

SITE-SPECIFIC MILLIMETER-WAVE COMPRESSIVE CHANNEL ESTIMATION ALGORITHMS WITH HYBRID MIMO ARCHITECTURES

Sai Subramanyam Thoota¹, Dolores Garcia Marti^{2,3}, Özlem Tuğfe Demir^{4,5}, Rakesh Mundlamuri⁶, Joan Palacios^{2,3}, Cenk M. Yetis⁷, Christo Kurisummoottil Thomas⁶, Sameera H. Bharadwaja¹, Emil Björnson^{4,5}, Pontus Giselsson⁷, Marios Kountouris⁶, Chandra R. Murthy¹, Nuria González-Prelcic⁸, Joerg Widmer²

¹Indian Institute of Science, Bangalore, India, ²IMDEA Networks, Madrid, Spain, ³Universidad Carlos III, Madrid, Spain, ⁴KTH Royal Institute of Technology, Stockholm, Sweden, ⁵Linköping University, Linköping, Sweden, ⁶EURECOM, Sophia-Antipolis, France, ⁷Lund University, Lund, Sweden, ⁸North Carolina State University, Raleigh, USA

NOTE: Corresponding author: Sai Subramanyam Thoota, thoota@iisc.ac.in

Abstract – In this paper, we present and compare three novel model-cum-data-driven channel estimation procedures in a millimeter-wave Multi-Input Multi-Output (MIMO) Orthogonal Frequency Division Multiplexing (OFDM) wireless communication system. The transceivers employ a hybrid analog-digital architecture. We adapt techniques from a wide range of signal processing methods, such as detection and estimation theories, compressed sensing, and Bayesian inference, to learn the unknown virtual beamspace domain dictionary, as well as the delay-and-beamspace sparse channel. We train the model-based algorithms with a site-specific training dataset generated using a realistic ray tracing-based wireless channel simulation tool. We assess the performance of the proposed channel estimation algorithms with the same site's test data. We benchmark the performance of our novel procedures in terms of normalized mean squared error against an existing fast greedy method and empirically show that model-based approaches combined with data-driven customization unanimously outperform the state-of-the-art techniques by a large margin. The proposed algorithms were selected as the top three solutions in the "ML5G-PHY Channel Estimation Global Challenge 2020" organized by the International Telecommunication Union.

Keywords – Bayesian inference, channel estimation, compressed sensing, data-driven, hybrid MIMO, mmWave

1. INTRODUCTION

Millimeter-Wave (mmWave) wireless communication is one of the potential technologies proposed for the next generation communication systems (5G and beyond) to meet the ever-increasing demand for high data rates. The mmWave frequency spectrum, ranging from 30 GHz to 300 GHz, is attractive because it offers large bandwidths (~ 2GHz), resulting in very high data rates and low latency. These advantages come at a cost of higher path loss due to several factors, such as blockages and oxygen absorption at mmWave frequencies, which in turn bring several engineering challenges in adopting this technology in commercial wireless communication systems.

A potential solution to overcome this problem is beamforming, which leverages the availability of multiple antennas at the transmitter and receiver. In particular, millimeter wavelengths enable one to accommodate a larger number of antennas into the same physical space, and thereby attain high beamforming gains. However, a fully digital architecture in a Multi-Input Multi-Output (MIMO) system, i.e., one Radio Frequency (RF) chain per antenna, and one complex-valued Analog-to-Digital Converter (ADC) per RF chain is less appealing both from commercial and engineering perspectives due to its high cost and energy requirements. Therefore, a hybrid MIMO architecture is proposed in the literature as a potential solution to solve this problem [1].

In a hybrid MIMO system, multiple antennas are connected to an RF chain using a phase shifter network (RF precoder/combiner), and a digital precoder/combiner is employed in the complex baseband side of the transceiver. The RF and digital precoders/combiners are configured by optimizing a system performance metric such as the sum rate or signal to interference noise ratio. Unlike a fully analog architecture, a hybrid architecture allows one to reduce the number of RF chains, while supporting multi-stream and multi-user transmissions. The major challenges then are in estimating the mmWave wireless channel and configuring the RF and digital precoders/combiners based on the channel estimate. The problem is exacerbated by the fact that only the low dimensional RF combined signals at the baseband are available for estimating the channel. Since the system does not have any knowledge of the channel state during the channel estimation phase, the baseband precoders/combiners are set to the identity matrix and random phase shifts are chosen for the RF precoders/combiners.

MmWave channel estimation in a hybrid MIMO architecture is a well studied problem, and we provide a brief overview of some of the key existing literature here. The simplest channel estimation method in hybrid MIMO systems is the Least Squares (LS)-based approach [2], which is inherited from conventional MIMO [3]. A more refined solution to channel estimation is to exploit both the delay

and angular domain sparsity that mmWave channels exhibit. In this approach, the channel estimation problem is formulated as a sparse recovery problem [4]. Such compressive sensing based estimation techniques were first developed for frequency-flat hybrid mmWave MIMO systems [5, 6]. Recently, frequency-selective channels with OFDM-based communications leading to a more complex estimation problem have also been considered, with different approaches to exploit the sparse channel characteristics [4, 7, 8]. Several model-based signal processing techniques for mmWave channel estimation under various system settings can be found in [9–23].

Machine Learning and Artificial Intelligence (ML/AI) have been shown to be powerful tools in diverse areas such as natural language processing, speech processing, and image recognition, where it is challenging to design specific model-based algorithms. However, the impact of ML/AI on the design and optimization of communication systems is yet to be extensively studied, especially under realistic and practically meaningful settings. We aim to address some of the aspects of ML/AI in wireless communications here.

In this paper, we study the potential advantage of using data-driven approaches for channel estimation in hybrid MIMO systems. The model-cum-data driven algorithms we develop in this paper were selected as the **top three** solutions in the “**ML5G-PHY Channel Estimation Global Challenge 2020**” organized by the International Telecommunication Union (ITU)¹. Our main goal in this paper is to present and contrast these three algorithms for estimating an mmWave channel in a hybrid MIMO system. We compare the Normalized Mean Squared Error (NMSE) performance of these approaches and discuss the machine learning techniques relevant for the challenge at hand. These approaches utilize the channel training datasets generated using the *Raymobtime* tool to customize the algorithms so that they perform well for a test dataset generated in a similar environment [24].

We provide a brief overview of the three solutions below:

1. We integrate a fast greedy search with a high-performing Bayesian inference method in the first approach.² We use a Multi-Level Greedy Search (MLGS) to learn the sparsifying virtual beamspace dictionary that reduces the dimensionality of the problem and use the learned dictionary to estimate the channel using a Sparse Bayesian Learning (SBL) method. We finally exploit the delay-domain sparsity to de-noise the estimated channels. We name the algorithm as MLGS-SBL.
2. As a second approach, we propose another SBL-based algorithm to exploit the sparsity of the channel. We utilize the pattern-coupling concept to

model possible block sparsity patterns among the consecutive Angle Of Arrivals (AoAs) and Angle Of Departures (AoDs). As a first step, we obtain the time-domain channels from the provided training dataset via the inverse Discrete Fourier Transform (DFT) and remove the channel taps with small magnitude. Then, we apply the algorithm to the ground truth time-domain channels to obtain the sparse representations. Using joint angular distribution learned from training data, we refine the grids and pattern-coupling relations in the testing stage to improve the channel estimation quality. This approach is called “Pattern-Coupled Sparse Bayesian Learning for Channel Estimation with Dominating Delay Taps (PCSBL-DDT)” in the paper.

3. The third approach, Projection Cuts Orthogonal Matching Pursuit (PC-OMP), is based on the Orthogonal Matching Pursuit (OMP) algorithm. This method makes use of the sparsity of the mm-wave channel to extract channel components. At each iteration of the OMP algorithm, a coarse estimate of the strongest path parameters (AoA, AoD, and delay) is obtained by a low resolution grid search. Then, each of the three parameters is refined alternately, assuming the other two to be known. In this way, we keep the algorithm’s complexity low without compromising on its accuracy. At the end of each iteration, a path detection hypothesis is tested, and, if successful, the path is subtracted from the channel. This process is repeated until no additional path is detected.

1.1 Notation

The operator $(\cdot)^*$ represents the conjugate transpose or conjugate for a matrix or a scalar, respectively. $\bar{\mathbf{A}}$, \mathbf{A}^T , and \mathbf{A}^\dagger denote the conjugate, transpose, and Moore-Penrose pseudoinverse of a matrix \mathbf{A} , respectively. The multivariate complex Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix \mathbf{C} is denoted by $\mathcal{CN}(\boldsymbol{\mu}, \mathbf{C})$ and its probability density function (pdf) of a random vector \mathbf{x} is denoted by $\mathcal{CN}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C})$. $\text{blkdiag}(\cdot)$ represents the block-diagonal part of a matrix. $\text{diag}(\mathbf{X})$ or $\text{diag}(\mathbf{x})$ represents a vector obtained by the diagonal elements of the matrix \mathbf{X} or the diagonal matrix obtained with the elements of \mathbf{x} in the diagonal, respectively. $\mathbf{A} \otimes \mathbf{B}$ denotes the Kronecker product of the matrices \mathbf{A} and \mathbf{B} . $\|\mathbf{A}\|_F$ denotes the Frobenius norm of a matrix \mathbf{A} . $\langle \mathbf{a}, \mathbf{b} \rangle$ is the inner product of the two vectors \mathbf{a} and \mathbf{b} . The trace of a matrix \mathbf{A} is denoted by $\text{tr}(\mathbf{A})$. Tx and Rx denote the transmitter and receiver, respectively. We use $\text{vec}(\mathbf{A})$ to vectorize the matrix \mathbf{A} column-wise. $\mathbb{E}[\cdot]$ denotes the expectation.

2. SYSTEM MODEL

We consider a single cell mmWave hybrid MIMO-OFDM system with N_t antennas at the transmitter (Tx) and N_r antennas at the receiver (Rx), as shown in Fig. 1.

¹<https://www.itu.int/en/ITU-T/AI/challenge/2020/Pages/default.aspx>

²The order in which the algorithms are presented is unrelated to their ranking in the ITU ML5G-PHY channel estimation challenge. The ordering is based on ease of presentation and readability of the paper.

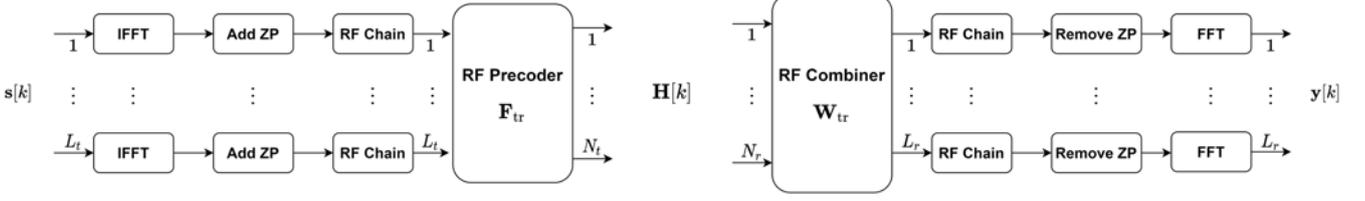


Fig. 1 – mmWave MIMO system based on a hybrid analog-digital architecture.

The Tx and Rx are equipped with L_t and L_r RF chains, respectively. The training input signal $\mathbf{s}[k] \in \mathbb{C}^{L_t \times 1}$ on the k^{th} subcarrier is OFDM modulated, up-converted to RF, and analog precoded using $\mathbf{F}_{\text{tr}} \in \mathbb{C}^{N_t \times L_t}$, and transmitted over the air to the Rx via an mmWave channel denoted by $\mathbf{H}[k]$ on the k^{th} subcarrier. The received signal is filtered using an RF combining matrix $\mathbf{W}_{\text{tr}} \in \mathbb{C}^{N_r \times L_r}$, down-converted to baseband, OFDM demodulated to obtain the k^{th} subcarrier's complex baseband signal $\mathbf{y}[k] \in \mathbb{C}^{L_r \times 1}$. We denote the total number of subcarriers by K .

In the initial access phase, the system has no prior knowledge of the channel, and therefore the precoder and combiner matrices cannot be designed to optimize any chosen performance metric. Hence, we choose random analog precoding and combining matrices (with unit modulus entries). In our system model, we adopt a fully connected phase shifter network for analog precoding/combining. The analog precoders and combiners are frequency-flat, and thus are the same for each subcarrier $k = 1, \dots, K$. The system operates with Uniform Linear Arrays (ULAs) at both the Tx and Rx with half wavelength spacing between consecutive antennas. The total number of training frames is denoted by M .

