Robust Recovery of Structured Sparse Signals With Uncertain Sensing Matrix: A Turbo-VBI Approach

An Liu, *Senior Member, IEEE*, Guanying Liu, Lixiang Lian, *Student Member, IEEE*, Vincent K. N. Lau, *Fellow, IEEE*, and Min-Jian Zhao, *Member, IEEE*

***Abstract*— In many applications in wireless communications, we need to recover a structured sparse signal from a linear measurement model with uncertain sensing matrix. There are two challenges of designing an algorithm framework for this problem. How to choose a flexible yet tractable sparse prior to capture different structured sparsities in specific applications? How to handle a sensing matrix with uncertain parameters and possibly correlated entries? As will be explained in the introduction, existing common methods in compressive sensing (CS), such as approximate message passing (AMP) and variational Bayesian inference (VBI), may not work well. To better address this problem, we propose a novel Turbo-VBI algorithm framework, in which a three-layer hierarchical structured (3LHS) sparse prior model is proposed to capture various structured sparsities that may occur in practice. By combining the message passing and VBI approaches via the turbo framework, the proposed Turbo-VBI algorithm is able to fully exploit the structured sparsity (as captured by the 3LHS sparse prior) for robust recovery of structured sparse signals under an uncertain sensing matrix. Finally, we apply the Turbo-VBI framework to solve two application problems in wireless communications and demonstrate its significant gain over the state-of-art CS algorithms.**

***Index Terms*— Variational Bayesian inference (VBI), structured sparse signal recovery, uncertain sensing matrix.**

I. Introduction

THE main objective of compressive sensing (CS) is to recover a high dimensional sparse signal $\boldsymbol{x} = [x_1, \ldots, x_N]^T \in \mathbb{C}^N$ from significantly fewer measurements $\boldsymbol{y} \in \mathbb{C}^M$ (i.e., $M \ll N$) under the following linear measurement model:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}, \tag{1}$$

Manuscript received April 24, 2019; revised November 11, 2019; accepted January 20, 2020. Date of publication February 10, 2020; date of current version May 8, 2020. This work was supported by the Science and Technology Program of Shenzhen, China, under Grant JCYJ20170818113908577. The work of An Liu was supported by the China Recruitment Program of Global Young Experts. The associate editor coordinating the review of this article and approving it for publication was M. Payaró. *(Corresponding authors: Vincent K. N. Lau; Min-Jian Zhao.)*

An Liu, Guanying Liu, and Min-Jian Zhao are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: anliu@zju.edu.cn; mjzhao@zju.edu.cn).

Lixiang Lian is with the Department of ECE, The Hong Kong University of Science and Technology, Hong Kong (e-mail: llianab@connect.ust.hk).

Vincent K. N. Lau is with the HKUST Shenzhen Research Institute, Shenzhen 518057, China (e-mail: eeknlau@ece.ust.hk).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2020.2971193

where $\boldsymbol{A} \in \mathbb{C}^{M\times N}$ is the sensing matrix, $\boldsymbol{w} \in \mathbb{C}^M$ is the noise vector with independent Gaussian entries $w_m \sim \mathcal{CN}\left(w_m; 0, \kappa_m^{-1}\right)$, $\mathcal{CN}(w; \mu, \Sigma)$ denotes the probability density function (PDF) of a complex Gaussian variable w with mean μ and variance Σ. CS plays a key role in many engineering applications, such as image signal processing [1], wireless communications [2], [3], autonomous driving [4], etc. In the standard CS model, the sensing matrix $\boldsymbol{A}$ is perfectly known and $\boldsymbol{x}$ is an i.i.d. sparse signal. However, in many applications, the sensing matrix $\boldsymbol{A}(\boldsymbol{\theta})$ may contain uncertain parameters $\boldsymbol{\theta} \in \mathbb{R}^K$. Moreover, in specific applications, the sparse signal $\boldsymbol{x}$ usually has structured sparsity that cannot be modeled easily by i.i.d. priors. Therefore, it is necessary to design more efficient and robust CS algorithms for the reconstruction of structured sparse signals with uncertain sensing matrix. There are three common methods in the literature.

Generalized LASSO: A widely used method to recover a sparse signal $\boldsymbol{x}$ is the generalized LASSO [5], which obtains an estimate $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ by solving the following problem:

$$\hat{\mathbf{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}||_2^2 + \tau f(\boldsymbol{x}). \tag{2}$$

where $f(\boldsymbol{x})$ is a penalty function, and $\tau > 0$ is a coefficient. The structured sparsity can be encoded in the penalty function. For example, if $\boldsymbol{x}$ is a group-sparse signal (i.e., $\boldsymbol{x}$ can be divided into B blocks $\boldsymbol{x} = [\boldsymbol{x}_1; \ldots; \boldsymbol{x}_B]$ and there are only a few non-zero blocks), we can choose $f(\boldsymbol{x}) = \sum_{i=1}^{B} \|\boldsymbol{x}_i\|$ to impose the group-sparse structure on $\boldsymbol{x}$ [6], [7]. There are several major drawbacks of generalized LASSO. First, the performance is sensitive to the choice of penalty function. Second, the structured sparsity cannot be well modeled using a simple convex penalty function $f(\boldsymbol{x})$. However, if we choose a non-convex penalty function, the complexity is high and the performance also tends to be bad because the algorithm can easily get stuck at a bad local optimum.

Approximate Message Passing (AMP): AMP is an iterative algorithm obtained by simplifying the well known sum-product message passing (SPMP) algorithm at the asymptotic region when $M, N \to \infty$ [8]. AMP can potentially provide near-optimal performance when i.i.d. Gaussian sensing matrices and i.i.d. sparse priors are involved [9]. Various variations of AMP have been proposed to handle non-i.i.d. sensing matrices and more complicated sparse priors. For example, orthogonal AMP (OAMP) has been proposed in [10] to achieve a better performance than the AMP when the

sensing matrix is a partial orthogonal matrix. In addition, AMP and turbo approach have been combined to design advanced CS algorithms such as Turbo-AMP [11] and Turbo-CS [12] algorithms, which can handle more complicated priors. However, the performance of these AMP-based algorithms can be very poor under a general sensing matrix.

Sparse Bayesian learning (SBL) and variational Bayesian inference (VBI): Recently, SBL/VBI has been proposed to solve CS problems, in which a two-layer hierarchical prior is used to model i.i.d. sparsity or group-sparsity [13]–[15]. In SBL/VBI, the unknown parameter $\boldsymbol{\theta}$ is first learned based on the expectation-maximization (EM) method [16], then the sparse signal $\boldsymbol{x}$ is recovered using a MAP estimator based on the learned parameter $\hat{\boldsymbol{\theta}}$ and measurements $\boldsymbol{y}$. However, due to the limitation of the two-layer hierarchical prior, the SBL/VBI methods cannot handle more sophisticated sparse priors, such as the Hidden Markov priors [17].

In this paper, we propose a novel Turbo-VBI framework to overcome the drawbacks of the existing methods. The Turbo-VBI framework can exploit structured sparsity to improve the recovery performance. It is robust w.r.t. the uncertain parameters in the sensing matrix and prior distribution and it works well for general sensing matrices. The main contributions are summarized below.

- **Three-layer hierarchical probability model for structured sparsity:** The choice of sparse probability model is paramount to robust and accurate recovery of structured sparse signals. A good sparse probability model should satisfy the following criteria: it is flexible to capture different structured sparsities, it is robust w.r.t. the imperfect prior information, it is tractable to enable low-complexity algorithm design. We propose a three-layer hierarchical structured (3LHS) sparse prior model to meet these criteria. Specifically, there are sufficient degrees of freedom in the model which can be used to fit the specific structure of sparse signals as well as incorporate the uncertainty of imperfect prior information.
- **Turbo-VBI algorithm design:** There still lacks efficient algorithms for solving CS problems with 3LHS sparse prior and potentially ill-conditioned sensing matrix. By combining the message passing and VBI approaches via the turbo framework, we propose a Turbo-VBI algorithm which is able to fully exploit the structured sparsity (as captured by the 3LHS sparse prior) under an uncertain (and possibly correlated) sensing matrix to achieve significant gain over the state-of-art CS algorithms.
- **Specific Turbo-VBI algorithm design for some important applications:** We apply the Turbo-VBI framework to solve two application problems in wireless communications. We believe that the proposed Turbo-VBI-based solutions for these application problems alone are of great interest to the wireless community.

The rest of the paper is organized as follows. In Section II, we present the 3LHS sparsity model. In Section III, we formulate the CS recovery problem under 3LHS sparse prior and uncertain sensing matrix, together with two application examples in wireless communications. The proposed Turbo-VBI algorithm is presented in Section IV and is applied to solve two important application problems in Section V. Finally, the conclusion is given in Section VI.

