boldsymbol{\mu}} = \mathbf{O}_{G \times 1}$.
1: $\mathbf{r} = \mathbf{y}$;
2: **for** $k = 1, 2, \cdots, k_{\max}$ **do**
3: **if** $k = 1$ **then**
4: BRANCH: Initialize $\boldsymbol{\Xi}^{(1)}$ with indices of n column supports that have the least residual $\|\mathbf{r}\|_2$;
5: **else**
6: BRANCH: Calculate residuals $\mathbf{r}_{i,j}$ for $1 \le i \le n, 1 \le j \le G-k+1$;
7: MERGE: Update $\widehat{\boldsymbol{\Theta}}^{(k)}$ with Eq. (14);
8: MERGE: $\boldsymbol{\Theta}^{(k)} \subseteq \widehat{\boldsymbol{\Theta}}^{(k)}$ with n smallest $\|\mathbf{r}_{i,j}\|_2$;
9: **end if**
10: All estimations using $\boldsymbol{\Theta}^{(k)}$ form $\mathbf{M} \in \mathbb{C}^{G \times n}$;
11: All ℓ_2-norms of residuals of $\mathbf{M}$ form $\mathbf{w} \in \mathbb{R}^{1 \times n}$;
12: **if** $\min(\mathbf{w}) < r_{\text{th}}$ **then** ▷ Prior is accurate enough.
13: **break**;
14: **end if**
15: **end for**
16: $\mathbf{k} = \|\mathbf{w}\|_2(\mathbf{1}_n \oslash \mathbf{w})$; ▷ Normalized likelihood.
17: Structure check on $\mathbf{M} \cdot \mathbf{k}^{\mathsf{T}}$ as $\widehat{\boldsymbol{\mu}}$; ▷ Combine estimations.
18: $\alpha = \|\mathbf{y}\|_2 / \min(\mathbf{w})$; ▷ Residual coefficient.
Output: Estimated prior $\widehat{\boldsymbol{\mu}}$, residual coefficient α.

2) Estimation (Steps 16~18): The iteration of BRANCH and MERGE terminates when it reaches maximum iterations $k_{\max}$ or becomes accurate enough in step 12 measured by the residual threshold r_{th}. The likelihood for each set of column supports is related to the residuals in step 16. The estimated prior combines the n estimations by likelihood in step 17. The channel structure check is then applied, because it can be easily proved that the beam space cascaded channel $\mathbf{U}_{N_r}^{\mathsf{H}}\mathbf{H}\mathbf{U}_{N_t}$ exhibits grid sparsity, i.e. non-zero elements lie on certain rows and columns. Consider the summed up version of $\widehat{\boldsymbol{\Lambda}} = \text{vec}_{N_r^G, N_t^G}^{-1}(\widehat{\boldsymbol{\lambda}})$ as $\mathbf{H}_S$, which also has grid sparsity, formulated as

$$\mathbf{H}_S(i, j) = \|\widehat{\boldsymbol{\Lambda}}(jN_r^G + N_t^G, :)\|_2^2, \tag{15}$$

for $i = 1, 2, \cdots, N_r^G$ and $j = 1, 2, \cdots, N_t^G$. The estimated grid structure indices set $\widehat{\mathcal{I}}$ and $\widehat{\mathcal{J}}$ are chosen by

$$\widehat{\mathcal{I}} \subseteq \{1, 2, \cdots, N_r^G\}, \quad \widehat{\mathcal{J}} \subseteq \{1, 2, \cdots, N_t^G\}, \quad \text{s.t.} \begin{cases} \max\limits_{|\widehat{\mathcal{I}}|=L_1} \sum\limits_{i \in \widehat{\mathcal{I}}} \|\mathbf{H}_S(i, :)\|_2^2, \\ \max\limits_{|\widehat{\mathcal{J}}|=L_2} \sum\limits_{j \in \widehat{\mathcal{J}}} \|\mathbf{H}_S(:, j)\|_2^2. \end{cases} \tag{16}$$