After RF combining, down-conversion, zero prefix removal and DFT, the complex baseband signal received during the m^{th} training frame for the k^{th} subcarrier, denoted by $\mathbf{y}^{(m)}[k] \in \mathbb{C}^{L_r \times 1}$ is given by

$$\mathbf{y}^{(m)}[k] = \mathbf{W}_{\text{tr}}^{(m)*} (\mathbf{H}[k] \mathbf{F}_{\text{tr}}^{(m)} \mathbf{q}^{(m)} t^{(m)}[k] + \mathbf{n}^{(m)}[k]), \quad (1)$$

for $m = 1, \dots, M$ where $\mathbf{H}[k] \in \mathbb{C}^{N_r \times N_t}$ represents the frequency domain MIMO channel matrix for the k^{th} subcarrier. We choose the m^{th} training signal as $\mathbf{s}^{(m)}[k] = \mathbf{q}^{(m)} t^{(m)}[k]$, where $t^{(m)}[k] \in \mathbb{C}$ is a subcarrier-dependent pilot symbol, and $\mathbf{q}^{(m)} \in \mathbb{C}^{L_t \times 1}$ is a frequency-flat vector whose entries are chosen as $\frac{1}{\sqrt{2L_t}}(a + jb)$, where $a, b \in \{-1, 1\}$ and are uniformly distributed. The noise vector $\mathbf{n}^{(m)}[k]$ is independently and identically distributed across K subcarriers as $\mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_{N_r})$. We define the transmit Signal-to-Noise Ratio (SNR) as $\rho = \frac{1}{\sigma_n^2}$. After compensating for $t^{(m)}[k]$, and vectorizing (1), we use the

result $\text{vec}(\mathbf{AXB}) = (\mathbf{B}^T \otimes \mathbf{A})\text{vec}(\mathbf{X})$ to obtain

$$\text{vec}(\mathbf{y}^{(m)}[k]) = \underbrace{(\mathbf{q}^{(m)T} \mathbf{F}_{\text{tr}}^{(m)T} \otimes \mathbf{W}_{\text{tr}}^{(m)*})}_{\Phi^{(m)}} \text{vec}(\mathbf{H}[k]) + \mathbf{W}_{\text{tr}}^{(m)*} \mathbf{n}^{(m)}[k]. \quad (2)$$

Next, we describe the mmWave channel model.

2.1 Channel model

We consider a frequency-selective geometric channel model that is constant across M training frames, and has N_c delay taps [4, 25]. The d^{th} delay tap is modeled as a clustered channel with L paths as

$$\mathbf{H}_d = \sqrt{\frac{N_t N_r}{L \rho_L}} \sum_{\ell=1}^L \alpha_{\ell} p(dT_s - \tau_{\ell}) \mathbf{a}_R(\phi_{\ell}) \mathbf{a}_T^*(\theta_{\ell}), \quad (3)$$

where ρ_L is the path loss between Tx and Rx, α_{ℓ} represents the complex path gain, ϕ_{ℓ} is the AoA, θ_{ℓ} is the AoD, τ_{ℓ} denotes the delay of the ℓ^{th} path. The corresponding Rx and Tx array steering vectors are denoted by $\mathbf{a}_R(\phi_{\ell}) \in \mathbb{C}^{N_r \times 1}$ and $\mathbf{a}_T(\theta_{\ell}) \in \mathbb{C}^{N_t \times 1}$, respectively. The pulse shaping and other low pass filtering evaluated at τ is represented by $p(\tau)$, and T_s is the sampling interval. We represent the MIMO channel \mathbf{H}_d in a matrix form as

$$\mathbf{H}_d = \mathbf{A}_R \Delta_d \mathbf{A}_T^*, \quad (4)$$

where $\mathbf{A}_R \in \mathbb{C}^{N_r \times L}$ and $\mathbf{A}_T \in \mathbb{C}^{N_t \times L}$ contain the Rx and Tx array steering vectors $\mathbf{a}_R(\phi_{\ell})$ and $\mathbf{a}_T(\theta_{\ell})$ as their columns for $\ell = 1, \dots, L$, respectively. $\Delta_d \in \mathbb{C}^{L \times L}$ is a diagonal matrix containing the complex channel gains. We take a K -point DFT of the delay-domain channel to get the frequency domain representation as

$$\mathbf{H}[k] = \sum_{d=0}^{N_c-1} \mathbf{H}_d \exp\left(-\frac{j2\pi kd}{K}\right) = \mathbf{A}_R \Delta[k] \mathbf{A}_T^*, \quad (5)$$

for $k = 0, \dots, K - 1$, and

$$\Delta[k] = \sum_{d=0}^{N_c-1} \Delta_d \exp\left(-\frac{j2\pi kd}{K}\right). \quad (6)$$

We adopt the extended virtual channel model in [25] to represent \mathbf{H}_d as

$$\mathbf{H}_d \approx \tilde{\mathbf{A}}_R \Delta_d^v \tilde{\mathbf{A}}_T^*, \quad (7)$$

where the dictionary matrices $\tilde{\mathbf{A}}_R \in \mathbb{C}^{N_r \times G_r}$ and $\tilde{\mathbf{A}}_T \in \mathbb{C}^{N_t \times G_t}$ contain the Rx and Tx array steering vectors evaluated on a grid of size G_r for the AoA and a grid of size G_t for the AoD, respectively. When G_r and G_t are chosen properly, i.e., much greater than L , $\Delta_d^v \in \mathbb{C}^{G_r \times G_t}$ becomes a sparse matrix containing the channel path gains on the locations that match with the actual AoDs and AoAs. We represent (7) in the frequency domain as

$$\mathbf{H}[k] \approx \tilde{\mathbf{A}}_R \Delta^v[k] \tilde{\mathbf{A}}_T^*, \quad (8)$$

for $k = 0, \dots, K-1$, and

$$\Delta^v[k] = \sum_{d=0}^{N_c-1} \Delta_d^v \exp\left(-\frac{j2\pi kd}{K}\right). \quad (9)$$

Note that the dictionary matrices $\tilde{\mathbf{A}}_R$ and $\tilde{\mathbf{A}}_T$ are common to all the subcarriers due to the frequency-flat array response vectors. Hence, the sparse matrices $\Delta^v[k]$ for $k = 1, \dots, K$ have the non-zero elements at the same indices. This means that they share a common sparsity pattern [4].

Now, we vectorize (8) to get

$$\text{vec}(\mathbf{H}[k]) = \left(\tilde{\mathbf{A}}_T \otimes \tilde{\mathbf{A}}_R\right) \text{vec}(\Delta^v[k]). \quad (10)$$

We define $\tilde{\mathbf{\Psi}} = \tilde{\mathbf{A}}_T \otimes \tilde{\mathbf{A}}_R \in \mathbb{C}^{N_t N_r \times G_t G_r}$ and $\mathbf{h}^v[k] = \text{vec}(\Delta^v[k]) \in \mathbb{C}^{G_t G_r}$, and substitute $\text{vec}(\mathbf{H}[k])$ in (2) to get

$$\text{vec}(\mathbf{y}^{(m)}[k]) = \tilde{\mathbf{\Phi}}^{(m)} \tilde{\mathbf{\Psi}} \mathbf{h}^v[k] + \mathbf{n}_c^{(m)}[k], \quad (11)$$

where $\mathbf{n}_c^{(m)}[k] = \mathbf{W}_{\text{tr}}^{(m)*} \mathbf{n}^{(m)}[k]$. By concatenating the RF combined signals of M training frames, we get

$$\underbrace{\begin{bmatrix} \mathbf{y}^{(1)}[k] \\ \vdots \\ \mathbf{y}^{(M)}[k] \end{bmatrix}}_{\mathbf{y}[k]} = \underbrace{\begin{bmatrix} \tilde{\mathbf{\Phi}}^{(1)} \\ \vdots \\ \tilde{\mathbf{\Phi}}^{(M)} \end{bmatrix}}_{\tilde{\mathbf{\Phi}}} \tilde{\mathbf{\Psi}} \mathbf{h}^v[k] + \underbrace{\begin{bmatrix} \mathbf{n}_c^{(1)}[k] \\ \vdots \\ \mathbf{n}_c^{(M)}[k] \end{bmatrix}}_{\mathbf{n}_c[k]}. \quad (12)$$

Now, by stacking the received signals of K subcarriers, we get the final system equation

$$\begin{aligned} \mathbf{Y} &= [\mathbf{y}[1] \quad \dots \quad \mathbf{y}[K]] \\ &= \tilde{\mathbf{\Phi}} \tilde{\mathbf{\Psi}} [\mathbf{h}^v[1] \quad \dots \quad \mathbf{h}^v[K]] + [\mathbf{n}_c[1] \quad \dots \quad \mathbf{n}_c[K]] \\ &= \tilde{\mathbf{\Phi}} \tilde{\mathbf{\Psi}} \mathbf{H}^v + \mathbf{N}_c. \end{aligned} \quad (13)$$

Our goal is to estimate $\mathbf{H}[k]$, for $k = 0, \dots, K-1$, given \mathbf{Y} and $\tilde{\mathbf{\Phi}}$. As the AoDs and AoAs are the same for all the subcarriers, $\mathbf{H}^v \in \mathbb{C}^{G_t G_r \times K}$ has a joint row sparse structure, i.e., the support set of each column of \mathbf{H}^v are the same. Also, we do not have the knowledge of the sparsifying dictionary $\tilde{\mathbf{\Psi}}$ and the noise variance, which makes the channel estimation problem more challenging. In the following sections, we present three different solutions to this channel estimation problem.

3. MLGS-SBL

In this section, we propose a model-based approach using the framework of Compressed Sensing (CS), to estimate the mmWave channel given the received pilot measurements and the frequency-flat transmit vector. We integrate a fast greedy search procedure and a high performing statistical inference method to estimate the channel. The algorithm consists of the following steps:

1. Preconditioning
2. Multi-level greedy search for dictionary learning
3. Noise variance estimation
4. Sparse Bayesian learning for channel estimation
5. Channel de-noising

We provide a detailed description of each step below.

3.1 Preconditioning

Sparse signal recovery using greedy algorithms, such as OMP, are likely to choose the correct support set when the noise covariance matrix is diagonal. In our mmWave channel estimation problem, RF combining by \mathbf{W}_{tr} at the front end of the receiver results in correlated noise, which needs to be whitened using a preconditioning filter [4].

The scaled noise covariance matrix before whitening is

$$\begin{aligned} \mathbf{C}_w &= \frac{\mathbb{E}[\mathbf{n}_c[k] \mathbf{n}_c^*[k]]}{\sigma^2} \\ &= \text{blkdiag}\{\mathbf{W}_{\text{tr}}^{(1)*} \mathbf{W}_{\text{tr}}^{(1)}, \dots, \mathbf{W}_{\text{tr}}^{(M)*} \mathbf{W}_{\text{tr}}^{(M)}\}. \end{aligned} \quad (14)$$

We get the above by noting that

$$\mathbb{E}[\mathbf{n}_c^{(i)}[k] \mathbf{n}_c^{(j)*}[k]] = \sigma^2 \mathbf{W}_{\text{tr}}^{(i)*} \mathbf{W}_{\text{tr}}^{(j)} \delta[i-j]. \quad (15)$$

We perform a Cholesky decomposition of \mathbf{C}_w to obtain $\mathbf{C}_w = \mathbf{D}_w^* \mathbf{D}_w$, where $\mathbf{D}_w \in \mathbb{C}^{ML_r \times ML_r}$ is upper triangular. Let us define \mathbf{D}_w^{-*} to denote the inverse of \mathbf{D}_w^* . Now, we multiply the RF combined received signal (12) by \mathbf{D}_w^{-*} to obtain the noise-whitened received signal:

$$\begin{aligned} \mathbf{y}_w[k] &= \mathbf{D}_w^{-*} \mathbf{y}[k] = \mathbf{D}_w^{-*} \tilde{\mathbf{\Phi}} \tilde{\mathbf{\Psi}} \mathbf{h}^v[k] + \mathbf{D}_w^{-*} \mathbf{n}_c[k] \\ &= \mathbf{\Upsilon}_w \mathbf{h}^v[k] + \mathbf{D}_w^{-*} \mathbf{n}_c[k], \end{aligned} \quad (16)$$

where $\mathbf{\Upsilon}_w = \mathbf{D}_w^{-*} \tilde{\mathbf{\Phi}} \tilde{\mathbf{\Psi}} \in \mathbb{C}^{ML_r \times G_t G_r}$. Concatenating the noise-whitened received signals of all the K subcarriers, we get

$$\mathbf{Y}_w = [\mathbf{y}_w[1] \quad \dots \quad \mathbf{y}_w[K]] = \mathbf{\Phi}_w \tilde{\mathbf{\Psi}} \mathbf{H}^v + \mathbf{N}_w, \quad (17)$$

where $\mathbf{Y}_w \in \mathbb{C}^{ML_r \times K}$, $\mathbf{\Phi}_w = \mathbf{D}_w^{-*} \tilde{\mathbf{\Phi}} \in \mathbb{C}^{ML_r \times N_t N_r}$, and $\mathbf{N}_w = \mathbf{D}_w^{-*} [\mathbf{n}_c[1] \quad \dots \quad \mathbf{n}_c[K]] \in \mathbb{C}^{ML_r \times K}$. Thus, we need to estimate the row sparse matrix \mathbf{H}^v given \mathbf{Y}_w and $\mathbf{\Phi}_w$.