Notation: For a vector $\boldsymbol{x} \in \mathbb{C}^N$ and a given index set $\mathcal{I} \subseteq \{1, \ldots, N\}$, $|\mathcal{I}|$ denotes its cardinality, $\boldsymbol{x}[\mathcal{I}] \in \mathbb{C}^{|\mathcal{I}|}$ denotes the subvector consisting of the elements of $\boldsymbol{x}$ indexed by the set $\mathcal{I}$. $\mathrm{diag}(\boldsymbol{x})$ denotes a block diagonal matrix with $\boldsymbol{x}$ as the diagonal elements. Finally, $1(\cdot)$ denotes the indication function and $x = \Theta(a)$ for $a > 0$ denotes that $\exists k_1, k_2 > 0$, such that $k_2 \cdot a \leq x \leq k_1 \cdot a$.

II. Three-Layer Hierarchical Structured Sparsity Model

A. Motivation of 3LHS Structured Sparsity

The probability model for structured sparsity provides the foundation for exploiting the specific sparse structures. There are two major existing probability models, as elaborated below.

1) Support-Based Probability Model: In AMP-based algorithms, a support-based probability model is used to capture the structured sparsity [18], where a vector $\boldsymbol{s}$ is introduced to indicate the support of the signal. In particular, $s_n = 1$ indicates that the signal coefficient x_n is active (non-zero), while $s_n = 0$ indicates that x_n is inactive (zero). Given vector $\boldsymbol{s}$, $\boldsymbol{x}$ is assumed to have independent but non-identically distributed entries,, $p(\boldsymbol{x}|\boldsymbol{s}) = \prod_{n=1}^{N} p(x_n|s_n)$, where

$$p(x_n|s_n) = s_n g_n(x_n) + (1 - s_n)\,\delta(x_n), \tag{3}$$

where $g_n(x_n)$ denotes the PDF of x_n conditioned on $s_n = 1$, which is often chosen as a Gaussian distribution. The structured sparsity is captured by the prior distribution $p(\boldsymbol{s})$ of the support vector. By choosing a proper $p(\boldsymbol{s})$, the support-based probability model has the flexibility to cover a wide range of structured sparsities, such as Markov sparsity [11] and Markov tree sparsity [12]. However, it is difficult to handle the binary vector $\boldsymbol{s}$ using the optimization-based algorithms and thus AMP-based algorithms are usually used to recover sparse signals $\boldsymbol{x}$ with the support-based probability model, which limits its application since AMP-based algorithms only work well for certain types of sensing matrix (e.g., i.i.d. or partial orthogonal sensing matrices).

2) Two-Layer Hierarchical Probability Model: In SBL/VBI, a two-layer hierarchical prior is used to promote i.i.d. or group sparsity [15], where a precision vector $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_N]^T$ (i.e., $1/\rho_n$ denotes the variance of x_n) is introduced to indicate whether the n-th element x_n is active ($\rho_n = \Theta(1)$) or inactive ($\rho_n \gg 1$). Given the vector $\boldsymbol{\rho}$, $\boldsymbol{x}$ is assumed to have independent but non-identically distributed Gaussian entries, i.e., $p(\boldsymbol{x}|\boldsymbol{\rho}) = \prod_{n=1}^{N} p(x_n|\rho_n)$, where

$$p(x_n|\rho_n) = \mathcal{CN}\left(x_n; 0, \rho_n^{-1}\right), \forall n. \tag{4}$$

and $\rho_n, \forall n$ are modeled as independent Gamma distributions, i.e., $p(\boldsymbol{\rho}) = \prod_{n=1}^{N} p(\rho_n)$ with

$$p(\rho_n) = \Gamma(\rho_n; a_n, b_n), \tag{5}$$

where a_n, b_n are set to be a small number to promote sparsity of $\boldsymbol{x}$ [15]. The performance of SBL/VBI with such two-layer

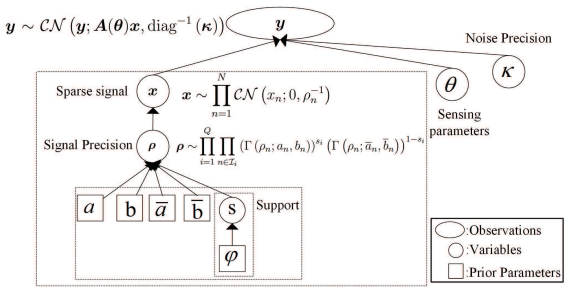


Fig. 1. Three-layer hierarchical structured sparse prior.

hierarchical prior is insensitive to the sensing matrix. It is also possible to model the group sparsity by assigning the same precision ρ_i to the i-th group of elements in $\boldsymbol{x}$. However, it is not flexible enough to model more complicated sparse structures, such as the Markov (tree) priors or Hidden Markov priors considered in [12], [17], because the precision vector $\boldsymbol{\rho}$ is fixed to be (independent) Gamma distributions to enable tractable/low-complexity algorithm design based on SBL/VBI. Moreover, in many practical applications, it is possible to obtain some statistical prior support information (PSI) which indicates the probability of each element being active (i.e., $\Pr(s_n = 1), \forall n$) [19]. However, it is difficult to incorporate such statistical PSI into the two-layer hierarchical prior.

In the following, we shall introduce a 3LHS sparse model to capture the more complicated structured sparsity that may occur in practice, by combining the advantages of the support-based probability model and two-layer hierarchical prior.

B. Probability Model for the 3LHS Structured Sparsity

Without loss of generality, suppose the index set $\{1, \ldots, N\}$ of $\boldsymbol{x}$ can be partitioned into Q non-overlapping subsets $\mathcal{I}_1, \ldots, \mathcal{I}_Q$ such that all the elements of $\boldsymbol{x}[\mathcal{I}_i], \forall i \in \{1, \ldots, Q\}$ are simultaneously active or inactive. Correspondingly, we introduce a support vector $\boldsymbol{s} = [s_1, \ldots, s_Q]^T \in \{0,1\}^Q$ to indicate whether the i-th subvector $\boldsymbol{x}[\mathcal{I}_i]$ is active ($s_i = 1$) or inactive ($s_i = 0$). Specifically, let $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_N]^T$ denote the precision vector of $\boldsymbol{x}$ (i.e., $1/\rho_n$ denotes the variance of x_n). When $s_i = 0$, the distribution of the associated precision parameters $\rho_n, \forall n \in \mathcal{I}_i$ is chosen to satisfy $\mathbb{E}[\rho_n] \gg 1, \forall n \in \mathcal{I}_i$ such that the expected variance of $x_n, \forall n \in \mathcal{I}_i$ is close to zero (inactive). Moreover, to improve the robustness w.r.t. the imperfect prior knowledge, we assume that the prior distribution $p(\boldsymbol{s}|\boldsymbol{\phi})$ of the support vector depends on some uncertain parameter $\boldsymbol{\phi}$ with a known prior distribution $p(\boldsymbol{\phi})$. Then for given uncertain prior parameter $\boldsymbol{\phi}$, the 3LHS sparse prior distribution (joint distribution of $\boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{s}$) is given by

$$p(\boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{s}|\boldsymbol{\phi}) = \underbrace{p(\boldsymbol{s}|\boldsymbol{\phi})}_{\text{Support}} \underbrace{p(\boldsymbol{\rho}|\boldsymbol{s})}_{\text{Precision}} \underbrace{p(\boldsymbol{x}|\boldsymbol{\rho})}_{\text{Sparse signal}}. \tag{6}$$

The 3LHS sparse model is illustrated in Fig. 1.

Probability Model for the Support Vector $\boldsymbol{s}$ (Layer 1): The prior distribution $p(\boldsymbol{s}|\boldsymbol{\phi})$ of the support vector is used to capture the structured sparsity in specific applications. In practice, $p(\boldsymbol{s}|\boldsymbol{\phi})$ is chosen based on the nature of the problem. For example, in [12], $p(\boldsymbol{s}|\boldsymbol{\phi})$ is chosen to be a Markov chain to model the clustered scattering environment in massive MIMO channel, where $\boldsymbol{\phi}$ denotes the (possibly unknown) transition probabilities in the Markov chain and it can be automatically learned from the observations, as will be detailed later.