Define the prior for the SBL stage as $\widehat{\boldsymbol{\mu}} \in \mathbb{R}^{N_t^G N_r^G M^G \times 1}$, and it is initialized as $\widehat{\boldsymbol{\mu}} = \text{abs}(\mathbf{M} \cdot \mathbf{k}^{\mathsf{T}})$. We perform the check

$$\widehat{\boldsymbol{\mu}}(kN_t^G N_r^G + jN_r^G + i) = \sqrt{\frac{\text{sum}\left(\mathbf{H}_S(\widehat{\mathcal{I}}, \widehat{\mathcal{J}})\right)}{L_1 L_2 M^G}}, \quad \text{if} \quad \mathbf{H}_S(i, j) < \beta \cdot \frac{\text{sum}\left(\mathbf{H}_S(\widehat{\mathcal{I}}, \widehat{\mathcal{J}})\right)}{L_1 L_2}, \tag{17}$$

for $i = 1, 2, \cdots, N_r^G$, $j = 1, 2, \cdots, N_t^G$, $k = 1, 2, \cdots, M^G$, and $\beta \in [0, 1)$ is the structure coefficient to control the degree of structure check. A large β results in a greater chance that the non-zero elements are marked as non-zero. Specially, when $\beta = 0$, no structure check is performed. In addition to the estimated prior $\widehat{\boldsymbol{\mu}}$, a residual coefficient α is calculated in step 18 by dividing the ℓ_2-norm of the received signal by the least residual $\min(\mathbf{w})$. This coefficient will be used in Eq. (18) of the SBL stage to initialize its hyperparameter.

3) Summary: The core of the OMPL stage is iterating to update the list of column supports indices by first extending $\boldsymbol{\Theta}^{(k-1)}$ $(k > 1)$ to $\widehat{\boldsymbol{\Theta}}^{(k)}$ and then choosing the optimal n sets from $\widehat{\boldsymbol{\Theta}}^{(k)}$ to form $\boldsymbol{\Theta}^{(k)}$. After this, prior for the SBL stage is calculated, with channel structure check. It can be derived that the conventional OMP algorithm is the special case for OMPL with $n = 1$.

B. The SBL Stage

The OMPL-SBL algorithm is depicted in Alg. 2 with a focus on the SBL stage.

Algorithm 2 OMPL-SBL

Input: $\mathbf{y}, \mathbf{Q}$.
Initialization: $\widehat{\boldsymbol{\lambda}} = \mathbf{O}_{G\times 1}$.
1: Calculate $\widehat{\boldsymbol{\mu}}, \alpha$ using **Algorithm 1**;
2: Initialize hyperparameter $\widehat{\boldsymbol{\gamma}}^{(1)}$ with Eq. (18);
3: **for** $k = 1, 2, \cdots, k_{\max}$ **do**
4: Prune hyperparameter $\boldsymbol{\gamma}^{(k)}$;
5: SBL iteration with updated $\widehat{\boldsymbol{\mu}}^{(k)}$, $\widehat{\boldsymbol{\Sigma}}^{(k)}$ and $\widehat{\boldsymbol{\gamma}}^{(k+1)}$ using Eq. (19);
6: **if** $\max \mathrm{abs}(\widehat{\boldsymbol{\gamma}}^{(k+1)} - \widehat{\boldsymbol{\gamma}}^{(k)}) < \varepsilon$ **then**
7: **break**; ▷ Reach termination condition.
8: **end if**
9: **end for**
10: **for** $i = 1, 2, \cdots, G$ **do**
11: **if** $|\widehat{\boldsymbol{\mu}}^{(k)}(i)| < \mu_{\mathrm{th}}$ **then**
12: $\widehat{\boldsymbol{\lambda}}(i) = 0$;
13: **else**
14: $\widehat{\boldsymbol{\lambda}}(i) = \widehat{\boldsymbol{\mu}}^{(k)}(i)$;
15: **end if**
16: **end for**
17: Estimate $\widehat{\mathbf{H}}$ with Eq. (10) where $\boldsymbol{\Lambda} = \mathrm{vec}^{-1}_{N_r^G, N_t^G}(\widehat{\boldsymbol{\lambda}})$;
Output: Estimated channel $\widehat{\mathbf{H}}$.