3.2 Multi-level greedy search

We obtain an initial channel estimate using the MLGS procedure with a coarsely quantized beamspace dictionary. We adopt the Simultaneously Weighted Orthogonal Matching Pursuit (SW-OMP) algorithm as our base algorithm to form an initial estimate of the channel [4]. As the sparsifying dictionary Ψ is unknown a priori, we use row-truncated DFT matrices of size $N_t \times G_t$ and $N_r \times G_r$ as the Tx and Rx array steering matrices, respectively. Let $\tilde{\Psi}$ be the initial sparsifying dictionary.

In the first step of MLGS, we select a column from $\tilde{\Psi}$ that is maximally correlated with the received signal. Mathematically,

$$\hat{i} = \arg \max_i \sum_{k=1}^K \left| (\Phi_w \tilde{\Psi}[:, i])^* \mathbf{y}_w[k] \right|^2, \quad (18)$$

where $|\cdot|$ denotes an element-wise modulus operation, and $\tilde{\Psi}[:, i]$ is the i^{th} column of $\tilde{\Psi}$. Once we select \hat{i} , we extract AoD $\theta_{\hat{i}}$ and AoA $\phi_{\hat{i}}$ using the structure of $\tilde{\Psi}$, and form a finely spaced dictionary of range $(\theta_{\hat{i}} - \Delta\theta, \theta_{\hat{i}} + \Delta\theta)$ and $(\phi_{\hat{i}} - \Delta\phi, \phi_{\hat{i}} + \Delta\phi)$, where $\Delta\theta$ and $\Delta\phi$ are appropriately chosen based on the spatial quantization of the previously chosen dictionary. We repeat (18) with $\tilde{\Psi}$ replaced by the newly formed dictionary, and choose a new {AoD, AoA} pair. We repeat this process N times and select one set of AoD and AoA. Then, we compute

$$\hat{\mathbf{H}}^v = (\Phi_w \hat{\Psi})^\dagger \mathbf{Y}_w, \quad (19)$$

where $\hat{\Psi}$ is formed using the currently chosen AoD and AoA. This whole procedure constitutes the first out of S iterations of the MLGS algorithm in which we recover a single tap.

In the s^{th} iteration of MLGS, we recover s channel taps by following the same steps as above, but with the residual $\mathbf{Y}'_w = \mathbf{Y}_w - \Phi_w \hat{\Psi} \hat{\mathbf{H}}^v$ as observations, where $\hat{\Psi}$ comprises the set of {AoD, AoA} pairs chosen in the first $s - 1$ iterations. Therefore, after S iterations, we recover S virtual beamspace channel taps. We summarize MLGS as a flow diagram in Fig. 2.

3.3 Noise variance estimation

We estimate the noise variance $\hat{\sigma}_n^2$ using the residual output from MLGS. The noise variance is computed as

$$\hat{\sigma}_n^2 = \frac{1}{MKL_r} \|\mathbf{Y}'_w\|_F^2. \quad (20)$$

3.4 Sparse Bayesian learning

In this step, our goal is to refine the channel estimates output by the MLGS procedure. For convenience, we recall the measurement equation:

$$\mathbf{Y}_w = \Phi_w \hat{\Psi} \mathbf{H}^v + \mathbf{N}_w, \quad (21)$$

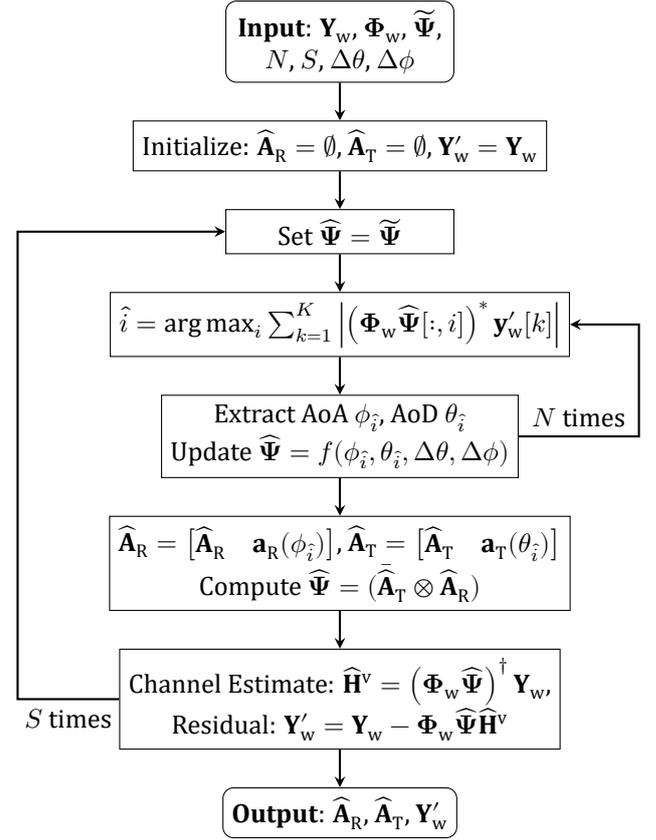


Fig. 2 - Flow diagram of MLGS.

where $\hat{\Psi} = (\hat{\mathbf{A}}_T \otimes \hat{\mathbf{A}}_R)$ is the dictionary output by MLGS. We adopt a statistical inference approach to infer the posterior distribution of \mathbf{H}^v given the measurements \mathbf{Y}_w , measurement matrix $\Phi_w \hat{\Psi}$, and noise variance $\hat{\sigma}_n^2$.

We use sparse Bayesian learning, a type-II maximum likelihood estimation procedure to obtain the channel estimate [26, 27]. In this method, we consider \mathbf{H}^v as a hidden variable, and obtain its posterior statistics given the observations. We impose a parameterized complex Gaussian prior on each column of the channel as $\mathcal{CN}(\mathbf{0}, \mathbf{\Gamma})$, where $\mathbf{\Gamma} = \text{diag}(\gamma)$. Using a common hyper-parameter γ across all the columns of \mathbf{H}^v aids in promoting common row sparsity in the solution. Now, we need to obtain the posterior distribution of \mathbf{H}^v , and the hyper-parameter γ . Since the prior and the noise are both Gaussian, obtaining the posterior statistics of \mathbf{H}^v is straightforward. But, computing γ requires computing the marginal probability distribution $p(\mathbf{Y}_w; \gamma)$ and maximizing it w.r.t. γ , which is called evidence maximization or type-II maximum likelihood estimation.

To solve this, we use the Expectation Maximization (EM) algorithm, which works by lower bounding the logarithm of the evidence $p(\mathbf{Y}_w; \gamma)$, and maximizing it iteratively. We treat \mathbf{H}^v as a hidden variable. In the expectation (E) step, we compute the expectation of the log likelihood of $(\mathbf{Y}_w, \mathbf{H}^v)$ w.r.t. $p(\mathbf{H}^v | \mathbf{Y}_w, \gamma)$. In the maximization (M) step, we compute the hyper-parameter γ by maximizing the

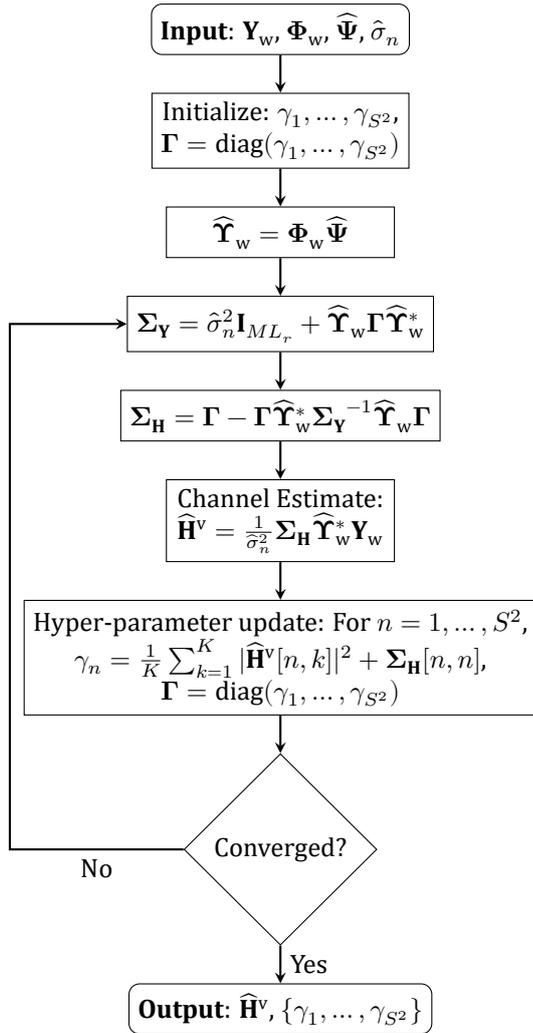


Fig. 3 – Flow diagram of MSBL.

function obtained in the E step. More details of SBL and type-II ML estimation can be found in [26, 28]. We provide a flow diagram of Multiple Measurement Vector SBL (MSBL) to compute the posterior mean and covariance of the channel, and the hyper-parameters, in Fig. 3. Specifically, in Fig. 3, the E-step of the EM algorithm corresponds to the computation of Σ_Y , Σ_H and $\hat{\mathbf{H}}^v$, and the M-step corresponds to the computation of Γ . We also elaborate on the E- and M-steps, albeit in the slightly different context of pattern-coupled sparse Bayesian learning, in Section 4.

Once we obtain the frequency domain channel estimate $\hat{\mathbf{H}}^v$, we estimate the support of the row sparse matrix and the channel coefficients using the hyper-parameters obtained using SBL. We estimate the noise variance using the Frobenius norm of the residual $\hat{\mathbf{Y}}_w = \mathbf{Y}_w - \Phi_w \hat{\Psi} \hat{\mathbf{H}}^v$.

3.5 Denoising

By analyzing the training dataset, we observed that the channel is sparse in both the virtual beamspace and delay domains. We exploited the beamspace sparsity and obtained the frequency domain channel estimates using MLGS and SBL. In this final step, we exploit the delay

domain sparsity to denoise the channel to further reduce the MSE between the original and estimated channels.

For each subcarrier k , we compute $(\hat{\mathbf{A}}_T \otimes \hat{\mathbf{A}}_R) \mathbf{H}^v[:, k]$, and reshape it to form k^{th} subcarrier's channel matrix of size $N_r \times N_t$. Then, for each transmit and receive antenna pair, we compute a K -point inverse DFT to obtain a delay-domain channel estimate. We retain the P dominant taps in the delay-domain channel estimate, and set the other $K - P$ taps to 0. We fix P based on the estimated noise variance, and the number of training frames M . The value of P is inversely proportional to $\hat{\sigma}_n^2$, and the training dataset is used to choose an appropriate P . From our experiments on the training dataset, we found that this denoising step leads to an approximately 2 dB reduction in NMSE.

This concludes the description of the MLGS-SBL approach, and we will describe the second approach in the next section.

4. PCSBL-DDT

In this section, we present another SBL based approach to the site-specific hybrid MIMO channel estimation problem. In this method, we adapt and extend the pattern-coupled SBL in [29] to our problem, by introducing sparsity connections (or couplings) between the consecutive AoAs and AoDs. We also impose a common sparsity model on the hyper-parameters such that all the delay taps share a common support. We will show that, together, these two innovations result in accurate channel estimates.

Recall that, in (12), the matrix $\Phi \in \mathbb{C}^{ML_r \times N_t N_r}$ is known, and we are given the received signals $\mathbf{y}[k]$ for $k = 1, \dots, K$. We use a fixed grid, although the grid points are different for training and testing stages. Hence, the dictionary matrix Ψ is also known in this method.