Probability Model for the Precision Vector $\boldsymbol{\rho}$ (Layer 2) : The conditional probability $p(\boldsymbol{\rho}|\boldsymbol{s})$ is given by

$$p(\boldsymbol{\rho}|\boldsymbol{s}) = \prod_{i=1}^{Q} \prod_{n \in \mathcal{I}_i} \left(\Gamma(\rho_n; a_n, b_n)\right)^{s_i} \left(\Gamma\left(\rho_n; \overline{a}_n, \overline{b}_n\right)\right)^{1-s_i}, \tag{7}$$

where $\Gamma(\rho; a, b)$ is a Gamma hyperprior with shape parameter a and rate parameter b. When $s_i = 1$, $\boldsymbol{x}[\mathcal{I}_i]$ is active. In this case, the shape and rate parameters a_n, b_n of its precision $\rho_n, \forall n \in \mathcal{I}_i$ should be chosen such that $\frac{a_n}{b_n} = \mathbb{E}[\rho_n] = \Theta(1)$ since the variance $1/\rho_n$ of $x_n, \forall n \in \mathcal{I}_i$ is $\Theta(1)$ when it is active. On the other hand, when $s_i = 0$, $\boldsymbol{x}[\mathcal{I}_i]$ is inactive. In this case, the shape and rate parameters $\overline{a}_n, \overline{b}_n$ of its precision $\rho_n, \forall n \in \mathcal{I}_i$ should be chosen to satisfy $\frac{\overline{a}_n}{\overline{b}_n} = \mathbb{E}[\rho_n] \gg 1$ such that the inactive coefficient $x_n, \forall n \in \mathcal{I}_i$ is close to zero.

The motivation of considering Gamma hyperprior for $p(\rho_n|s_n)$ is twofold. First, it is conjugate to Gaussian, hence the associated Bayesian inference can be performed in closed form as will be detailed later. Moreover, as explained above, the conditional probability $p(\boldsymbol{\rho}|\boldsymbol{s})$ can be used to capture the sparsity structure by controlling the mean of the precisions (i.e., the shape parameter a and rate parameter b of the Gamma hyperprior) based on the support vector $\boldsymbol{s}$.

Probability Model for the Sparse Signal $\boldsymbol{x}$ (Layer 3): The conditional probability $p(\boldsymbol{x}|\boldsymbol{\rho})$ for the sparse signal is assumed to have a product form $p(\boldsymbol{x}|\boldsymbol{\rho}) = \prod_{n=1}^{N} p(x_n|\rho_n)$ and each $p(x_n|\rho_n)$ is modeled as a complex Gaussian prior distribution

$$p(x_n|\rho_n) = \mathcal{CN}\left(x_n; 0, \rho_n^{-1}\right), \quad \forall n = 1, \ldots, N. \tag{8}$$

The motivation of considering complex Gaussian distribution for $p(x_n|\rho_n)$ is twofold. First, random signals in the nature tend to have a Gaussian distribution due to the central limit theorem. Second, assuming conditional Gaussian prior distribution facilitates low-complexity VBI algorithm design with closed-form update equations [15]. It is well known that the performance of CS recovery algorithms is usually not sensitive to the true distribution of the sparse signal $\boldsymbol{x}$ [20], as long as the proposed probability model can capture the first-order sparse structure of $\boldsymbol{x}$.

The proposed 3LHS sparse model can enjoy the benefits of both the support-based probability model and two-layer hierarchical prior. On one hand, the flexibility of the support-based probability model is preserved as we can choose a proper $p(\boldsymbol{s}|\boldsymbol{\phi})$ to model different structured sparsities in various applications. For example, in [11], $p(\boldsymbol{s}|\boldsymbol{\phi})$ is chosen to be a Markov tree prior to model the wavelet structure in image processing, where $\boldsymbol{\phi}$ denotes the statistical parameters (e.g., the transition probabilities) in the Markov tree prior, and it can be automatically learned using, e.g., the EM-like method. Such complicated sparse structure, however, cannot be modeled by the two-layer hierarchical prior in (4) and (5). On the

other hand, the 3LHS sparse model also facilitates the design of a Turbo-VBI algorithm, in which the observation model $\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x} + \boldsymbol{w}$ with a general sensing matrix is handled using the VBI approach, while the structured sparsity captured by the prior $p(\boldsymbol{s})$ is handled using the message passing approach. Note that there is no need to specify the layer 2 and 3 distributions $p(\boldsymbol{\rho}|\boldsymbol{s})$ and $p(\boldsymbol{x}|\boldsymbol{\rho})$ for each application since they are fixed as in (7) and (8) for all applications.

III. CS Problem Formulation With 3LHS Sparse Prior

Recall the CS model with an uncertain sensing matrix

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x} + \boldsymbol{w}. \tag{9}$$

Let $p(\boldsymbol{\theta})$, $p(\boldsymbol{\phi})$ and $p(\boldsymbol{\kappa})$ denote the known (or assumed) prior distributions of the sensing parameter, prior parameter and noise precision $\boldsymbol{\kappa} = [\kappa_1, \ldots, \kappa_M]^T$. Our primary goal is to estimate the sparse signal $\boldsymbol{x}$, support $\boldsymbol{s}$, and the uncertain parameters $\boldsymbol{\xi} = [\boldsymbol{\theta}; \boldsymbol{\phi}; \boldsymbol{\kappa}]$, given the observations $\boldsymbol{y}$. In particular, for given $\boldsymbol{\xi}$, we are interested in computing the conditional marginal posteriors $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\xi})$ and $p(s_i|\boldsymbol{y}, \boldsymbol{\xi}), \forall i$ (i.e., perform Bayesian inference for $\boldsymbol{x}$ and $s_i, \forall i$), where

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\xi}) &\propto \sum_{\boldsymbol{s}} \int p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{s}|\boldsymbol{\xi})\, d\boldsymbol{\rho} \\ &= \sum_{\boldsymbol{s}} \int p(\boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{s}|\boldsymbol{\phi})\, p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\xi})\, d\boldsymbol{\rho}, \end{aligned} \tag{10}$$

$$p(s_i|\boldsymbol{y}, \boldsymbol{\xi}) \propto \sum_{\boldsymbol{s}_{-i}} \iint p(\boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{s}|\boldsymbol{\phi})\, p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\xi})\, d\boldsymbol{\rho} d\boldsymbol{x}, \tag{11}$$

where $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\xi}) = \mathcal{CN}\left(\boldsymbol{y}; \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}, \mathrm{diag}^{-1}(\boldsymbol{\kappa})\right)$. We denote equality after scaling as $\propto$. $\boldsymbol{s}_{-i}$ denotes $\{s_{i'}, \forall i' \neq i\}$. The uncertain parameters are obtained by MAP estimation as

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\mathrm{argmax}} \ln p(\boldsymbol{\xi}|\boldsymbol{y}) = \underset{\boldsymbol{\xi}}{\mathrm{argmax}} \ln \sum_{\boldsymbol{s}} \iint p(\boldsymbol{y}, \boldsymbol{v}, \boldsymbol{\xi})\, d\boldsymbol{\rho} d\boldsymbol{x}, \tag{12}$$

where $\boldsymbol{v} = \{\boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{s}\}$ is the collection of variables. Once we obtain the MAP estimate of $\boldsymbol{\xi}$, and the associated conditional marginal posteriors $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\xi}^*)$, $p(s_i|\boldsymbol{y}, \boldsymbol{\xi}^*), \forall i$, we can obtain the conditional MAP estimate of $\boldsymbol{x}$ and s_i (conditioned on $\boldsymbol{\xi} = \boldsymbol{\xi}^*$) as $\boldsymbol{x}^* = \mathrm{argmax}_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\xi}^*)$ and $s_i^* = \mathrm{argmax}_{s_i} p(s_i|\boldsymbol{y}, \boldsymbol{\xi}^*)$.

It is very challenging to calculate the exact posterior in (10) because the factor graph of the underlying model in (10) has loops. In the next section, we shall propose a Turbo-VBI algorithm which approximately calculates the marginal posteriors $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\xi})$ and $p(s_i|\boldsymbol{y}, \boldsymbol{\xi}), \forall i$ and finds an approximate solution for (12). The proposed Turbo-VBI algorithm is shown in the simulations to achieve a good performance.

The above CS problem formulation embraces many applications. In the following, we give two application examples.