Step 1 uses the estimation result of Alg. 1. Then in step 2, the i-th element of hyperparameter prior $\widehat{\boldsymbol{\gamma}}^{(1)}$ is calculated by

$$\widehat{\boldsymbol{\gamma}}^{(1)}(i) = \begin{cases} 1 + \alpha(\widehat{\boldsymbol{\mu}}(i) - A)/B, & \text{if } \widehat{\boldsymbol{\mu}}(i) > 0, \\ 0, & \text{if } \widehat{\boldsymbol{\mu}}(i) = 0, \end{cases} \tag{18}$$

for $i = 1, 2, \cdots, G$, where $A = \min(\widehat{\boldsymbol{\mu}} \cap \mathbb{R}^+)$ is the minimum positive element in $\widehat{\boldsymbol{\mu}}$ and $B = \max \widehat{\boldsymbol{\mu}}$. The residual coefficient α indicates the reliability of the OMPL prior, since a larger α corresponding to a smaller final residual indicates that the support selection tends to be more accurate. As a result, the hyperparameter prior can be set with a larger variance to encourage faster convergence. Steps 3 to 9 follow the traditional SBL algorithm. The EM method for parameter update of step 5 used in [4] can be expressed as

$$\begin{cases} \widehat{\boldsymbol{\mu}}^{(k)} = \dfrac{1}{\sigma_n^2} \widehat{\boldsymbol{\Sigma}}^{(k)} \mathbf{Q}_r^{\mathsf{H}} \mathbf{y}, \\ \widehat{\boldsymbol{\Sigma}}^{(k)} = \left(\sigma_n^{-2} \mathbf{Q}_r^{\mathsf{H}} \mathbf{Q}_r + \left(\widehat{\boldsymbol{\Gamma}}^{(k)} \right)^{-1} \right)^{-1} \\ \qquad\quad = \widehat{\boldsymbol{\Gamma}}^{(k)} - \widehat{\boldsymbol{\Gamma}}^{(k)} \mathbf{Q}_r^{\mathsf{H}} \widehat{\boldsymbol{\Sigma}}_y^{-1} \mathbf{Q}_r \widehat{\boldsymbol{\Gamma}}^{(k)}, \end{cases} \tag{19}$$

where $\mathbf{Q}_r$ is the effective (pruned) sensing matrix, $\widehat{\boldsymbol{\Sigma}}_y = \sigma_n^2 \mathbf{I}_m + \mathbf{Q}_r \widehat{\boldsymbol{\Gamma}}^{(k)} \mathbf{Q}_r^{\mathsf{H}} \in \mathbb{C}^{m\times m}$. Hyperparameter is updated as $\widehat{\boldsymbol{\gamma}}^{(k+1)}(i) = \widehat{\boldsymbol{\Sigma}}^{(k)}(i,i) + |\widehat{\boldsymbol{\mu}}^{(k)}(i)|^2$ for $i = 1, 2, \cdots, G$. During the process, the hyperparameter $\widehat{\boldsymbol{\gamma}}^{(k)}$ becomes more sparse and noise estimation $\sigma_n^{(k)}$ is also updated [10]. At the same time, the size of $\widehat{\boldsymbol{\gamma}}^{(k)}$ can be reduced by setting a pruning threshold and prune $\widehat{\boldsymbol{\gamma}}^{(k)}$ in each iteration by eliminating all entries whose magnitude is smaller than the pruning threshold. This is of great significance to the proposed algorithm as the prior $\widehat{\boldsymbol{\mu}}$ given in the OMPL stage is already sparse compared to the non-informative prior that initializes all hyperparameter with non-zero values. As a result, the dimension of hyperparameter is much smaller so not only is each iteration more efficient, but also it cuts down the total iteration numbers. The complexity comparisons will be discussed later in III-C. The normal iteration termination condition is set as the largest hyperparameter change being smaller than a chosen threshold ε in step 6 or the number of iterations reaches $k_{\max}$. Steps 10 to 16 finally give the estimation of $\widehat{\boldsymbol{\lambda}}$ using hyperparameter. For those entries that correspond to a very small value smaller than the threshold μ_{th}, we can set it as zero in $\widehat{\boldsymbol{\lambda}}$ shown in step 12. Otherwise, the entry in $\widehat{\boldsymbol{\lambda}}$ is set as the corresponding mean value shown in step 14. The estimation result of the cascaded channel is calculated in step 17.