The lag-domain representation of the channel is of length K , with $N_c \ll K$ nonzero taps, which makes the channel sparse in the time-domain. Furthermore, the nonzero taps occur in clusters. To exploit the sparsity in the time-domain, we apply the pattern-coupled SBL algorithm on the time-domain signals. As a first step, we take the inverse DFT of the received signal sequence and scale it appropriately to keep the noise variance the same, i.e.,

$$\begin{aligned} \tilde{\mathbf{y}}[d] &= \frac{1}{\sqrt{K}} \left(\sum_{k=0}^{K-1} \mathbf{y}[k] \exp\left(\frac{j2\pi kd}{K}\right) \right) \\ &= \Phi \Psi \tilde{\mathbf{h}}^v[d] + \tilde{\mathbf{n}}_c[d], \quad d \in \mathcal{D}, \end{aligned} \quad (22)$$

where $\tilde{\mathbf{h}}^v[d] = \text{vec}(\Delta_d^v)$, and the noise $\tilde{\mathbf{n}}_c[d]$ has the same distribution as $\mathbf{n}_c[k]$. Here, $\mathcal{D} \subset \{0, \dots, K-1\}$ denotes the set of indices of the dominant delay taps. This set is determined heuristically by a simple threshold on the total energy of the received signals $\tilde{\mathbf{y}}[d]$, for $d = 0, \dots, K-1$.

This operation is done to increase the SNR by eliminating possibly all-noise samples.

As a next step, we apply a whitening filter as in the previous approach. The whitened time-domain signal is obtained similar to (16) as

$$\tilde{\mathbf{y}}_w[d] = \mathbf{D}_w^{-*} \tilde{\mathbf{y}}[d] = \mathbf{D}_w^{-*} \Phi \Psi \tilde{\mathbf{h}}^v[d] + \tilde{\mathbf{n}}_w[d], \quad (23)$$

where $\tilde{\mathbf{n}}_w[d] = \mathbf{D}_w^{-*} \tilde{\mathbf{n}}_c[d] \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_{ML_r})$.

Note that the following approach is first applied to the true channels from the training data by regularizing it with a very small variance white Gaussian noise and uniform grids for AoAs and AoDs. Then, in the testing stage, the grid points are refined based on the joint AoA/AoD pattern that is extracted from the training data. Since the sparse model and overall procedure is the same in the training and testing phases except for the measurement matrices (there is an additional matrix $\mathbf{D}_w^{-*} \Phi$ multiplying the true channels from the left in testing stage in (23)), we directly present the method used in the testing stage. The channel estimator in both phases operates on the received signals $\tilde{\mathbf{y}}_w[d]$, for $d \in \mathcal{D}$.

The pattern-coupled SBL method in [29] assumes noisy measurements of the form of

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \quad (24)$$

where \mathbf{y} is the observed vector, \mathbf{A} is the measurement matrix, and the \mathbf{x} is the sparse signal with some unknown block-sparsity patterns. The vector \mathbf{n} is the zero-mean Gaussian noise with scaled identity covariance matrix. Hence, the model is in accordance with the one in (23). Let us define $\mathbf{A} = \mathbf{D}_w^{-*} \Phi \Psi$, $\mathbf{y}^d = \tilde{\mathbf{y}}_w[d]$, $\mathbf{x}^d = \tilde{\mathbf{h}}^v[d]$ and $\mathbf{n}^d = \tilde{\mathbf{n}}_w[d]$. Then, we have all the measurements from (23) for $d \in \mathcal{D}$ in the form

$$\mathbf{y}^d = \mathbf{A}\mathbf{x}^d + \mathbf{n}^d, \quad d \in \mathcal{D}. \quad (25)$$

Let us express the sparse vector $\mathbf{x}^d \in \mathbb{C}^{G_r G_t}$ in the following form with special indices:

$$\mathbf{x}^d = \begin{bmatrix} x_{1,1}^d \\ \vdots \\ x_{G_r,1}^d \\ x_{1,2}^d \\ \vdots \\ x_{G_r,2}^d \\ \vdots \\ x_{1,G_t}^d \\ \vdots \\ x_{G_r,G_t}^d \end{bmatrix}, \quad d \in \mathcal{D}. \quad (26)$$

Note that the elements of \mathbf{n}^d are independent and identically distributed zero-mean complex Gaussian random variables with variance σ^2 .

4.1 Proposed pattern-coupled hierarchical model

To exploit both the block-sparse structure along AoAs, AoDs, and the common sparsity for all the delay taps, we define a prior over $\mathbf{x} \triangleq \{\mathbf{x}^d : d \in \mathcal{D}\}$ as

$$p(\mathbf{x}|\alpha) = \prod_{g_r=1}^{G_r} \prod_{g_t=1}^{G_t} \prod_{d \in \mathcal{D}} \mathcal{CN}(x_{g_r,g_t}^d | 0, \eta_{g_r,g_t}^{-1}). \quad (27)$$

To model the pattern-coupled block sparsity, we express the common parameter η_{g_r,g_t} among the delay taps as

$$\eta_{g_r,g_t} = \alpha_{g_r,g_t} + \beta_r \alpha_{g_r-1,g_t} + \beta_r \alpha_{g_r+1,g_t} + \beta_t \alpha_{g_r,g_t-1} + \beta_t \alpha_{g_r,g_t+1}, \quad (28)$$

where $\alpha = \{\alpha_{g_r,g_t}\}$ are the hyper-parameters controlling the sparsity of \mathbf{x} . The parameters $\beta_r \in [0, 1]$ and $\beta_t \in [0, 1]$ indicate the pattern relevance between x_{g_r,g_t}^d and its neighboring coefficients and they are taken as known constants in accordance with the related works. Different from [29], we do not impose any Gamma prior for the hyper-parameters $\{\alpha_{g_r,g_t}\}$. Instead, we consider these hyper-parameters to be deterministic and unknown, which is equivalent to assuming a non-informative prior. In our experiments, we find that this approach works better than imposing the Gamma prior.

Note that in the testing stage, the noise variance is not given explicitly. Instead a range information is provided. So, we assume that we do not know $\gamma = 1/\sigma^2$, but we introduce a uniform prior on γ , i.e., $\gamma \sim \mathcal{U}[\gamma_{\text{low}}, \gamma_{\text{upp}}]$ where the bounds are provided along with the test data. This assumption also differs from the Gamma distribution that is considered in [29].

We utilize an EM algorithm for learning the sparse signal \mathbf{x} and the hyper-parameters $\Theta \triangleq \{\alpha, \gamma\}$. In the EM formulation, the signal \mathbf{x} is treated as a hidden variable, and we iteratively maximize a lower bound on the posterior probability $p(\Theta|\mathbf{y})$ (this lower bound is also referred to as the Q-function). The algorithm alternates between an E-step and an M-step. We explain these two steps below.

4.2 E-Step

In the E-step, we need to compute the posterior distribution of \mathbf{x} conditioned on the observed data and the hyper-parameters estimated from the s^{th} iteration, i.e.,

$$p(\mathbf{x}|\mathbf{y}, \Theta^{(s)}) \propto p(\mathbf{x}|\alpha^{(s)}) p(\mathbf{y}|\mathbf{x}, \gamma^{(s)}). \quad (29)$$

The posterior probability can be computed as a multivariate Gaussian distribution with mean and covariance matrix for \mathbf{x}^d as

$$\boldsymbol{\mu}^{d(s)} = \gamma^{(s)} (\gamma^{(s)} \mathbf{A}^* \mathbf{A} + \mathbf{D}^{(s)})^{-1} \mathbf{A}^* \mathbf{y}^d, \quad d \in \mathcal{D} \quad (30)$$

$$\boldsymbol{\chi}^{d(s)} = (\gamma^{(s)} \mathbf{A}^* \mathbf{A} + \mathbf{D}^{(s)})^{-1}, \quad d \in \mathcal{D} \quad (31)$$

from [29] where $\mathbf{D}^{(s)} \in \mathbb{R}^{G_r G_t \times G_r G_t}$ is a diagonal matrix with the diagonal elements $\eta_{g_r, g_t}^{(s)}$ that are ordered according to the indexing in (26). Let μ_{g_r, g_t}^d and χ_{g_r, g_t}^d denote the elements of $\boldsymbol{\mu}^d$ and $\boldsymbol{\chi}^d$ corresponding to the index ordering in (26).

4.3 M-Step

In the M-step, the hyper-parameters $\Theta = \{\boldsymbol{\alpha}, \gamma\}$ are estimated by treating \mathbf{x} as hidden variables and iteratively maximizing the Q-function, i.e.,

$$\begin{aligned} \Theta^{(s+1)} &= \arg \max_{\Theta} Q(\Theta | \Theta^{(s)}) \\ &= \arg \max_{\Theta} \mathbb{E}_{\mathbf{x} | \mathbf{y}, \Theta^{(s)}} [\ln p(\Theta | \mathbf{x}, \mathbf{y})], \end{aligned} \quad (32)$$

where the expectation is with respect to the posterior distribution $p(\mathbf{x} | \mathbf{y}, \Theta^{(s)})$. We can express the above maximization with respect to Θ as

$$\begin{aligned} &\text{maximize}_{\Theta} \mathbb{E}_{\mathbf{x} | \mathbf{y}, \Theta^{(s)}} [\ln p(\boldsymbol{\alpha}) p(\mathbf{x} | \boldsymbol{\alpha})] \\ &\quad + \mathbb{E}_{\mathbf{x} | \mathbf{y}, \Theta^{(s)}} [\ln p(\boldsymbol{\gamma}) p(\boldsymbol{\gamma})]. \end{aligned} \quad (33)$$

We can implement the iterative updates in an alternating manner as follows:

1) Update for $\boldsymbol{\alpha}$:

Following a similar approach in [29], we can obtain a sub-optimal update for $\boldsymbol{\alpha}$ as (the optimal update is not available in closed form due to the coupled variables):

$$\begin{aligned} \alpha_{g_r, g_t}^d{}^{(s+1)} &= \frac{|\mathcal{D}|}{\omega_{g_r, g_t}^{(s)}}, \quad d \in \mathcal{D}, \quad g_r = 1, \dots, G_r, \\ &\quad g_t = 1, \dots, G_t, \end{aligned} \quad (34)$$

where

$$\begin{aligned} \omega_{g_r, g_t}^{(s)} &= \sum_{d \in \mathcal{D}} \left(\left| \mu_{g_r, g_t}^d{}^{(s)} \right|^2 + \chi_{g_r, g_t}^d{}^{(s)} \right) \\ &\quad + \beta_r \left(\left| \mu_{g_r-1, g_t}^d{}^{(s)} \right|^2 + \chi_{g_r-1, g_t}^d{}^{(s)} \right) \\ &\quad + \beta_r \left(\left| \mu_{g_r+1, g_t}^d{}^{(s)} \right|^2 + \chi_{g_r+1, g_t}^d{}^{(s)} \right) \\ &\quad + \beta_t \left(\left| \mu_{g_r, g_t-1}^d{}^{(s)} \right|^2 + \chi_{g_r, g_t-1}^d{}^{(s)} \right) \\ &\quad + \beta_t \left(\left| \mu_{g_r, g_t+1}^d{}^{(s)} \right|^2 + \chi_{g_r, g_t+1}^d{}^{(s)} \right), \\ &\quad g_r = 1, \dots, G_r, \quad g_t = 1, \dots, G_t. \end{aligned} \quad (35)$$

2) Update for γ :

The hyper-parameter γ , which is the inverse of the noise variance and has a uniform prior distribution on $[\gamma_{\text{low}}, \gamma_{\text{upp}}]$ can be updated by adapting the derivation in [30] to the uniform prior considered here, as follows:

$$\gamma^{(s+1)} = \arg \max_{\gamma} \mathbb{E}_{\mathbf{z} | \mathbf{y}, \Theta^{(s)}} [\ln p(\gamma) p(\mathbf{z} | \mathbf{y}, \gamma)]. \quad (36)$$

Algorithm 1 EM Algorithm for the Sparse Estimation of \mathbf{x}

Input: The set of indices of the dominating delay taps: \mathcal{D} . The measurement matrix: \mathbf{A} . The noisy measurement vectors: \mathbf{y}^d , $d \in \mathcal{D}$. The pattern relevance parameters: β_r and β_t . The solution accuracy: ϵ_{EM} . The minimum and maximum iteration numbers: s_{min} and s_{max} . Initial hyper-parameters: $\Theta^{(0)} = \{\boldsymbol{\alpha}^{(0)}, \gamma^{(0)}\}$. The lower and upper bounds for γ : γ_{low} and γ_{upp} . Initialize the iteration index $s \leftarrow 0$.