A. High-Mobility Massive MIMO Channel Estimation

Consider a massive MIMO system with one static single-antenna base station (BS) serving a fast-moving user,[1] where the user is equipped with an arbitrary antenna array comprised of $N \gg 1$ antennas and $N_b < N$ RF chains. The downlink channel vector $\tilde{\boldsymbol{h}}_i \in \mathbb{C}^{N\times 1}$ in the i-th symbol duration of the current frame can be represented as $\tilde{\boldsymbol{h}}_i = \sum_{n=1}^{L} \tilde{x}_n \boldsymbol{a}_R(\theta_n) e^{j2\pi f_d i cos\theta_n}$, where there are L paths, θ_n is the angles of arrival (AoA) of the n-th path, $\boldsymbol{a}_R(\theta_n) \in \mathbb{C}^{N\times 1}$ is the array response vector, $\tilde{x}_n$ is the complex gain of the n-th path, f_d is the maximum Doppler offset. Note that, we have assumed that the parameters $L, \tilde{x}_n, \theta_n, f_d$ are fixed within each frame but may change over different frames. However, the channel $\tilde{\boldsymbol{h}}_i$ itself may change over different symbols due to the term $2\pi f_d i cos\theta_n$. For simplicity, we consider the widely used discrete channel model [22], where the AoAs of the user are assumed to take values from a discrete set: $\{\theta_{R,n} : sin(\theta_{R,n}) = \frac{2}{\tilde{N}}\left(n - \left\lfloor \frac{\tilde{N}-1}{2} \right\rfloor\right)\}$, where $\tilde{N} \geq N$ are the total number of discrete AoAs. In this case, the vector $\tilde{\boldsymbol{h}}_i$ can be represented as $\tilde{\boldsymbol{h}}_i = \boldsymbol{F}_i(f_d)\boldsymbol{h}$, where $\boldsymbol{h} = [x_1, \ldots, x_{\tilde{N}}]^T$ is the angular domain channel, and $\boldsymbol{F}_i(f_d) = [\boldsymbol{a}_R(\theta_{R,1}) e^{j2\pi f_d i cos(\theta_{R,1})}, \ldots, \boldsymbol{a}_R(\theta_{R,\tilde{N}}) e^{j2\pi f_d i cos(\theta_{R,\tilde{N}})}]$ is the array response matrix with Doppler offset f_d.

To recover the vector $\tilde{\boldsymbol{h}}_i$, the BS transmits N_p training symbols $p_i, i \in \mathcal{N}_p$, which are uniformly distributed in the current frame, where $\mathcal{N}_p$ denotes the index set of the training symbols. The user employs $\mathbf{U}_i \in \mathbb{C}^{N\times N_b}$ as a combining matrix. Then the received training signal $\boldsymbol{y} \in \mathbb{C}^{N_p N_b \times 1}$ in the current frame can be written as

$$\boldsymbol{y} = \left[\mathbf{U}_i^H \boldsymbol{F}_i(f_d) p_i \boldsymbol{h} + \mathbf{U}_i^H \boldsymbol{w}_i\right]_{i\in\mathcal{N}_p} = \boldsymbol{A}\boldsymbol{h} + \boldsymbol{w}, \tag{13}$$

where $\boldsymbol{w}_i \sim CN(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ is the Gaussian noise, $\boldsymbol{A}_i(f_d) = \mathbf{U}_i^H \boldsymbol{F}_i(f_d) p_i$, $\boldsymbol{A} = [\boldsymbol{A}_i(f_d)]_{i\in\mathcal{N}_p}$, and $\boldsymbol{w} = \left[\mathbf{U}_i^H \boldsymbol{w}_i\right]_{i\in\mathcal{N}_p}$, where $[\mathbf{x}_n]_{n\in\mathcal{S}}$ denotes a column vector consisting of the elements of $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ indexed by the set $\mathcal{S}$.

Let $\boldsymbol{s}$ denote the support vector of the angular domain channel $\boldsymbol{h}$. In practice, only the signals reflected/scattered from a few scattering clusters at the BS side can reach the receiver. As a result, the non-zero elements of the channel support vector $\boldsymbol{s}$ concentrate on a few clusters [3], [12]. Such cluster structure can be modeled using a Markov chain as

$$p(\boldsymbol{s}|\boldsymbol{\phi}) = p(s_1) \prod_{n=2}^{\tilde{N}} p(s_n|s_{n-1}), \tag{14}$$

with the transition probability given by

$$p(s_n|s_{n-1}) = \begin{cases} (1-p_{01})^{1-s_n} p_{01}^{s_n} & s_{n-1} = 0 \\ p_{10}^{1-s_n} (1-p_{10})^{s_n} & s_{n-1} = 1, \end{cases} \tag{15}$$

where the uncertain prior parameter $\boldsymbol{\phi} = \{\lambda, p_{01}\}$. Note that once λ and p_{01} are obtained, p_{10} can be determined from the relation $\frac{p_{01}}{p_{01}+p_{10}} = \lambda$, as explained below. The Markov parameters p_{01} and p_{10} determine the average cluster size

[1] In the high-speed railway communication scenario, it is possible that the BS has a small scale antenna and the user (high-speed train) has a massive MIMO array [21]. We consider single-antenna BS only for the purpose of easy illustration. The proposed Turbo-VBI approach also works for the case with massive MIMO arrays at both the BS and user.

and the average gap between two clusters in $\boldsymbol{x}$. The initial distribution $p(s_1)$ is set to be the steady state distribution, i.e., $p(s_1 = 1) \triangleq \lambda$, where λ reflects the sparsity of $\boldsymbol{x}$. In practice, we usually do not have exact knowledge about the sparsity λ.

The goal of massive MIMO channel estimation is to recover the sparse angular domain channel $\boldsymbol{h}$ and Doppler offset f_d from the received signal $\boldsymbol{y}$. This is an instance of the CS problem formulation in (9), where $Q = \tilde{N}$ (i.e., we do not impose group sparsity), the sensing matrix $\boldsymbol{A}$ contains an uncertain Doppler parameter f_d, the layer 3 variable (sparse signal) is the angular domain channel $\boldsymbol{h}$, the layer 2 variable (precision vector) represents the pathloss (inverse of the average channel gain), and the prior distribution of the layer 1 variable (support vector) $p(\boldsymbol{s})$ is chosen to be the Markov sparse prior in (14) with uncertain parameters $\boldsymbol{\phi} = \{\lambda, p_{01}\}$.

For uniform linear array (ULA) and negligible Doppler effect (i.e., $f_d = 0$), the sensing matrix $\boldsymbol{A}$ is partial orthogonal and an AMP-based algorithm (Turbo-CS) is used in [12] to recover the sparse channel $\boldsymbol{h}$ with the support-based Markov prior as described in (3) and (14). However, for an arbitrary antenna array with non-orthogonal array response matrix, the Turbo-CS performs poorly, as shown in the simulations. On the other hand, the conventional VBI cannot fully exploit the Markov sparse structure of massive MIMO channel. Therefore, it is better to use the proposed Turbo-VBI for high-mobility massive MIMO channel estimation with an arbitrary antenna array.

B. 5G-Based Localization for Autonomous Driving

One of the main challenges in autonomous driving is the requirement of both high-accuracy and robust localization of the vehicle. In order to achieve this challenging task, it is necessary to integrate multiple localization technologies such as Global Navigation Satellite Systems (GNSS), cameras or sensors [4]. The recent studies have shown the potential of integrating 5G-based localization technology into autonomous driving, thanks to the improved localization accuracy enabled by the massive MIMO technology [23]. In the following, we first give the system model for 5G-based localization. Then we provide a sparse representation for grid-based localization in the 5G-assisted autonomous driving system. Finally, based on the sparse representation, we formulate the vehicle localization problem as an instance of the CS problem in (9).

1) System Model: Consider a 2D geographical area $\mathcal{R}$. There is an autonomous vehicle, which is equipped with a 5G transceiver of one antenna. The vehicle is in the communication range of L massive MIMO BSs, each equipped with a massive array of $M \gg 1$ antennas. The L BSs are connected to a cloud. At the current time slot, the vehicle is located at $\boldsymbol{p} = [p^x, p^y]^T$ in $\mathcal{R}$, the center of the gravity of BSs' arrays are located at $\tilde{\boldsymbol{p}}_l = [\tilde{p}_l^x, \tilde{p}_l^y]^T$. We denote $\boldsymbol{a}_l(\theta) \in \mathbb{C}^{M\times 1}$ as the array response vector at BS l when the AoA is θ.

At each time slot, the vehicle transmits T pilot symbols $u(t)$ to the BSs to estimate its position. We consider flat fading channel. Specifically, there are $L_l + 1$ paths to the BS l. The

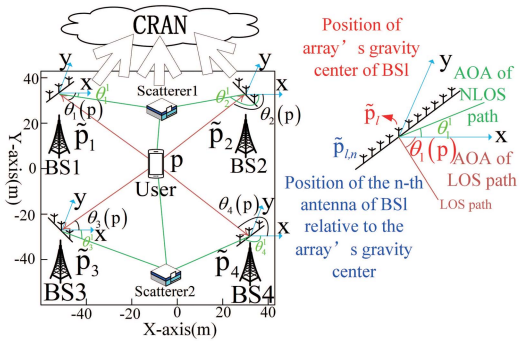


Fig. 2. A localization model in 5G massive MIMO system.