C. Complexity Comparisons

The computational complexity and run time comparisons are shown in Table I. The simulation is run on macOS with Apple M1 chip, and the system parameters used in run time evaluation are shown in Table II. Table I clearly shows the complexity of the above algorithms can be ranked as

$$\text{OMP} < \text{OMPL-SBL} < \text{BP-ADMM} \ll \text{SBL-EM}.$$

TABLE I
COMPUTATIONAL COMPLEXITY COMPARISONS.

Algorithm	Complexity Order	Run Time [s]
OMP	$\mathcal{O}(mLG)$	0.070 (1.00×)
BP-ADMM	$\mathcal{O}(KG^2)$	3.825 (54.6×)
SBL-EM	$\mathcal{O}(Qm^2G)$	150.0 (2143×)
OMPL stage*	$\mathcal{O}(nmLG)$	1.216 (17.4×)
OMPL-SBL	$\mathcal{O}(m(nLG + qmg))$	1.780 (25.4×)

[1] K is the average number of iterations in BP-ADMM.
[2] q and Q are the number of iterations in SBL-EM and the SBL stage of OMPL-SBL, respectively, and $q \ll Q$.
[3] $g \ll G = N_t^G N_r^G M^G$ is the effective size in the SBL stage.

D. Algorithm Analysis

The prior calculated by OMPL plays a key role in the channel estimation. The informative prior should be accurate

enough for the SBL algorithm to function well. Note that the informative prior produced by the OMPL does not necessarily distinguish all zero-value entries but set most zero-value entries to zero and carefully not to miss non-zero elements. If the value is set to zero, it is pruned out in the SBL stage. Setting most zero-value entries to zero will accelerate the convergence of SBL remarkably.

The performance of OMPL alone is not worse than the OMP algorithm, since the introduction of list ensures the residual is smaller than that of OMP, mitigating the problem of OMP's greedy selection of support. Moreover, OMPL can achieve a relatively satisfactory performance in support selection even at low SNRs, making OMPL-SBL stable at low SNRs.

A key feature of the proposed algorithm is that it is universal for almost all SBL algorithms. In section IV the EM-based SBL is tested, and other parameter update methods should also work under the proposed SBL stage framework.

One important flexibility of the proposed OMPL-SBL algorithm is that the parameters can be adjusted according to different requirements. The OMPL list size n and the channel structure check coefficient β are involved in the tradeoff between estimation accuracy and the computational complexity. It is evident that a larger OMPL list size n and/or channel structure check coefficient β will contribute to more non-zero elements in the SBL prior, i.e. a larger g. Thus, the complexity increases with n and β, as is discussed in III-C. Here we mainly consider the estimation performance concerning n and β.