1: repeat

- 2: Compute $\{\eta_{g_r, g_t}^{(s)}\}$ according to (28) using $\boldsymbol{\alpha}^{(s)}$.
- 3: Update $\boldsymbol{\mu}^d$ and $\boldsymbol{\chi}^d$, for $d \in \mathcal{D}$ according to (30)-(31) using $\{\eta_{g_r, g_t}^{(s)}\}$ and $\gamma^{(s)}$.
- 4: Update $\boldsymbol{\alpha}^{(s+1)}$ and $\gamma^{(s+1)}$ according to (34) and (37) using $\boldsymbol{\mu}^d$ and $\boldsymbol{\chi}^d$, for $d \in \mathcal{D}$.
- 5: Set $s \leftarrow s + 1$.
- 6: **until** $s = s_{\text{max}}$ or $s \geq s_{\text{min}}$ with

$$\frac{\sum_{d \in \mathcal{D}} \left\| \boldsymbol{\mu}^d{}^{(s-1)} - \boldsymbol{\mu}^d{}^{(s-2)} \right\|^2}{\sum_{d \in \mathcal{D}} \left\| \boldsymbol{\mu}^d{}^{(s-1)} \right\|^2} \leq \epsilon_{\text{EM}}. \quad (39)$$

Output: $\hat{\mathbf{x}}^d = \boldsymbol{\mu}^d{}^{(s-1)}$, for $d \in \mathcal{D}$.

Using the uniform prior, we can obtain $\gamma^{(s+1)}$ as (37) at the top of the next page, where

$$\Pi_{\gamma}(x) = \begin{cases} \gamma_{\text{low}} & \text{if } x \leq \gamma_{\text{low}} \\ x & \text{if } \gamma_{\text{low}} < x \leq \gamma_{\text{upp}} \\ \gamma_{\text{upp}} & \text{if } x > \gamma_{\text{upp}} \end{cases}. \quad (38)$$

The overall EM algorithm is implemented by applying the updates iteratively until the difference between $\boldsymbol{\mu}^d{}^{(s)}$ and $\boldsymbol{\mu}^d{}^{(s-1)}$ is negligible. At the final iteration, the sparse vector estimate $\hat{\mathbf{x}}^d$ is set to $\boldsymbol{\mu}^d{}^{(s)}$, for $d \in \mathcal{D}$. The overall algorithm is summarized in Algorithm 1. After multiplying $\hat{\mathbf{x}}^d$ with the dictionary matrix $\boldsymbol{\Psi}$, we obtain the time-domain channel estimates at the dominant delay taps in \mathcal{D} . Then, we take the K -point DFT of the time channels and scale them by $1/\sqrt{K}$ to obtain the final frequency channel estimates.

We describe the overall method in the next section in more detail.

4.4 Learning the joint relations between AoAs and AoDs

As a first step, we construct the dictionary matrix $\boldsymbol{\Psi}$ by $G_r = 96$ AoA and $G_t = 24$ AoD grid points that are uniformly selected from $[0, \pi]$. We only consider this angle range since the array steering vectors for the other angles are the same as those with the angles in $[0, \pi]$. Then using 10,000 true frequency channels provided in the training data set, we add a white Gaussian complex noise to the time-domain channels to obtain the sparse model

$$\gamma^{(s+1)} = \Pi_\gamma \left(\frac{ML_r|\mathcal{D}|}{\sum_{d \in \mathcal{D}} \left(\|\mathbf{y}^d - \mathbf{A}\boldsymbol{\mu}^{d(s)}\|^2 + (\gamma^{(s)})^{-1} (G_r G_t - \text{tr}(\boldsymbol{\chi}^{d(s)} \mathbf{D}^{(s)})) \right)} \right) \quad (37)$$

$$\mathbf{y}_{\text{training}}^d = \boldsymbol{\Psi} \mathbf{x}_{\text{training}}^d + \mathbf{n}_{\text{training}}^d. \quad (40)$$

Note that the variance of the noise is selected as a very small value, e.g., 10^{-4} . The motivation is to regularize the model and apply the EM algorithm described previously without any numerical issues. We apply the EM algorithm in the previous section by keeping the inverse noise variance $\gamma = 10^4$ fixed in all the 10,000 models obtained from the training dataset. Then, using all the sparse estimates $\hat{\mathbf{x}}_{\text{training}}^d$, we estimate the power distribution along $2G_r = 192$ AoA points and $2G_t = 48$ AoD points as in Fig. 4. Here, we apply a linear interpolation to both the AoA and AoD axes since we will utilize this in the grid construction algorithm in the testing stage. As Fig. 4 shows, some AoA/AoD grid points are more probable for the given simulation site. To exploit this learned information, we propose a grid construction algorithm, i.e., Algorithm 2, to locate the grid points more densely in the yellow regions compared to the blue regions.

We first start with a uniform grid for both AoA and AoD in $[0, \pi]$ with $96 \cdot 24$ points in total. Then, we assign additional $96 \cdot 8$ grid points to the most yellow regions in Fig. 4 by sorting the power values in decreasing order. In the next stage, we change the locations of the points to move them to the places where the power of the sparse vectors obtained from the training data is greater. At the same time, we try to prevent the neighboring grid points from being far away via judicious tuning and adjustments. For this, we consider six different distance thresholds that correspond to the maximum allowable distance between two consecutive grid points in horizontal and vertical directions. If the logarithm of the mean power value at a particular grid point is high, then a smaller (more restrictive) distance threshold is used. The motivation behind using logarithm is that the power differences across the AoA/AoD grid points are observed to be more emphasized after applying logarithm operation. In the end, the constructed grid point map is shown in Fig. 5 where the yellow points denote the selected $96 \cdot 32$ grid points to be utilized in constructing the dictionary matrix in the testing stage. Note that the minimum distance threshold value in the vector \mathbf{d} is two instead of one since there is already an interpolation by a factor of two. The number of power levels, i.e., six, is chosen heuristically.

In the testing stage, after constructing the dictionary matrix $\boldsymbol{\Psi}$ according to the pattern in Fig. 5, we also modify the pattern-coupling relations accordingly. For this new grid structure, the AoA and AoD pattern-coupled block sparsity relations in (28) and (35) are modified such that

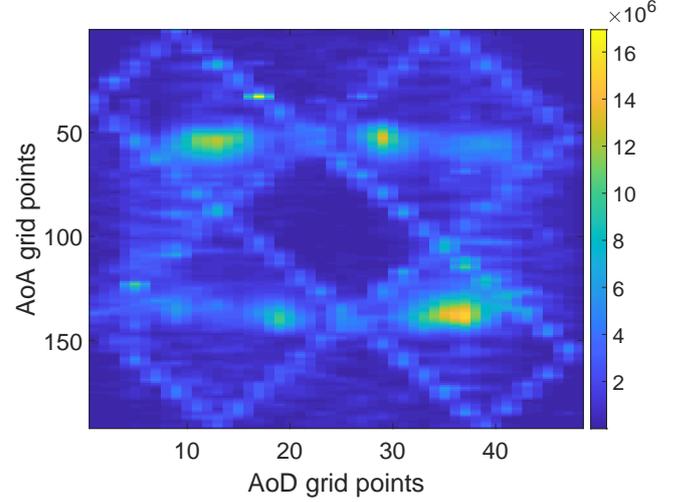


Fig. 4 – Heatmap for the power distribution of the sparse vector among AoA and AoD grid points.

the consecutive AoA and AoD grid points in Fig. 5 are constructed as coupled by keeping only the pairs with some distance threshold, i.e., not being far away more than two grid points. The updates in the EM algorithm are the same except for the indices according to the pattern-coupled block sparsity pattern.

This concludes the description of the PCSBL-DDT algorithm, and we will describe the third and last approach in the next section.

5. PC-OMP

In this section, we present the Projection Cuts Orthogonal Matching Pursuit (PC-OMP) approach for the site-specific hybrid MIMO channel estimation problem. This method makes use of the sparsity of the mm-wave channel and extracts paths parameters one by one using an OMP algorithm. A novel, custom detection method is used to detect paths, which is optimized using training data.

We express the frequency-domain channel at the k^{th} sub-carrier in (5) as

$$\mathbf{H}[k] = \sum_{\ell=1}^L \tilde{\alpha}_\ell \exp(-j2\pi\tau_\ell k) \mathbf{a}_R(\phi_\ell) \mathbf{a}_T^*(\theta_\ell), \quad (41)$$

where the effect of pulse shaping and other scaling factors except the delay of the ℓ^{th} path, i.e., τ_ℓ in (3), are embedded into $\tilde{\alpha}_\ell$. Using the identity $\text{vec}(\mathbf{a}\mathbf{b}^T) = \mathbf{b} \otimes \mathbf{a}$, $\mathbf{H}[k]$ vectorizes into

$$\mathbf{h}_k = \sum_{\ell=1}^L \tilde{\alpha}_\ell \exp(-j2\pi\tau_\ell k) \bar{\mathbf{a}}_T(\theta_\ell) \otimes \mathbf{a}_R(\phi_\ell) \quad (42)$$

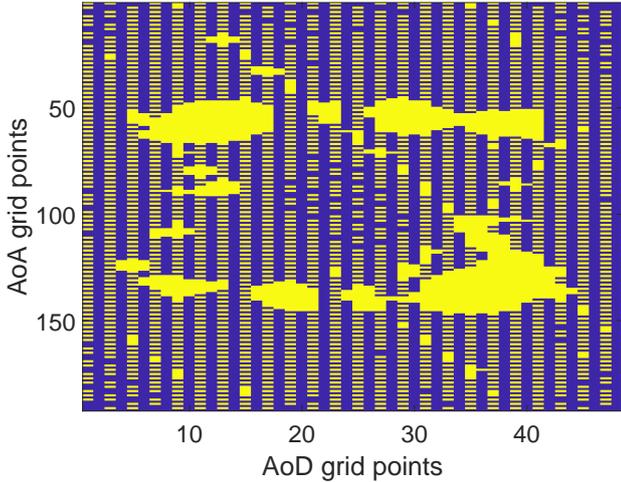


Fig. 5 – Non-uniform grid pattern for AoA and AoD in testing stage of the algorithm. The yellow pixels correspond to the selected $96 \cdot 32$ grid points.

and can be horizontally stacked into the matrix from (13) as

$$\hat{\mathbf{H}} = \Psi \mathbf{H}^v = \sum_{\ell=1}^L \tilde{\alpha}_{\ell} (\bar{\mathbf{a}}_T(\theta_{\ell}) \otimes \mathbf{a}_R(\phi_{\ell})) \mathbf{a}_F^T(\tau_{\ell}) \quad (43)$$

with $[\mathbf{a}_F(\tau_{\ell})]_k = \exp(-j2\pi\tau_{\ell}k)$. Then the stacked measurements in (13) can be expressed as

$$\mathbf{Y} = \Phi \hat{\mathbf{H}} + \mathbf{N}_c. \quad (44)$$

To obtain independent and identically distributed noise entries, we apply the whitening given in (17), i.e.,

$$\mathbf{Y}_w = \Phi_w \hat{\mathbf{H}} + \mathbf{N}_w. \quad (45)$$

To ease the notation, we will define the spatial component of a path as

$$\mathbf{a}_{R-T}(\phi_{\ell}, \theta_{\ell}) = \bar{\mathbf{a}}_T(\theta_{\ell}) \otimes \mathbf{a}_R(\phi_{\ell}) \quad (46)$$

and the channel component of a path as

$$\mathbf{a}_{R-T-F}(\phi_{\ell}, \theta_{\ell}, \tau_{\ell}) = \mathbf{a}_F(\tau_{\ell}) \otimes \mathbf{a}_{R-T}(\phi_{\ell}, \theta_{\ell}). \quad (47)$$

5.1 Approach

Our strategy capitalizes on the fact that $\hat{\mathbf{H}}$ is a very sparse matrix in the sense that the number of paths L is much smaller than the maximum matrix rank $\min(N_r N_t, K)$. Hence, the path components $\mathbf{a}_F(\tau_{\ell}) \otimes \bar{\mathbf{a}}_T(\theta_{\ell}) \otimes \mathbf{a}_R(\phi_{\ell})$ are nearly orthogonal to each other, i.e.,

$$\langle \mathbf{a}_{R-T-F}(\phi_{\ell}, \theta_{\ell}, \tau_{\ell}), \mathbf{a}_{R-T-F}(\phi_{\ell'}, \theta_{\ell'}, \tau_{\ell'}) \rangle \approx 0, \forall \ell \neq \ell'. \quad (48)$$

Our algorithm consists of extracting the path parameters $(\phi_{\ell}, \theta_{\ell}, \tau_{\ell})$ one by one and then subtracting their contribution in an OMP algorithm. Two key aspects here are

Algorithm 2 Grid Construction Algorithm Using the Power Distribution of the Sparse Vectors

Input: The interpolated power distribution of $\{\hat{\mathbf{x}}_{\text{training}}^d\}$ along $2G_r = 192$ AoA points and $2G_t = 48$ AoD points. Set $\mathcal{P}_{\min} = 0$ and $\mathcal{P}_{\max} = \infty$. Construct the vector of logarithms of the six power levels that are equally spaced between the logarithms of minimum and maximum power value in the AoA/AoD power distribution. Denote this vector by \mathbf{p} . Construct the corresponding distance threshold vector $\mathbf{d} = [7 \ 6 \ 5 \ 4 \ 3 \ 2]^T$.