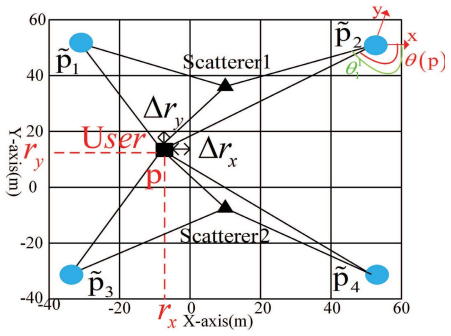


Fig. 3. Off-grid basis for localization.

received signal at BS l is given by [23]

$$\boldsymbol{y}_l(t) = \boldsymbol{a}_l(\theta_l(\boldsymbol{p}))\,\alpha_l u(t) + \sum_{k=1}^{L_l} \boldsymbol{a}_l\left(\theta_l^k\right)\alpha_l^k u(t) + \boldsymbol{w}_l(t), \tag{16}$$

where $\boldsymbol{w}_l(t) \sim \mathcal{CN}(\boldsymbol{0}, \sigma_l^2 \boldsymbol{I})$ is the Gaussian noise at BS l, α_l and θ_l stand for the complex channel gain and the AoA corresponding to the line-of-sight (LOS) path, α_l^k and θ_l^k stand for the complex channel gain and the AoA corresponding to the k-th NLOS path. The LOS AoA is related to the user location $\boldsymbol{p}$ through [23]

$$\theta_l(\boldsymbol{p}) = \arctan\left(\frac{p^y - \tilde{p}_l^y}{p^x - \tilde{p}_l^x}\right) + \pi \cdot 1\left(p^x < \tilde{p}_l^x\right), \tag{17}$$

which is computed with respect to the x-axis and anticlockwise. We model the noise precision $\kappa_l = \sigma_l^{-2}, \forall l$ as a Gamma hyperprior $p(\kappa_l) = \Gamma(\kappa; a_\kappa, b_\kappa)$, where we set $a_\kappa, b_\kappa \to 0$ as in [13] so as to obtain a broad hyperprior.

2) Sparse Representation for Grid-Based Localization : We adopt a grid-based method to locate the vehicle on a continuous map by exploiting the sparsity [23]. First, we introduce a uniform grid of $\widetilde{Q}$ locations

$$\mathcal{P} = \left\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{\widetilde{Q}}\right\} \subset \mathcal{R}. \tag{18}$$

and a uniform grid of $\widetilde{M}$ AoAs over $[0, 2\pi)$

$$\mathcal{A} = \left\{\vartheta_1, \ldots, \vartheta_{\widetilde{M}}\right\} \subset [0, 2\pi).$$

In practice, the true positions usually do not lie exactly on the grid point in $\mathcal{P}$, as shown in Fig. 3. In this case,

there will be mismatches between the true positions and the nearest grid point in $\mathcal{P}$. To overcome this issue, we introduce an off-grid basis for the sparse representation. Specifically, let $i^* = \text{argmin}_i \|\boldsymbol{p} - \boldsymbol{r}_i\|$ denote the index of the position grid point nearest to the vehicle position. We introduce a position offset vector $\triangle \boldsymbol{r} = \left[\triangle \boldsymbol{r}_1; \ldots; \triangle \boldsymbol{r}_{\widetilde{Q}}\right]$ such that $\triangle \boldsymbol{r}_i = \boldsymbol{p} - \boldsymbol{r}_{i^*}, i = i^*$ and $\triangle \boldsymbol{r}_i = 0, i \neq i^*$. Similarly, let $m_l^k = \text{argmin}_m \left\|\theta_l^k - \vartheta_m\right\|$ denote the index of the AoA grid point nearest to the AoA of the k-th NLOS path at BS l. For each channel between vehicle and BS l, we introduce an AoA offset vector $\triangle \boldsymbol{\vartheta}_l = \left[\triangle \vartheta_l^1, \ldots, \triangle \vartheta_l^{\widetilde{M}}\right]^T$ such that $\triangle \vartheta_l^m = \theta_l^k - \vartheta_{m_l^k}, k = 1, \ldots, L_l$ and $\triangle \vartheta_l^m = 0, \forall m \notin \left\{m_l^1, \ldots, m_l^{L_l}\right\}$.

With the above definitions of position offsets, we can define the offgrid basis for the LOS path at BS l (i.e., LOS array response matrix at BS l with position offset $\triangle \boldsymbol{r}$) as

$$\overline{\boldsymbol{A}}_l\left(\triangle \boldsymbol{r}\right) = \left[\boldsymbol{a}_l\left(\theta_l\left(\boldsymbol{r}_1 + \triangle \boldsymbol{r}_1\right)\right), \ldots, \boldsymbol{a}_n\left(\theta_l\left(\boldsymbol{r}_{\widetilde{Q}} + \triangle \boldsymbol{r}_{\widetilde{Q}}\right)\right)\right],$$

and the offgrid basis for the NLOS paths between vehicle and BS l (i.e., the NLOS array response matrix at BS l with AoA offset $\triangle \boldsymbol{\vartheta}_l$) as

$$\overline{\boldsymbol{A}}_l^{\text{NL}}\left(\triangle \boldsymbol{\vartheta}_l\right) = \left[\boldsymbol{a}_l\left(\vartheta_1 + \triangle \vartheta_l^1\right), \ldots, \boldsymbol{a}_l\left(\vartheta_{\widetilde{M}} + \triangle \vartheta_l^{\widetilde{M}}\right)\right].$$

Then the received signal at BS l in (16) could be rewritten as

$$\boldsymbol{y}_l\left(t\right) = u\left(t\right)\overline{\boldsymbol{A}}_l\left(\triangle \boldsymbol{r}\right)\boldsymbol{x}_l + u\left(t\right)\overline{\boldsymbol{A}}_l^{\text{NL}}\left(\triangle \boldsymbol{\vartheta}_l\right)\boldsymbol{z}_l + \boldsymbol{w}_l\left(t\right), \tag{19}$$

where $\boldsymbol{x}_l \in \mathbb{C}^{\widetilde{Q}}$ and $\boldsymbol{z}_l \in \mathbb{C}^{\widetilde{M}}$ are called the sparse LOS and NLOS channel vectors at BS l, respectively. To be more specific, the i-th entry of $\boldsymbol{x}_l$, denoted by $x_{l,i}$, represents the complex gain of a LOS path from the i-th off-grid location $\boldsymbol{r}_i + \triangle \boldsymbol{r}_i$ to BS l. The m-th entry of $\boldsymbol{z}_l$, denoted by $z_{l,m}$, represents the complex gain of a NLOS path at BS l with AoA $\vartheta_m + \triangle \vartheta_l^m$. Each $\boldsymbol{x}_l$ only has one non-zero element corresponding to the true position of the vehicle, and the index of the non-zero element of $\boldsymbol{x}_l, \forall l = 1, \ldots, L$ is identical. Similarly, each $\boldsymbol{z}_l$ only has $L_l \ll M$ non-zero elements corresponding to the AoAs of the L_l NLOS paths.

Since all LOS channel vectors $\boldsymbol{x}_l$'s share a common support corresponding to the true position, the sparse vector $\boldsymbol{x} = [\boldsymbol{x}_1; \ldots; \boldsymbol{x}_L]$ must obey a group sparsity [23] and we use $\boldsymbol{s} = \left[s_1, \ldots, s_{\widetilde{Q}}\right]^T$ to denote the common support vector of $\boldsymbol{x}_l, \forall l$, where $s_i = 1$ indicates that $x_{l,i}, l = 1, \ldots, L$ are active. We use $\boldsymbol{v}_l = \left[v_{l,1}, \ldots, v_{l,\widetilde{M}}\right]^T$ to denote the support vector of the NLOS channel vectors $\boldsymbol{z}_l, \forall l$, where $v_{l,i} = 1$ indicates that $z_{l,i}, \forall l$ is active. Since the non-zero elements of $\boldsymbol{z}_l$ often concentrate on a few clusters, $\boldsymbol{v}_l$ can be modeled using a Markov prior as in (22). In the above sparse representation, the coarse vehicle position $\boldsymbol{r}_{i^*}$ is determined by the index i^* of the non-zero element of the common support vector $\boldsymbol{s}$, and the mismatch between the coarse position $\boldsymbol{r}_{i^*}$ and the true position $\boldsymbol{p}$ is given by the corresponding position offset $\triangle \boldsymbol{r}_{i^*}$.