The prerequisite of an accurate estimation is that all supports are selected without miss during the OMPL stage. It is already shown in III-D that OMPL offers a non-greedy estimation with support selection, and that a larger n results in better support selection. The upper bound of probability of missing support error P_e after channel structure check with b original misses can be approximated as

$$P_e \approx \frac{b(b+n-1)}{N_t^G N_r^G}\left(1-\beta\left(\frac{1+n}{L+n}+\frac{2L^2-3L}{6(L+n)^2}\right)\right)^2, \quad (20)$$

for $\beta > 0$. It is worth noting that P_e increases with n for a fixed value of b, but $\mathbb{E}[b]$ can be much smaller with a larger n. Overall, a larger β and a larger but not too large n contributes to better support selection.

From the perspective of the SBL stage, the prior provided by OMPL should satisfy certain requirements. Intuitively, a larger n makes OMPL less greedy, while $n = 1$ falls back on the conventional OMP which is greedy and therefore not reliable enough for support selection. When the number of non-zero elements in the prior g is large enough and the corresponding effective sensing matrix $\mathbf{Q}_r$ (columns of $\mathbf{Q}$ that correspond to non-zero elements in hyperparameter) satisfies the unique representation property (URP), the upper bound for the number of SBL local minima in an ideal scenario without noise is given as [11]

$$\binom{g}{m} = \frac{g!}{m!(g-m)!}, \quad (21)$$

which is increasing with g. Thus, the upper bound for the number of SBL local minima is smaller in the proposed method than the traditional SBL algorithm. According to [12], smaller number of local minima contributes to less convergence error. Therefore, if no support is missed in the OMPL stage, the performance is enhanced with our proposed method. But in order to ensure a reliable support and URP for $\mathbf{Q}_r$, g should not be too small, thus requiring n and β to be large enough.

The simulated tradeoff at $\mathrm{SNR} = 10\,\mathrm{dB}$ with pilot overhead $TN_t^B = 256$ is depicted in Fig. 1, which shows the existence of Pareto front with different n and β values. Specially, the yellow dots show that the NMSE performance deteriorates severely when no channel structure check is applied ($\beta = 0$).

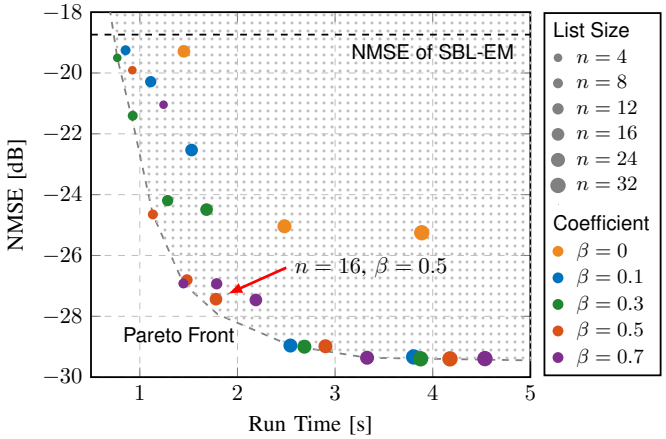


Fig. 1. NMSE performance and run time tradeoff in relation to parameters with Pareto front.