- 1: Select the points in the uniform grid for both AoA and AoD in $[0, \pi]$ with $96 \cdot 24$ points in total.
- 2: Select additional $96 \cdot 8$ grid points with the greatest power values.
- 3: **repeat**
- 4: For each grid point, compute the mean power of the three consecutive vertical and horizontal points with the considered point being at the center. Denote this mean power by p_{g_r, g_t} for grid point (g_r, g_t) .
- 5: Count the number of entries in \mathbf{p} which are less than or equal to the logarithm of p_{g_r, g_t} for each grid point (g_r, g_t) . Set the corresponding distance threshold d_{g_r, g_t} as the element of \mathbf{d} at this index, i.e., the number of entries.
- 6: Update \mathcal{P}_{\max} by the maximum of the mean values computed above among the non-selected grid points. Set the corresponding grid point as a candidate for inclusion.
- 7: Update \mathcal{P}_{\min} by the minimum of the mean values computed above among the selected grid points whose removal will not alter the maximum allowable distance d_{g_r, g_t} between consecutive horizontal and vertical selected points. Set the corresponding grid point as a candidate for removal.
- 8: Remove the grid point found in Step 7 from the grid pattern and add the grid point found in Step 6 to the grid pattern.
- 9: **until** $\mathcal{P}_{\min} > \mathcal{P}_{\max}$

Output: The updated non-uniform grid pattern

(a) how to set the dictionary for determining the path parameters and (b) how to know when to stop the OMP iterations. Key to the success of our algorithm lies in the way we address these two aspects, i.e., how we search for path parameters by using projections and how we detect the presence of a new path. We describe these in the sequel.

At each step of OMP, we want to obtain the best matching channel component, i.e., we want to solve

$$\max_{\phi_{\ell}, \theta_{\ell}, \tau_{\ell}} |\langle \text{vec}(\Phi \mathbf{a}_{R-T}(\phi_{\ell}, \theta_{\ell}) \mathbf{a}_F^T(\tau_{\ell})), \text{vec}(\mathbf{Y}_w) \rangle| \quad (49)$$

which can be simplified into

$$\max_{\phi_{\ell}, \theta_{\ell}, \tau_{\ell}} |\mathbf{a}_{R-T}^*(\phi_{\ell}, \theta_{\ell}) \Phi^* \mathbf{Y}_w \bar{\mathbf{a}}_F(\tau_{\ell})|, \quad (50)$$

where we have used $\langle \text{vec}(\mathbf{A}), \text{vec}(\mathbf{B}) \rangle = \text{tr}(\mathbf{A}^* \mathbf{B})$ and the cyclic shift property of the trace.

The above maximization problem can be solved by searching over a discretized set of values of the path parameters, $(\phi_\ell, \theta_\ell, \tau_\ell)$. We start by considering a small set of values for the path parameters equally spaced in their domains. We choose a resolution of $4K$ values for τ_ℓ and of $N_t/2$ ($N_t/2$) values for ϕ_ℓ (θ_ℓ). Since we are considering a smaller resolution for the angles than that required for them to cover the entire angular spectrum, we substitute each $\mathbf{a}_R(\phi_\ell)$ by a sector beam-pattern $\hat{\mathbf{a}}_R(\phi_\ell)$ of width $4\pi/N_t$. The same manipulation applies to the phase angles at the transmitter, and thus we can define $\hat{\mathbf{a}}_{R-T}(\phi_\ell, \theta_\ell) = \hat{\mathbf{a}}_T(\theta_\ell) \otimes \hat{\mathbf{a}}_R(\phi_\ell)$. The sector beam-pattern we consider is the same as the one defined in [31]. With this definition, we can extract a coarse version of the path parameters $(\phi_\ell, \theta_\ell, \tau_\ell)$ by maximizing

$$d_{\phi_\ell, \theta_\ell, \tau_\ell} = |\hat{\mathbf{a}}_{R-T}^*(\phi_\ell, \theta_\ell) \Phi^* \mathbf{Y}_w \bar{\mathbf{a}}_F(\tau_\ell)|. \quad (51)$$

5.2 Detection

Now we want to know if those parameters can be considered as a path detection. To this end, we take into account the null hypothesis of $\mathbf{H} = \mathbf{0}$, in that case all elements of \mathbf{Y}_w are independent white noise and thus $\hat{\mathbf{a}}_{R-T}^*(\phi_\ell, \theta_\ell) \Phi^* \mathbf{Y}_w \bar{\mathbf{a}}_F(\tau_\ell)$ is also comprised of white noise components. Consequently, $d_{\phi_\ell, \theta_\ell, \tau_\ell}$ follows a Rayleigh distribution, which has the cumulative distribution function

$$F(d_{\phi_\ell, \theta_\ell, \tau_\ell}) = 1 - \exp\left(-\frac{d_{\phi_\ell, \theta_\ell, \tau_\ell}^2}{2\sigma^2}\right). \quad (52)$$

Since there is a channel contribution only for a small number of path parameters, we have that the *median* of all computed values of $d_{\phi_\ell, \theta_\ell, \tau_\ell}$ should be close to that of the Rayleigh distribution $\sigma\sqrt{2\ln(2)}$. This insight is key to our algorithm. Then, σ can be approximated as

$$\sigma \simeq \mu(d_{\phi_\ell, \theta_\ell, \tau_\ell}) / \sqrt{2\ln(2)}. \quad (53)$$

In this case, the cumulative function of $\max x_k$ can be computed as $F_{\max}(x) = \prod F_k(x)$. Explicitly, it is given by

$$F_{\max}(\max(d_{\phi_\ell, \theta_\ell, \tau_\ell})) = \left(1 - \exp\left(-\frac{\max(d_{\phi_\ell, \theta_\ell, \tau_\ell})^2}{2\sigma^2}\right)\right)^{N_t N_t K}. \quad (54)$$

Using this, we can compute a detection threshold corresponding to a confidence level of δ as

$$\mu(d_{\phi_\ell, \theta_\ell, \tau_\ell}) \sqrt{-\log_2(1 - (\delta)^{\frac{1}{N_t N_t K}})}. \quad (55)$$

We compare the optimal value of $d_{\phi_\ell, \theta_\ell, \tau_\ell}$ obtained by solving (51) with the above threshold to decide whether

the path is sufficiently significant to be included in the model, or whether to stop the OMP iterations. The value of δ is optimized using the dataset information as described later in Section 6.3.

5.3 Refinement

Once a path has been detected, we proceed to refine the path components by iterative projections. We do this by freezing two of the variables and increasing the resolution of the third, in an alternating fashion.

First steps: First, we adapt our estimation to handle a higher resolution due to the manipulation we did with the angular resolution.

We start by increasing the time resolution by computing the maximum of $|\hat{\mathbf{a}}_{R-T}^*(\phi_\ell, \theta_\ell) \Phi^* \mathbf{Y}_w \bar{\mathbf{a}}_F(\tau_\ell)|$ for fixed (ϕ_ℓ, θ_ℓ) and τ_ℓ with a resolution of $32K$ equally spaced points.

Then, by fixing τ_ℓ and using the identity $\text{vec}(\mathbf{A}^* \mathbf{B} \mathbf{C}) = (\bar{\mathbf{C}} \otimes \mathbf{A})^* \text{vec}(\mathbf{B})$, we can simplify the expression to

$$|\hat{\mathbf{a}}_R^*(\phi_\ell) \bar{\mathbf{H}}(\tau_\ell) \hat{\mathbf{a}}_T(\theta_\ell)| \quad (56)$$

with $\bar{\mathbf{H}}(\tau_\ell)$ such that $\text{vec}(\bar{\mathbf{H}}(\tau_\ell)) = \Phi^* \mathbf{Y}_w \bar{\mathbf{a}}_F(\tau_\ell)$.

We then proceed to refine the angle components with the highest number of antennas. For simplicity, let us assume that $N_t > N_r$. By increasing the resolution of θ_ℓ to $32N_t$ equally spaced points, we do not need to use the sector beam-pattern manipulation, thus we can simply maximize

$$|\hat{\mathbf{a}}_R^*(\phi_\ell) \bar{\mathbf{H}}(\tau_\ell) \mathbf{a}_T(\theta_\ell)| \quad (57)$$

over θ_ℓ while the other path parameters are fixed.

Finally, we refine the expression with respect to the remaining angle. Again, the manipulation is not required, and we can maximize

$$|\mathbf{a}_R^*(\phi_\ell) \bar{\mathbf{H}}(\tau_\ell) \mathbf{a}_T(\theta_\ell)| \quad (58)$$

over θ_ℓ while the other path parameters are fixed.

Iteration steps: Now that with the first steps we removed the angle uncertainty caused by the sector beam-pattern, we can proceed to repeat the same steps iteratively by substituting all sector beam-patterns $\hat{\mathbf{a}}$ by array responses \mathbf{a} .

Once the parameters of the path have been estimated, $(\phi_\ell, \theta_\ell, \tau_\ell)$, we use them to reconstruct the path and subtract it from the received pilots, thereby altering the residual. Then, the residual is updated, and the next path is obtained following the same steps. The residual is updated until a stopping condition is reached, as discussed in Section 5.2. How to obtain the best stopping condition for our algorithm is discussed in Section 6.3.

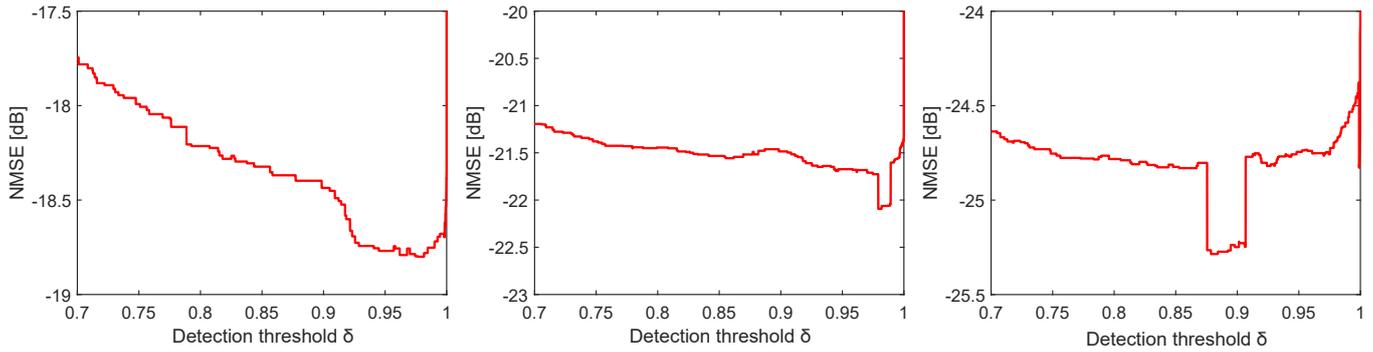


Fig. 6 – NMSE behavior over the decision threshold of the 3 training datasets.

Table 1 – NMSE table for training data

SNR (dB)	Algorithm	-15	-10	-5
Pilot Frames: 20	SW-OMP	-1.45 dB	-5.70 dB	-9.68 dB
	MLGS-SBL	-4.29 dB	-9.13 dB	-12.34 dB
	PCSBL-DDT	-8.16 dB	-10.62 dB	-11.07 dB
	PC-OMP	-8.34 dB	-12.36 dB	-16.15 dB
Pilot Frames: 40	SW-OMP	-3.95 dB	-7.95 dB	-11.87 dB
	MLGS-SBL	-7.55 dB	-11.19 dB	-14.15 dB
	PCSBL-DDT	-10.56 dB	-12.14 dB	-12.62 dB
	PC-OMP	-12.66 dB	-16.33 dB	-19.78 dB
Pilot Frames: 80	SW-OMP	-7.33 dB	-11.60 dB	-15.63 dB
	MLGS-SBL	-13.02 dB	-16.37 dB	-18.94 dB
	PCSBL-DDT	-11.90 dB	-13.10 dB	-13.63 dB
	PC-OMP	-18.70 dB	-21.49 dB	-24.48 dB

Table 2 – NMSE table for test data

SNR (dB)	Algorithm	[-20, -11)	[-11, -6)	[-6, 0]
Pilot Frames: 20	MLGS-SBL	-7.66 dB	-10.97 dB	-12.34 dB
	PCSBL-DDT	-8.94 dB	-9.99 dB	-10.31 dB
	PC-OMP	-9.09 dB	-12.45 dB	-14.22 dB
Pilot Frames: 40	MLGS-SBL	-11.87 dB	-12.79 dB	-14.20 dB
	PCSBL-DDT	-10.82 dB	-11.33 dB	-11.89 dB
	PC-OMP	-13.79 dB	-15.24 dB	-16.79 dB
Pilot Frames: 80	MLGS-SBL	-13.62 dB	-16.23 dB	-20.08 dB
	PCSBL-DDT	-11.74 dB	-12.47 dB	-12.98 dB
	PC-OMP	-16.32 dB	-19.07 dB	-23.91 dB

6. NUMERICAL RESULTS

In this section, we discuss the NMSE performance of our proposed algorithms with the training and testing data generated using *Raymobtime*, a ray tracing based mmWave channel generation tool. We train the mmWave channel estimation algorithms using 10,000 independent channel realizations, each consisting of 100 paths between the Tx and Rx. More details about the channel generation methodology can be found in [24]. We used 20, 40, and 80 pilot frames during both the training and testing phases of the proposed algorithms. For the training phase, we used SNR values of $\{-15, -10, -5\}$ dB. We benchmark the NMSE performance of our proposed algorithms with a reference state-of-the-art model-based greedy search algorithm called SW-OMP [4].