3) Localization Formulation With Statistical Prior Support Information: In practice, the vehicle position usually changes smoothly over time. The LOS channel support vector $\boldsymbol{s}$ also change slowly compared to the duration of time slot in wireless communications. As a result, the BSs can obtain some statistical PSI from previously estimated LOS channel supports [19]. In general, the statistical PSI is given by [19]

$$p\left(s_i = 1\right) = \alpha_i, \quad i = 1, \ldots, \widetilde{Q}, \tag{20}$$

where α_i indicates the probability of $s_i = 1$. The LOS support distribution $p\left(\boldsymbol{s}\right)$ is given by

$$p\left(\boldsymbol{s}\right) = \prod_{i=1}^{\widetilde{Q}} \alpha_i^{s_i}\left(1 - \alpha_i\right)^{1 - s_i}, \tag{21}$$

which incorporates the statistical PSI in the localization problem. On the other hand, the cluster structure of the vector $\boldsymbol{v}_l$ of the NLOS channel $\boldsymbol{z}_l$ can be modeled using a Markov chain

$$p\left(\boldsymbol{v}_l\right) = p\left(v_{l,1}\right)\prod_{i=2}^{\widetilde{M}} p\left(v_{l,i} | v_{l,i-1}\right) \tag{22}$$

The received signal in (19) can be expressed as

$$\boldsymbol{y}_l = \boldsymbol{A}_l\left(\triangle \boldsymbol{r}, \triangle \boldsymbol{\vartheta}_l\right)\begin{bmatrix}\boldsymbol{x}_l \\ \boldsymbol{z}_l\end{bmatrix} + \boldsymbol{w}_l, \quad \forall l, \tag{23}$$

where $\boldsymbol{w}_l = [\boldsymbol{w}_l\left(1\right); \ldots; \boldsymbol{w}_l\left(T\right)]$, $\boldsymbol{y}_l = [\boldsymbol{y}_l\left(1\right); \ldots; \boldsymbol{y}_l\left(T\right)]$, and $\boldsymbol{A}_l\left(\triangle \boldsymbol{r}, \triangle \boldsymbol{\vartheta}_l\right) \in \mathbb{C}^{MT \times (\widetilde{Q} + \widetilde{M})}$ is given by

$$\boldsymbol{A}_l\left(\triangle \boldsymbol{r}, \triangle \boldsymbol{\vartheta}_l\right) = \begin{bmatrix} u\left(1\right)\overline{\boldsymbol{A}}_l\left(\triangle \boldsymbol{r}\right) u\left(1\right)\overline{\boldsymbol{A}}_l^{\text{NL}}\left(\triangle \boldsymbol{\vartheta}_l\right) \\ \vdots \qquad\qquad\qquad \vdots \\ u\left(T\right)\overline{\boldsymbol{A}}_l\left(\triangle \boldsymbol{r}\right) \qquad u\left(T\right)\overline{\boldsymbol{A}}_l^{\text{NL}}\left(\triangle \boldsymbol{\vartheta}_l\right) \end{bmatrix}. \tag{24}$$

The goal of localization is to recover the LOS channel support vector $\boldsymbol{s}$ (which indicates the coarse position) and the position offset $\triangle \boldsymbol{r}$ from the received signal $\boldsymbol{y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L]$ with uncertain sensing matrix $\boldsymbol{A}\left(\triangle \boldsymbol{r}, \triangle \boldsymbol{\vartheta}\right) = \text{BlockDiag}\left[\boldsymbol{A}_1\left(\triangle \boldsymbol{r}, \triangle \boldsymbol{\vartheta}_1\right), \ldots, \boldsymbol{A}_L\left(\triangle \boldsymbol{r}, \triangle \boldsymbol{\vartheta}_L\right)\right]$. This is an instance of the CS problem formulation in (9), where $Q = \widetilde{Q} + \widetilde{M}L$ with the first $\widetilde{Q}$ groups given by $\mathcal{I}_i = \left\{(l-1)\widetilde{Q} + i : \ l = 1, \ldots, L\right\}, i = 1, \ldots, \widetilde{Q}$ and the rest groups $\mathcal{I}_i, i = \widetilde{Q} + 1, \ldots, \widetilde{Q} + \widetilde{M}L$ only containing one element, the sensing matrix $\boldsymbol{A}$ contains uncertain parameters $\triangle \boldsymbol{r}$ and $\triangle \boldsymbol{\vartheta}$, the layer 3 variable is the LOS channel $\boldsymbol{x}$ and the NLOS channel $\boldsymbol{z}_l$'s, and the prior distribution of the layer 1 variable $p\left(\boldsymbol{s}\right)$ and $p\left(\boldsymbol{v}_l\right)$ is given in (21) and (22).

In the above localization problem, the sensing matrix $\boldsymbol{A}$ has correlated columns and thus the performance of the AMP-based algorithm is poor. Although the conventional VBI can also be applied to the localization problem, it cannot incorporate the statistical PSI in (20) and the Markov structure in (22) and thus the performance is worse than the proposed Turbo-VBI.

IV. TURBO-VBI ALGORITHM

The basic idea of Turbo-VBI is to simultaneously approximate the intractable posterior $p(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi})$ with a tractable variational distribution $q(\boldsymbol{v};\boldsymbol{\xi})$ and maximize the marginal posterior $\ln p(\boldsymbol{y},\boldsymbol{\xi})$ with respect to the uncertain parameter $\boldsymbol{\xi}$ as in (12). In summary, the Turbo-VBI algorithm performs iterations between the following two major steps until convergence.

- **Turbo-VBI-E Step:** For given $\boldsymbol{\xi}$, evaluate $q(\boldsymbol{v};\boldsymbol{\xi})$ to approximate the posterior $p(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi})$ by combining the message passing and VBI approaches via the turbo framework, as will be elaborated in Section IV-C and IV-D;
- **Turbo-VBI-M Step:** Given $q(\boldsymbol{v};\boldsymbol{\xi}) \approx p(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi})$, construct a surrogate function for function $\ln p(\boldsymbol{y},\boldsymbol{\xi})$, partition $\boldsymbol{\xi}$ into B blocks $\boldsymbol{\xi}=(\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_B)$, then alternatively maximize the surrogate function with respect to $\boldsymbol{\xi}_j$, as will be elaborated in Section IV-A.

We first elaborate the Turbo-VBI-M step, which is an extension of the in-exact block majorization-minimization (MM) method in [24]. Then we show how to construct the surrogate function, which requires the posterior $p(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi})$. Finally, we elaborate how to approximately calculate the posterior $p(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi})$ in the Turbo-VBI-E Step.

A. Turbo-VBI-M Step (In-Exact Block MM)

It is difficult to directly maximize $\ln p(\boldsymbol{y},\boldsymbol{\xi})$ because there is no closed-form expression due to the multi-dimensional integration over $\boldsymbol{x},\boldsymbol{\rho},\boldsymbol{s}$. To make the problem tractable, in the Turbo-VBI-M Step, we first properly partition $\boldsymbol{\xi}$ into B blocks $\boldsymbol{\xi}=(\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_B)$, such that the resultant subproblem in (28) w.r.t. each block can be solved efficiently (e.g., has a closed-form or low-complexity solution). In many cases, $\boldsymbol{\xi}$ consists of several subsets of parameters $\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_B$, where each subset $\boldsymbol{\xi}_j$ has a distinct physical meaning. In this case, $\boldsymbol{\xi}$ can be naturally partitioned into blocks $\boldsymbol{\xi}=(\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_B)$ according to the physical meaning of each $\boldsymbol{\xi}_j$. Then, we alternatively maximize a surrogate function of $\ln p(\boldsymbol{y},\boldsymbol{\xi})$ with respect to each $\boldsymbol{\xi}_j, j\in\{1,\ldots,B\}$.

The surrogate function is chosen such that the alternating maximization w.r.t. each variable $\boldsymbol{\xi}_j$ has a closed-form solution. Let $u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ be the surrogate function constructed at fixed point $\dot{\boldsymbol{\xi}}$, which satisfies the following properties:

$$u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right) \le \ln p(\boldsymbol{y},\dot{\boldsymbol{\xi}}), \quad \forall\boldsymbol{\xi}, \tag{25}$$

$$u\left(\dot{\boldsymbol{\xi}};\dot{\boldsymbol{\xi}}\right) = \ln p(\boldsymbol{y},\dot{\boldsymbol{\xi}}), \tag{26}$$

$$\frac{\partial u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)}{\partial\boldsymbol{\xi}}|_{\boldsymbol{\xi}=\dot{\boldsymbol{\xi}}} = \frac{\partial \ln p\left(\mathbf{y},\boldsymbol{\xi}\right)}{\partial\boldsymbol{\xi}}|_{\boldsymbol{\xi}=\dot{\boldsymbol{\xi}}}. \tag{27}$$