IV. Simulation Results

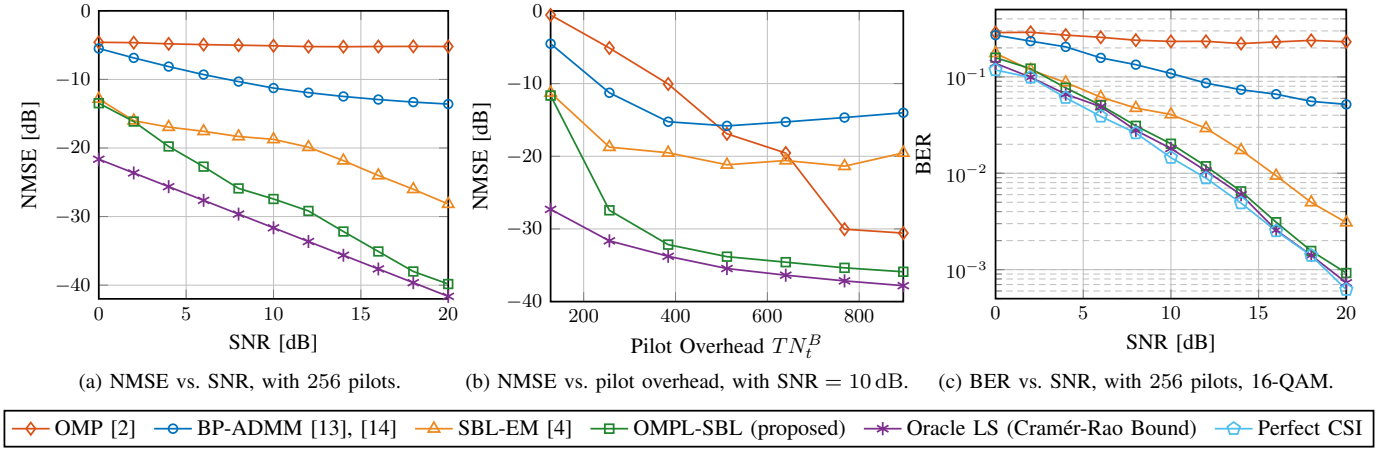
In this section, we evaluate the performance of the OMPL-SBL algorithm by C++ simulation with Armadillo library [15] for efficient linear algebra computation. System parameters used in simulation are shown in Table II. We choose appropriate parameters in OMPL-SBL based on the Pareto front in Fig. 1 as $n = 16$ and $\beta = 0.5$.

TABLE II
System Parameters in Simulations.

Parameters	Values
(N_t, N_r), (N_t^G, N_r^G), (N_t^B, N_r^B)	$(4,16)$, $(4,16)$, $(2,2)$
$M = M_x \times M_y$, $M^G = M_x^G \times M_y^G$	$64 = 8 \times 8$, $64 = 8 \times 8$
(L_1, L_2)	$(4,3)$

OMP algorithm [2] is a widely adopted channel estimation scheme with low complexity for IRS-assisted mmWave systems. [8], [9] adopt the OMP algorithm due to its low complexity. Since the second stage of the OMPL-SBL algorithm is SBL, we also compare it with the SBL [4] with EM update. It has been proven that SBL is equivalent to iterative reweighted ℓ_1-minimization [16]. Therefore, a popular ℓ_1-minimization based algorithm basis pursuit (BP) [13] with alternating direction method of multipliers (ADMM) solver [14] is also compared. The channel estimation performance is measured by the normalized mean square error (NMSE) which is formulated as $\mathbb{E}[\|\widehat{\mathbf{H}} - \mathbf{H}\|_F^2 / \|\mathbf{H}\|_F^2]$.

Fig. 2(a) compares the proposed OMPL-SBL algorithm with the conventional SBL, BP-ADMM and OMP methods



(a) NMSE vs. SNR, with 256 pilots. (b) NMSE vs. pilot overhead, with SNR $= 10\,\mathrm{dB}$. (c) BER vs. SNR, with 256 pilots, 16-QAM.

Fig. 2. Simulation results. Tests use C++ with Armadillo library [15] for efficient linear algebra computation.

in terms of NMSE vs. SNR. The performance of the proposed OMPL-SBL algorithm with OMPL list size $n = 16$ is better than conventional SBL algorithm by about $10\,\mathrm{dB}$ at high SNRs ($> 10\,\mathrm{dB}$) and is also close to the Cramér-Rao bound specified by Oracle LS. By contrast, OMP, BP-ADMM and SBL-EM cannot achieve a high estimation accuracy. The relationship between NMSE performance and pilot overhead number TN_t^B is shown in Fig. 2(b). The NMSE performance of the proposed algorithm stands out even with a small number of pilot overhead. We also evaluate the bit error rate (BER) performance in relation to SNR in Fig. 2(c). The minimum mean square error (MMSE) MIMO detector is applied for the 16-QAM signal. It demonstrates that the proposed OMPL-SBL algorithm can achieve low BER close to that of Oracle LS and that with perfect CSI.