We note that while the three new algorithms presented in this paper have been fine-tuned based on the training dataset, the baseline algorithm, SW-OMP, has been implemented as-is from the literature. On the other hand, in our implementation of SW-OMP, we consider the case where the true AoDs and AoAs are contained in the sparsifying dictionary. While the proposed algorithms do suffer from the off-grid effects, the SW-OMP algorithm is insulated from the performance degradation caused by them.

6.1 MLGS-SBL

Upon analyzing the training channels, we set the maximum number of paths obtained from MLGS to $S = 10$, and the number of levels in MLGS to $N = 5$. We set the number of columns in the initial AoD/AoA steering matrices to 256. We use the estimated noise variance after SBL to threshold the number of dominant delay taps of the channel denoiser. As this approach is primarily a model-based method, and uses few statistics from the training data, it is suitable for general mmWave channel estimation problems also. Further, the thresholds are set keeping in mind the computational complexity of the MLGS-SBL algorithm. By increasing the number of paths output by MLGS, we can potentially improve the performance of the algorithm, but at the cost of higher computational complexity. We include the NMSE values obtained for the training and testing datasets in Table 1 and Table 2, respectively. The final performance score achieved, which is a weighted combination of the NMSE performance in Table 2 when the number of pilot frames is 20, using our proposed algorithm in the mmWave channel estimation challenge is -9.16 dB.

6.2 PCSBL-DDT

In this approach, we adopted an EM-based sparse Bayesian learning method to exploit the shared sparsity between different delay taps and possible sparsity pattern couplings between consecutive AoAs and AoDs. We applied the algorithm to the time-domain received signals by only retaining the dominant delay taps to increase effective SNR in the signal used to form the channel estimate. First, we used the pattern-coupled Sparse Bayesian learning algorithm to the ground-truth channels in the training dataset by adding a small noise to regularize the data. In this way, we obtained the sparse representations for all the channels in the provided dataset. Then, using the respective sparse vectors and exploiting the density map of joint AoA/AoD grids, we selected a non-uniform grid and refined the pattern couplings between hyperparameters. The algorithm is applied to the test dataset to obtain the channel estimates. The final performance score in the channel estimation challenge is -9.49 dB. The NMSE values for the specific scenarios are shown in Table 2 for the testing dataset.

6.3 PC-OMP

Before evaluating the performance of PC-OMP we optimize the value of the detection threshold value δ , described in Section 5.2, in order to improve the results. We create a specific optimization method for the structured problem that arises in our approach. This optimization method is focused on reducing the optimization time while being able to perform a high-resolution grid search for the parameter values. We base our training algorithm on the fact that our approach is a greedy algorithm with a carefully chosen stopping condition. This means that

we can predict when a change in the solution will happen based on the selected threshold. Knowing this, we create a modified version of our approach that saves all channel iterations together with the computed threshold required for them to pass up to a minimum threshold value, in our case 0.7. Once outside the function, we can evaluate these channel estimations and compute the error as a step-wise function of the threshold. Then, we apply the average operation to the error step-wise functions for different scenarios to get a better and smoother result of the error behavior over different threshold values. Fig. 6 shows the smoothed step-wise error function for the different datasets. The selected threshold value is $\delta \approx 0.98$.

The algorithm with the custom detection method is applied to the test data and the obtained results are shown in Table 2. The PC-OMP algorithm outperforms the other two algorithms for the different data sets, as can be observed from the table. Specially, in lower SNRs where the channel estimation performance is lower due to noise, the PC-OMP algorithm achieves gains of up to 3 dB compared with the other two algorithms. At higher SNRs, e.g., $[-6, 0]$ dB, PC-OMP outperforms the other two algorithms by up to 4 dB.

The final performance score on the test dataset in the channel estimation challenge of the PC-OMP algorithm is -10.64 dB, outperforming the MLGS-SBL and the PCSBL-DDT methods by 1.48 dB and 1.15 dB, respectively. Also, from Table 1, we can see that the PCSBL-DDT and the PC-OMP algorithms are tuned better than the MLGS-SBL method for the training dataset that result in their better NMSE performances at SNR -15 dB and pilot frames $\{20, 40\}$. But the performance gap between MLGS-SBL and PCSBL-DDT reduces for the testing data for SNR $[-20, -11]$ dB and 20 pilot frames. Moreover, MLGS-SBL performs better than PCSBL-DDT at SNR $[-20, -11]$ dB and 40 pilot frames. This can be attributed to the fact that extracting more features from a training dataset may result in an excellent performance during training but slightly inferior performance while testing. This shows that a good model-based signal processing solution has to be combined with appropriate training, while taking into account the training and testing performance trade-off.

7. CONCLUSION

We have presented three novel signal processing approaches to estimate an mmWave channel in a hybrid analog-digital MIMO setup. We have adapted model-driven procedures to utilize the AoD, AoA, and channel gain information from a training dataset, and fine-tuned the algorithms to reduce the NMSE in the testing dataset. We empirically showed that our algorithms unanimously performed better than a purely model-based approach by a large margin on a given training data set. Hence, machine learning approaches can be potentially used in conjunction with model-driven based approaches to fine-tune them and thereby obtain better performance in physical layer wireless communication problems in realistic channel environments.

ACKNOWLEDGEMENTS

The work of Sai Subramanyam Thoota and Chandra R. Murthy was financially supported by research grants from Intel Inc. and the Dept. of Telecommunications, Govt. of India. The work of Ozlem Tuğfe Demir, Cenk M. Yetis, Emil Björnson, and Pontus Giselsson was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The work of Marios Kountouris' contribution was partially supported from a Huawei France-funded Chair towards Future Wireless Networks. The work of Christo Kurisummoottil Thomas was funded by a French FUI project titled MASS-START.

REFERENCES

- [1] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 436–453, 2016.
- [2] R. Méndez-Rial, C. Rusu, N. González-Prelcic, A. Alkhateeb, and R. W. Heath, "Hybrid MIMO architectures for millimeter wave communications: Phase shifters or switches?" *IEEE Access*, vol. 4, pp. 247–267, 2016.
- [3] E. Karami, "Tracking performance of least squares MIMO channel estimation algorithm," *IEEE Trans. Commun.*, vol. 55, no. 11, pp. 2201–2209, 2007.
- [4] J. Rodríguez-Fernández, N. González-Prelcic, K. Venugopal, and R. W. Heath, "Frequency-domain compressive channel estimation for frequency-selective hybrid millimeter wave MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 2946–2960, 2018.
- [5] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 831–846, 2014.
- [6] Z. Marzi, D. Ramasamy, and U. Madhow, "Compressive channel estimation and tracking for large arrays in mm-wave picocells," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 514–527, 2016.
- [7] S. Liu, F. Yang, W. Ding, X. Wang, and J. Song, "Two-dimensional structured-compressed-sensing-based NBI cancellation exploiting spatial and temporal correlations in MIMO systems," *IEEE Trans. Veh. Tech.*, vol. 65, no. 11, pp. 9020–9028, 2016.
- [8] K. Venugopal, A. Alkhateeb, N. G. Prelcic, and R. W. Heath, "Channel estimation for hybrid architecture-based wideband millimeter wave systems," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1996–2009, 2017.
- [9] E. Vlachos, G. C. Alexandropoulos, and J. Thompson, "Wideband MIMO channel estimation for hybrid beamforming millimeter wave systems via random spatial sampling," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 1136–1150, 2019.
- [10] Y. Wang, Y. Zhang, Z. Tian, G. Leus, and G. Zhang, "Super-resolution channel estimation for arbitrary arrays in hybrid millimeter-wave massive MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 947–960, 2019.
- [11] F. Dong, W. Wang, Z. Huang, and P. Huang, "High-resolution angle-of-arrival and channel estimation for mmWave massive MIMO systems with lens antenna array," *IEEE Trans. Veh. Tech.*, vol. 69, no. 11, pp. 12 963–12 973, 2020.
- [12] H. Xie and N. González-Prelcic, "Dictionary learning for channel estimation in hybrid frequency-selective mmWave MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7407–7422, 2020.
- [13] J. P. González-Coma, J. Rodríguez-Fernández, N. González-Prelcic, L. Castedo, and R. W. Heath, "Channel estimation and hybrid precoding for frequency selective multiuser mmWave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 2, pp. 353–367, 2018.
- [14] S. Gao, X. Cheng, and L. Yang, "Estimating doubly-selective channels for hybrid mmWave massive MIMO systems: A doubly-sparse approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5703–5715, 2020.
- [15] M. Jian, F. Gao, Z. Tian, S. Jin, and S. Ma, "Angle-domain aided UL/DL channel estimation for wideband mmWave massive MIMO systems with beam squint," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3515–3527, 2019.
- [16] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and S. Li, "Channel estimation for millimeter-wave multiuser MIMO systems via PARAFAC decomposition," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7501–7516, 2016.
- [17] J. Yang, C. Wen, S. Jin, and F. Gao, "Beamspace channel estimation in mmWave systems via cosparse image reconstruction technique," *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4767–4782, 2018.
- [18] L. Lian and V. K. N. Lau, "Configuration optimization and channel estimation in hybrid beamforming mmWave systems with channel support side information," *IEEE Trans. Signal Process.*, vol. 68, pp. 6026–6039, 2020.

- [19] C. K. Anjinappa, A. C. Gürbüz, Y. Yapıcı, and İ. Güvenç, "Off-grid aware channel and covariance estimation in mmWave networks," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3908–3921, 2020.
- [20] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and R. S. Blum, "Low-rank tensor decomposition-aided channel estimation for millimeter wave MIMO-OFDM systems," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 7, pp. 1524–1538, 2017.
- [21] B. Wang, M. Jian, F. Gao, G. Y. Li, and H. Lin, "Beam squint and channel estimation for wide-band mmWave massive MIMO-OFDM systems," *IEEE Trans. Signal Process.*, vol. 67, no. 23, pp. 5893–5908, 2019.
- [22] F. Bellili, F. Sahrabi, and W. Yu, "Generalized approximate message passing for massive MIMO mmWave channel estimation with Laplacian prior," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3205–3219, 2019.
- [23] F. Talaei and X. Dong, "Hybrid mmWave MIMO-OFDM channel estimation based on the multi-band sparse structure of channel," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1018–1030, 2019.
- [24] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath, "5G MIMO data for machine learning: Application to beam-selection using deep learning," in *Proc. IEEE Inf. Theory and App. Workshop. IEEE*, 2018, pp. 1–9.
- [25] P. Schniter and A. Sayeed, "Channel estimation and precoder design for millimeter-wave communications: The sparse way," in *Proc. Asilomar Conf. on Signals, Syst., and Comput.* IEEE, 2014, pp. 273–277.
- [26] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, 2004.
- [27] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 912–926, 2011.
- [28] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, no. Jun, pp. 211–244, 2001.
- [29] J. Fang, L. Zhang, and H. Li, "Two-dimensional pattern-coupled sparse Bayesian learning via generalized approximate message passing," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2920–2930, 2016.
- [30] J. Fang, Y. Shen, H. Li, and P. Wang, "Pattern-coupled sparse Bayesian learning for recovery of block-sparse signals," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 360–372, 2015.
- [31] J. Palacios, D. De Donno, and J. Widmer, "Lightweight and effective sector beam pattern synthesis with uniform linear antenna arrays," *IEEE Antennas and Wireless Prop. Letters*, vol. 16, pp. 605–608, 2017.