In the Turbo-VBI-M Step of the i-th iteration, we update $\boldsymbol{\xi}_j$ alternatively for $j=1,\ldots,B$ as

$$\boldsymbol{\xi}_j^{(i+1)} = \underset{\boldsymbol{\xi}_j}{\operatorname{argmax}}\, u\left(\boldsymbol{\xi}_j,\boldsymbol{\xi}_{-j}^{(i)};\boldsymbol{\xi}_j^{(i)},\boldsymbol{\xi}_{-j}^{(i)}\right), \tag{28}$$

where $(\cdot)^{(i)}$ stands for the i_{th} iteration, $\boldsymbol{\xi}_{-j}^{(i)} = \left(\boldsymbol{\xi}_1^{(i+1)},\ldots,\boldsymbol{\xi}_{j-1}^{(i+1)},\boldsymbol{\xi}_{j+1}^{(i)},\ldots,\boldsymbol{\xi}_B^{(i)}\right)$. The update rule in (28) guarantees the convergence of the algorithm to a stationary point of (12) [24]. The initial value of $\boldsymbol{\xi}$ is set according to the specific application scenario based on the available prior knowledge of $\boldsymbol{\xi}$. If it is difficult to find the global optimal solution of (28) for some $j\in\overline{\mathcal{J}}_c\subseteq\{1,\ldots,B\}$ (e.g., when $u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right), \forall j\in\overline{\mathcal{J}}_c$ is non-convex w.r.t. $\boldsymbol{\xi}_j$), we can partition the index set $\{1,\ldots,B\}$ into two subsets $\overline{\mathcal{J}}_c$ and $\mathcal{J}_c=\{1,\ldots,B\}\setminus\overline{\mathcal{J}}_c$ such that for $j\in\mathcal{J}_c$, $u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ is strongly convex w.r.t. $\boldsymbol{\xi}_j$, while for $j\in\overline{\mathcal{J}}_c$, we do the following gradient update:

$$\boldsymbol{\xi}_j^{(i+1)} = \boldsymbol{\xi}_j^{(i)} + \gamma^{(i)}\frac{\partial u\left(\boldsymbol{\xi}_j,\boldsymbol{\xi}_{-j}^{(i)};\boldsymbol{\xi}_j^{(i)},\boldsymbol{\xi}_{-j}^{(i)}\right)}{\partial\boldsymbol{\xi}_j}|_{\boldsymbol{\xi}_j=\boldsymbol{\xi}_j^{(i)}}, \tag{29}$$

where $\gamma^{(i)}$ is the step size determined by the Armijo rule [25].

In the original in-exact block MM method for channel estimation in [24], there is only one non-convex block (i.e., $\left|\overline{\mathcal{J}}_c\right|=1$), and the solution of maximizing the surrogate function over each convex block in (28) for all $j\in\mathcal{J}_c$ is unique. The convergence proof in [24] also replies on this fact. However, in the more general problem considered in this paper, it is possible that there are multiple non-convex blocks (i.e., $\left|\overline{\mathcal{J}}_c\right|>1$). As a result, the convergence proof in [24] cannot be applied to our problem. To address this challenge, we impose an additional condition that $u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ must be strongly convex w.r.t. $\boldsymbol{\xi}_j, \forall j\in\mathcal{J}_c$, and obtain the following convergence theorem for the above Turbo-VBI algorithm. Please refer to Appendix A for the detailed proof.

Theorem 1 (Convergence of In-Exact MM): Suppose the surrogate function $u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ satisfies (25) - (27) and it is strongly convex w.r.t. $\boldsymbol{\xi}_j, \forall j\in\mathcal{J}_c$. If at each iteration, we do the exact update as in (28) for $j\in\mathcal{J}_c$, and in-exact (gradient) update as in (29) for $j\in\overline{\mathcal{J}}_c$, the iterates generated by the Turbo-VBI algorithm converge to a stationary point of Problem (12).

Curious readers may wonder how the Turbo-VBI-E step plays a role in the convergence proof. It turns out that in order to construct a surrogate function $u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ that satisfies the conditions in (25) - (27), we need to obtain the posterior $p(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi})$ using the Turbo-VBI-E step. Therefore, the Turbo-VBI-E step is implicitly required in the construction of surrogate function, as explained in the next subsection.

B. EM-Based Surrogate Function

Inspired by the EM method [16], we use the following surrogate function:

$$u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right) = u^{\mathrm{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right) + \sum_{j\in\mathcal{J}_c^1}\tau_j\left\|\boldsymbol{\xi}_j-\dot{\boldsymbol{\xi}}_j\right\|^2, \tag{30}$$

where $u^{\mathrm{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right) = \int p\left(\boldsymbol{v}|\boldsymbol{y},\dot{\boldsymbol{\xi}}\right)\ln\frac{p(\boldsymbol{v},\boldsymbol{y},\boldsymbol{\xi})}{p\left(\boldsymbol{v}|\boldsymbol{y},\dot{\boldsymbol{\xi}}\right)}d\boldsymbol{v}$ is the EM surrogate function used in [24], $\mathcal{J}_c^1\subseteq\{1,\ldots,B\}$ is the index set such that $u^{\mathrm{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ is convex but not strongly convex w.r.t. $\boldsymbol{\xi}_j, \forall j\in\mathcal{J}_c^1$, and $\tau_j>0$ can be any constant. The second term is added to ensure that (30) is strongly convex w.r.t. $\boldsymbol{\xi}_j, \forall j\in\mathcal{J}_c$, where $\mathcal{J}_c$ is the index set such that

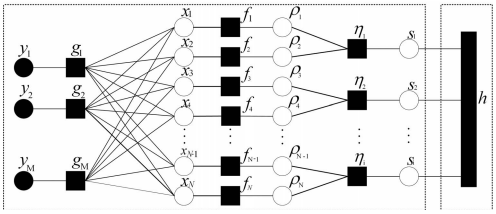


Fig. 4. Factor graph of the joint distribution in (10). For easy illustration, we assume every two adjacent elements of the sparse signal $\boldsymbol{x}$ form a group, i.e., $Q = N/2$ and $\mathcal{I}_i = \{2i-1, 2i\}$.

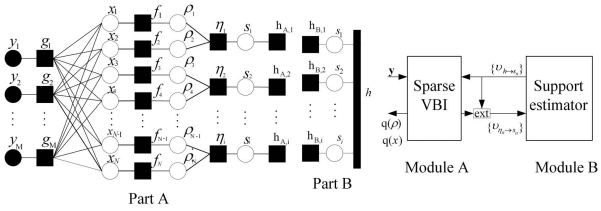


Fig. 5. Modules of the Turbo-VBI algorithm and message flows between different modules.

$u^{\mathrm{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ is convex w.r.t. $\boldsymbol{\xi}_j, \forall j \in \mathcal{J}_c$. It can be shown that the surrogate function in (30) satisfies (25) - (27). Therefore, if we can calculate the exact posterior $p\left(\boldsymbol{v}|\boldsymbol{y},\dot{\boldsymbol{\xi}}\right)$ for given $\dot{\boldsymbol{\xi}}$, we can construct the surrogate function in (30) and the corresponding Turbo-VBI algorithm converges to a stationary point of (12). Unfortunately, in many cases, the exact posterior $p\left(\boldsymbol{v}|\boldsymbol{y},\dot{\boldsymbol{\xi}}\right)$ is intractable. Thus, we propose to combine the message passing and VBI approaches via the turbo framework to find an alternative probability density function $q\left(\boldsymbol{v};\dot{\boldsymbol{\xi}}\right)$ to approximate the posterior $p\left(\boldsymbol{v}|\boldsymbol{y},\dot{\boldsymbol{\xi}}\right)$ for any given $\dot{\boldsymbol{\xi}}$ in the E Step, which is expected to be close to the true posterior [15]. Then we construct a tractable surrogate function as

$$\hat{u}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right) = \hat{u}^{\mathrm{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right) + \sum_{j\in\hat{\mathcal{J}}_c^1} \tau_j \left\|\boldsymbol{\xi}_j - \dot{\boldsymbol{\xi}}_j\right\|^2, \qquad (31)$$

where $\hat{u}^{\mathrm{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right) = \int q\left(\boldsymbol{v};\dot{\boldsymbol{\xi}}\right) \ln \frac{p(\boldsymbol{v},\boldsymbol{y},\boldsymbol{\xi})}{q\left(\boldsymbol{v};\dot{\boldsymbol{\xi}}\right)} d\boldsymbol{v}$ is an approximation of $u^{\mathrm{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$, $\hat{\mathcal{J}}_c^1$ is defined based on the convexity of $\hat{u}^{\mathrm{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ w.r.t. each block similar to $\mathcal{J}_c^1$. Since the posterior approximation $q\left(\boldsymbol{v};\dot{\boldsymbol{\xi}}\right)$ obtained in the E Step is usually good enough [15], $\hat{u}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ is expected to approximately satisfy (25) - (27), and thus such approximation has little effect on the convergence of the proposed algorithm, as verified in the simulations. Therefore, after the convergence of the Turbo-VBI with the tractable surrogate function in (31), we not only obtain an approximate stationary solution $\hat{\boldsymbol{\xi}}$ of (12), but also the associated (approximate) conditional marginal posteriors $p\left(\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\xi}}\right) \approx q\left(\boldsymbol{x};\hat{\boldsymbol{\xi}}\right)$ and $p\left(s_i|\boldsymbol{y},\hat{\boldsymbol{\xi}}\right) \approx q\left(s_i;\hat{\boldsymbol{\xi}}\right), \forall i$.