Simulation also shows the convergence of the algorithm is much faster in the SBL stage, with approximately 8 to 16 iterations in total. By contrast, it takes the conventional SBL algorithm much longer to converge, owing to a larger total iteration number and longer calculation time in each iteration.

V. Conclusion

In this paper, we propose the OMPL-SBL algorithm for IRS-aided mmWave channel estimation that is not only accurate and stable, but also more efficient than the conventional SBL algorithm. The proposed algorithm can be readily adapted for orthogonal frequency-division multiplexing (OFDM) systems and the common sparsity among subcarriers can be exploited to further reduce the estimation complexity. In the presence of wideband beam squint effect, subcarrier grouping [17] can be employed to mitigate the influences. The simulation results show that the algorithm remarkably improves the performance with significantly lower time complexity. The proposed algorithm is able to provide precise CSI for beamforming design and IRS passive reflection design, which are of great significance in mmWave MIMO systems due to the high gain requirement. Moreover, OMPL list size n and channel structure coefficient β can be selected to adapt to different scenarios or changing circumstances, providing more flexibility and application potentials.

References

[1] C. Huang, A. Zappone, G. C. Alexandropoulos *et al.*, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4157–4170, Aug. 2019.

[2] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[3] X. Wei, D. Shen, and L. Dai, "Channel estimation for RIS assisted wireless communications–part II: An improved solution based on double-structured sparsity," *IEEE Commun. Lett.*, vol. 25, no. 5, pp. 1403–1407, May 2021.

[4] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Jun. 2001.

[5] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Bayesian compressive sensing using Laplace priors," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 53–63, Jan. 2010.

[6] S. Srivastava, A. Mishra, A. Rajoriya *et al.*, "Quasi-static and time-selective channel estimation for block-sparse millimeter wave hybrid MIMO systems: Sparse Bayesian learning (SBL) based approaches," *IEEE Trans. Signal Process.*, vol. 67, no. 5, pp. 1251–1266, Mar. 2019.

[7] J. Fang, Y. Shen, F. Li *et al.*, "Support knowledge-aided sparse Bayesian learning for compressed sensing," in *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2015, pp. 3786–3790.

[8] P. Wang, J. Fang, H. Duan *et al.*, "Compressed channel estimation for intelligent reflecting surface-assisted millimeter wave systems," *IEEE Signal Process. Lett.*, vol. 27, pp. 905–909, May 2020.

[9] K. Ardah, S. Gherekhloo, A. L. de Almeida *et al.*, "TRICE: A channel estimation framework for RIS-aided millimeter-wave MIMO systems," *IEEE Signal Process. Lett.*, vol. 28, pp. 513–517, Feb. 2021.

[10] Z. Zhang, T.-P. Jung, S. Makeig *et al.*, "Spatiotemporal sparse Bayesian learning with applications to compressed sensing of multichannel physiological signals," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 6, pp. 1186–1197, Nov. 2014.

[11] D. P. Wipf, "Bayesian methods for finding sparse representations," Ph.D. dissertation, Dept. of Electr. Eng., Univ. California, San Diego, 2006.

[12] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[13] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[14] S. Boyd, N. Parikh, E. Chu *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011, sec. 6.2.

[15] C. Sanderson and R. Curtin, "Armadillo: a template-based C++ library for linear algebra," *J. Open Source Softw.*, vol. 1, no. 2, p. 26, 2016.

[16] D. Wipf and S. Nagarajan, "Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 317–329, Apr. 2010.

[17] J. Rodriguez-Fernandez and N. Gonzalez-Prelcic, "Channel estimation for frequency-selective mmWave MIMO systems with beam-squint," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.