AUTHORS



Sai Subramanyam Thoota

(IEEE Graduate Student Member) received his B. E. degree in Electronics and Communication Engineering from the College of Engineering, Guindy, Anna University, Chennai, India, in 2003, and his M. Tech. degree in Electrical Engineering from the Indian Institute of Technology,

Madras, Chennai, India, in 2006. From 2008 to 2015, he worked as a Chief engineer for Samsung Research India Bangalore, India, where he worked on GSM/GPRS/EDGE baseband transceiver design. From 2015 to 2016, he worked as a Project Associate at the Dept. of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India. Since 2016, he is pursuing his Ph. D. at the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India. His research interests include massive MIMO systems, sparse signal recovery and statistical signal processing.



Dolores Garcia Marti

received her bachelor's degree in Mathematics from the University of Valencia, Spain in 2017. She received her M. Sc. in Theoretical Physics degree from Imperial College, London, UK in 2018. Since 2018, she is pursuing a Ph. D. in communications at IMDEA Networks associated to the University of Carlos III in Madrid, Spain. Her research interest lie in the intersection between machine learning and millimeter-wave communications.



Özlem Tuğfe Demir

received her B. S., M. S., and Ph. D. degrees in Electrical and Electronics Engineering from Middle East Technical University, Ankara, Turkey, in 2012, 2014, and 2018, respectively. She was a Postdoctoral Researcher with Linköping University, Sweden in 2019–2020. She is currently a Postdoc with KTH Royal Institute of Technology, Sweden. She has authored the textbook *Foundations of User-Centric Cell-Free Massive MIMO* (2021). Her research interests focus on signal processing and optimization in wireless communications, massive MIMO, beyond 5G multiple antenna technologies, deep learning, and green communications.

She is a recipient of the IEEE SIU 2015 Conference Student Best Paper Award, the Best Thesis Award for M.S. Program and Graduate Courses Performance Award at the Middle East Technical University.



Rakesh Mundlamuri received his B. Tech. degree in Electronics and Communication Engineering from Shiv Nadar University, Greater Noida, India in 2018. From 2018 to 2019, he worked as a project assistant in the 5G testbed development project at the Dept. of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India.

Since 2019, he is pursuing his M. Sc. in mobile computing systems at EURECOM, France. In his M. Sc. he worked on the intersection of the topics of sparse Bayesian learning and machine learning as well as real-time 5G mm-wave systems using OpenAirInterface. His research interests include machine learning in communication systems and real-time systems.



Joan Palacios received his B. Sc. degree in mathematics from the Universitat de Valencia, Spain, in 2015, and the M. Sc. degree in multimedia and communications in 2016 and PhD degree in 2020, both from IMDEA Networks and the University of Carlos III, Spain. His areas of interest lie in millimeter-wave communications with main

research focus on the development of efficient schemes for beam training and tracking, hybrid analog-digital beamforming, and angle difference of arrival-based localization for quasi-optical communication systems.



Cenk M. Yetis [S'00, M'10] (cenkmyetis@ieee.org) received his B. Sc. degree in electronics engineering from Isik University, Istanbul, Turkey, in 2001, his M. Sc. degree in telecommunications engineering from Istanbul Technical University (ITU), in 2004, and his Ph. D. degree in satellite communications and remote sensing from ITU in 2010. He received The Scientific and Technological Research Council of Turkey (TUBITAK) scholarship from 2005 to 2009. From 2003 to 2007, he was with Turk Telekom (formerly Avea), one of the top three wireless services providers in Turkey, where he held rotational responsibilities in operation and planning groups. From 2007 to 2010, he was a visiting researcher at Ohio State University and the University of California, Irvine. From 2010 to 2019, he held academic positions in universities and research centers in Hong Kong, Singapore, Turkey, and Taiwan (Province of China). In 2019, he joined Lund University, Sweden as a Researcher to extend his research and teaching activities. In 2017, he received a

2-year Academia Sinica fellowship. His research interests include signal processing, information theory, communication theory, and optimization theory for wireless communications.



Christo Kurisummoottil Thomas received his B. Tech. degree in Electronics and Communication Engineering from National Institute of Technology, Calicut, India, in 2010, his M. E. degree in telecommunication engineering from the Indian Institute of Science, Bengaluru, India, in 2012, and his Ph. D. degree in communication systems from EURECOM, France, in 2020. From 2012 to 2014, he was a staff design engineer with Broadcom Communication Technologies Pvt., Ltd., Bengaluru and from 2014 to 2017, he was a design engineer with Intel Corporation, in Bengaluru. Since November 2020, he has been a staff engineer with the Wireless Research and Development, Qualcomm Inc., Espoo, Finland. His research interests include areas in wireless communication, statistical signal processing, compressed sensing, and the analysis and design of mmWave systems.

Sameera Bharadwaja Hayavadana received his B. E. degree in Telecommunication Engineering from P. E. S. Institute of Technology, Bangalore, India in 2009 and his M. Tech. degree in Communication Systems from the Indian Institute of Technology Roorkee, Roorkee, India in 2011. From 2011 to 2013, he worked as a systems engineer in Innovation Labs, Tata Consultancy Services, Delhi where he worked on machine learning algorithms and applications and in 2013, he joined Samsung Semiconductor India R&D (SSIR), Bangalore, India where he is currently working as a senior staff engineer. His work at SSIR involves design and development of machine learning algorithm modules for semiconductor FAB equipment diagnosis. Since 2019, he is pursuing his Ph. D. as an external registrant from SSIR at Signal Processing for Communications lab in the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India. His research interests include group testing, sparse signal recovery and machine learning.



Emil Björnson (Senior Member, IEEE) received his M. S. degree in engineering mathematics from Lund University, Sweden, in 2007, and his Ph. D. degree in telecommunications from the KTH Royal Institute Technology, Sweden, in 2011. From 2012 to 2014, he held a joint postdoctoral position with the Alcatel-Lucent Chair on Flexible Radio,

Emil Björnson (Senior Member, IEEE) received his M. S. degree in engineering mathematics from Lund University, Sweden, in 2007, and his Ph. D. degree in telecommunications from the KTH Royal Institute Technology, Sweden, in 2011. From 2012 to 2014, he held a joint postdoctoral position with the Alcatel-Lucent Chair on Flexible Radio,



in 2011. From 2012 to 2014, he held a joint postdoctoral position with the Alcatel-Lucent Chair on Flexible Radio,

SUPELEC, France, and the KTH Royal Institute of Technology. He joined Linköping University, Sweden, in 2014, where he is currently an associate professor. In September 2020, he became a part-time Visiting Full Professor at the KTH Royal Institute of Technology.

He has authored the textbooks *Optimal Resource Allocation in Coordinated Multi-Cell Systems* (2013), *Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency* (2017), and *Foundations of User-Centric Cell-Free Massive MIMO* (2021). He is dedicated to reproducible research and has made a large amount of simulation code publicly available. He performs research on MIMO communications, radio resource allocation, machine learning for communications, and energy efficiency. He has been on the Editorial Board of the IEEE Transactions on Communications since 2017. He has been a member of the Online Editorial Team of the IEEE Transactions on Wireless Communications since 2020. He has also been a guest editor of multiple special issues.

He has performed MIMO research for over ten years, his papers have received more than 10000 citations, and he has filed more than twenty patent applications. He is a host of the podcast Wireless Future and has a popular YouTube channel. He has received the 2014 Outstanding Young Researcher Award from IEEE ComSoc EMEA, the 2015 Ingvar Carlsson Award, the 2016 Best Ph.D. Award from EURASIP, the 2018 IEEE Marconi Prize Paper Award in Wireless Communications, the 2019 EURASIP Early Career Award, the 2019 IEEE Communications Society Fred W. Ellersick Prize, and the 2019 IEEE Signal Processing Magazine Best Column Award. He also co-authored papers that received Best Paper Awards at conferences, including WCSP 2009, the IEEE CAMSAP 2011, the IEEE SAM 2014, the IEEE WCNC 2014, the IEEE ICC 2015, and WCSP 2017.



Pontus Giselsson received his M. Sc. and Ph. D. degrees from Lund University, Sweden, in 2006 and 2012, respectively. During 2013 and 2014, he held a postdoctoral position at Stanford University. He is an associate professor in the Department of Automatic Control at Lund University. In 2012, he received the Young Author Prize at

the Advanced Control of Chemical Processes IFAC Symposium; in 2014, he received the Young Author Prize at the IFAC World Congress; and in 2015, he received the Ingvar Carlsson Award from the Swedish Foundation for Strategic Research. His research interests include mathematical optimization and its wide range of applications, e.g., in machine learning, control, signal processing, and wireless communication.



Marios Kountouris (S'04–M'08–SM'15) received his diploma degree in electrical and computer engineering from NTUA, Athens, Greece in 2002 and the M. Sc. and Ph. D. degrees in electrical engineering from Télécom Paris, France in 2004 and 2008, respectively. He is currently an associate

professor and Chair PI on Advanced Wireless Systems at EURECOM, France. Prior to his current appointment, he has held positions at Huawei Paris Research Center, France, CentraleSupélec, France, The University of Texas at Austin, USA, and Yonsei University, S. Korea. He was awarded a consolidator grant by the European Research Council (ERC) in 2020 on goal-oriented semantic communication. He has served as Associate Editor for the IEEE Transactions on Wireless Communications, the IEEE Transactions on Signal Processing, and the IEEE Wireless Communication Letters. He has received several awards, including the 2020 IEEE ComSoc Young Author Best Paper Award, the 2016 IEEE ComSoc CTTC Early Achievement Award, the 2013 IEEE ComSoc Outstanding Young Researcher Award for the EMEA Region, the 2012 IEEE SPS Signal Processing Magazine Award, the IEEE SPAWC 2013 Best Paper Award, and the IEEE Globecom 2009 CT Best Paper Award.



Chandra R. Murthy (S'03–M'06–SM'11) received his B. Tech. degree in Electrical Engineering from the Indian Institute of Technology, Madras in 1998, M. S. and Ph. D. degrees in Electrical and Computer Engineering from Purdue University and the University of California, San Diego, in 2000 and 2006,

respectively. From 2000 to 2002, he worked as an engineer for Qualcomm Inc., where he worked on WCDMA baseband transceiver design and 802.11b baseband receivers. From Aug. 2006 to Aug. 2007, he worked as a staff engineer at Beceem Communications Inc. on advanced receiver architectures for the 802.16e Mobile WiMAX standard. In Sept. 2007, he joined the Department of Electrical Communication Engineering at the Indian Institute of Science, Bangalore, India, where he is currently working as a professor.

His research interests are in the areas of energy harvesting communications, 5G/6G technologies and compressed sensing. He has over 71 journal and 98 conference papers to his credit. He is a recipient of the MeitY Young Faculty Fellowship from the Govt. of India and the Prof. Satish Dhawan state award for engineering from the Karnataka State Government. He was an associate editor for the IEEE SIGNAL PROCESSING LETTERS during 2012-16 and the SADHANA ACADEMY PROCEEDINGS IN ENGINEERING SCIENCES during 2017-18.

He is a past Chair of the IEEE Signal Processing Society, Bangalore Chapter. He was an elected member of the IEEE SPCOM Technical Committee during 2014-19. He is currently serving as a senior area editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and as an associate editor for the IEEE TRANSACTIONS ON COMMUNICATIONS and the IEEE TRANSACTIONS ON INFORMATION THEORY.



Nuria González-Prelcic (S'96-M'00-SM'17) is currently an associate professor in the Electrical and Computer Engineering Department at North Carolina State University. Her main research interests include signal processing theory and signal processing and

machine learning for wireless communications: filter banks, compressive sampling and estimation, multicarrier modulation, massive MIMO, MIMO processing for millimeter-wave communication and sensing, including vehicle-to-everything (V2X), air-to-everything (A2X) and satellite MIMO communication. She has published more than 80 papers in the topic of signal processing for millimeter-wave communications. She is an Editor for the IEEE Transactions on Wireless Communications. She is a member of the IEEE Sensor Array and Multichannel Signal Processing Technical Committee. She was the founder director of the Atlantic Research Center for Information and Communication Technologies (atlanTTic) at the University of Vigo from July 2008 to January 2017.



Joerg Widmer (M'06-SM'10-F'20) is Research Professor and Research Director of IMDEA Networks in Madrid, Spain. Before, he held positions at DOCOMO Euro-Labs in Munich, Germany and EPFL, Switzerland. He was a visiting researcher at the International Computer Science Institute in Berkeley, USA, University College

London, UK, and TU Darmstadt, Germany. His research focuses on wireless networks, ranging from extremely high frequency millimeter-wave communication and MAC layer design to mobile network architectures. Joerg Widmer authored more than 150 conference and journal papers and three IETF RFCs, and holds 13 patents. He was awarded an ERC consolidator grant, the Friedrich Wilhelm Bessel Research Award of the Alexander von Humboldt Foundation, a Mercator Fellowship of the German Research Foundation, a Spanish Ramon y Cajal grant, as well as eight best paper awards. He is an IEEE Fellow and distinguished member of the ACM.