C. Modules of the Turbo-VBI-E Step

The factor graph of the joint distribution $p\left(\boldsymbol{y},\boldsymbol{v}|\boldsymbol{\xi}\right)$, denoted by $\mathcal{G}$, is shown in Fig. 4, where the function expression of each factor node is listed in Table I. Since $\mathcal{G}$ is a dense graph with many loops, directly applying the sum-product message passing (SPMP) [26] over the entire factor graph $\mathcal{G}$ usually cannot achieve a good performance. When the sensing matrix is i.i.d. Gaussian or partial orthogonal, Turbo-AMP [11] and Turbo-CS [12] algorithms can be used to achieve approximate message passing over dense graphs. However, in our problem, the sensing matrix $\boldsymbol{A}(\boldsymbol{\theta})$ can be ill conditioned and the performance of Turbo-AMP or Turbo-CS is very poor, as will be shown in the simulations. To overcome this challenge, we combine the VBI [15] and message passing approaches [26] via the turbo framework to design the Turbo-VBI-E step that can achieve approximate message passing over $\mathcal{G}$ with a good performance.

Specifically, we follow the turbo framework and partition the factor graph $\mathcal{G}$ into two parts, as shown in Fig. 5, where Part A contains the dense subgraph $\mathcal{G}_x$ (as well as a copy of the support vector $\boldsymbol{s}$ and an additional set of factor nodes $h_{A,i}$), and Part B contains subgraph $\mathcal{G}_s$ (and an additional set of factor nodes $h_{B,i}$). Correspondingly, the Turbo-VBI-E step has two modules to perform Bayesian inference over Part A and Part B, respectively. Moreover, Module A and B also need to exchange messages, as shown in Fig. 5. In particular, the messages $\{\nu_{\eta_i\to s_i}(\cdot)\}$ form the outputs of Module A and the inputs of Module B, while the messages $\{\nu_{h\to s_i}(\cdot)\}$ form the outputs of Module B and the inputs of Module A. The two modules are executed iteratively until convergence. In the following, we elaborate Module A and Module B.

Based on the observation $\boldsymbol{y}$ and messages $\{\nu_{h\to s_i}(\cdot)\}$ from Module B, Module A performs the sparse VBI [15] to calculate the approximate conditional marginal posteriors. To be more specific, the factor nodes

$$h_{A,i}(s_i) \triangleq \nu_{h\to s_i}(s_i), \quad i = 1,\ldots,Q,$$

incorporate the prior information from Module B. Therefore, the following prior distribution is assumed when performing the sparse VBI in Module A:

$$\begin{aligned}\hat{p}(\boldsymbol{x},\boldsymbol{\rho},\boldsymbol{s}) &= \hat{p}(\boldsymbol{s})\, p(\boldsymbol{\rho}|\boldsymbol{s})\, p(\boldsymbol{x}|\boldsymbol{\rho}),\\ \hat{p}(\boldsymbol{s}) &= \prod_i (\pi_i)^{s_i}(1-\pi_i)^{1-s_i}, \end{aligned} \qquad (32)$$

where

$$\pi_i \triangleq \hat{p}(s_i = 1) = \frac{\nu_{h\to s_i}(1)}{\nu_{h\to s_i}(1) + \nu_{h\to s_i}(0)}.$$

Note that the only difference between the prior in (32) and the original prior in (6) is that the prior distribution $p\left(\boldsymbol{s}|\boldsymbol{\phi}\right)$ of the support vector is replaced with a prior $\hat{p}\left(\boldsymbol{s}\right)$ with independent entries. The corresponding posterior distribution of $\boldsymbol{x}$ obtained by the sparse VBI is complex Gaussian, as will be given in (37), and the posterior distribution $q\left(s_i\right)$ of s_i will be given by (43). After that, the messages $\{\nu_{\eta_i\to s_i}(s_i)\}$ from Module A to Module B can be calculated from the posterior distribution $q\left(s_i\right)$ by subtracting the input message $\{\nu_{h\to s_i}(s_i)\}$ as [10]

$$\nu_{\eta_i\to s_i}(s_i) = \frac{q(s_i)}{\nu_{s_i\to\eta_i}(s_i)}, \qquad (33)$$

TABLE I
FACTORS, DISTRIBUTIONS AND FUNCTIONAL FORMS IN FIG. 4. $\boldsymbol{A}_m(\boldsymbol{\theta})$ DENOTES THE m-TH ROW OF $\boldsymbol{A}(\boldsymbol{\theta})$

Factor	Distribution	Functional form
$g_m(y_m, \boldsymbol{x})$	$p(y_m\|\boldsymbol{x}, \boldsymbol{\xi})$	$\mathcal{CN}(y_m; \boldsymbol{A}_m(\boldsymbol{\theta})\boldsymbol{x}, \kappa_m^{-1})$
$f_n(x_n, \rho_n)$	$p(x_n\|\rho_n)$	$\prod_{n\in\mathcal{I}_i}\mathcal{CN}(x_n; 0, \rho_n^{-1})$
$\eta_i(\boldsymbol{\rho}[\mathcal{I}_i], s_i)$	$p(\boldsymbol{\rho}[\mathcal{I}_i]\|s_i)$	$\prod_{n\in\mathcal{I}_i}(\Gamma(\rho_n; a_n, b_n))^{s_i}\left(\Gamma\left(\rho_n; \overline{a}_n, \overline{b}_n\right)\right)^{1-s_i}$
$h(\boldsymbol{s})$	$p(\boldsymbol{s})$	depends on application

where the denominator of (33) equals to $\nu_{h\to s_i}(s_i)$ according to the sum product rule.

Based on the messages $\{\nu_{\eta_i\to s_i}(\cdot)\}$ from Module A, Module B further exploits the structured sparsity as captured by the prior distribution $p(\boldsymbol{s})$ of the support vector to improve the estimation performance, by performing the SPMP algorithm [26] over the support subgraph $\mathcal{G}_s$ in Part B. To be more specific, in Part B, the factor nodes

$$h_{B,i}(s_i) \triangleq \nu_{\eta_i\to s_i}(s_i), \quad i = 1, \ldots, Q$$

incorporate the prior information from Module A, and the factor node h incorporates the structured sparsity. After that, the messages $\{\nu_{h\to s_i}(\cdot)\}$ from Module B to Module A can be calculated according to the sum-product rule.

D. Sparse VBI Estimator (Module A)

1) Outline of Sparse VBI: For convenience, we use $\boldsymbol{v}^k$ to denote an individual variable in $\boldsymbol{v}$. Let $\mathcal{H} = \{k|\forall \boldsymbol{v}^k \in \boldsymbol{v}\}$. Based on the VBI method, the approximate conditional marginal posterior could be calculated by minimizing the Kullback-Leibler divergence (KLD) [15] between $\hat{p}(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\xi})$ and $q(\boldsymbol{v}; \boldsymbol{\xi})$, subject to a factorized form constraint as

$$\mathscr{A}_{\mathrm{VBI}}: \quad q^*(\boldsymbol{v}; \boldsymbol{\xi}) = \arg\min_{q(\boldsymbol{v};\boldsymbol{\xi})} \int q(\boldsymbol{v}; \boldsymbol{\xi}) \ln \frac{q(\boldsymbol{v}; \boldsymbol{\xi})}{\hat{p}(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\xi})} d\boldsymbol{v} \tag{34}$$

$$\text{s.t.} \quad q(\boldsymbol{v}; \boldsymbol{\xi}) = \prod_{k\in\mathcal{H}} q\left(\boldsymbol{v}^k; \boldsymbol{\xi}\right), \quad \int q\left(\boldsymbol{v}^k; \boldsymbol{\xi}\right) d\boldsymbol{v}^k = 1, \tag{35}$$

where $\hat{p}(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\xi})$ is the posterior distribution of $\boldsymbol{v}$ with the prior $\hat{p}(\boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{s})$ in (32), and for discrete variable $\boldsymbol{s}$, $\int(\cdot)\, d\boldsymbol{v}^k$ means the summation over the set of all possible discrete values of $\boldsymbol{s}$. In this section, we will omit the argument $\boldsymbol{\xi}$ in $q(\boldsymbol{v}; \boldsymbol{\xi})$ for simplicity. Solving $\mathscr{A}_{\mathrm{VBI}}$ yields a good approximation of the true posterior $\hat{p}(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\xi})$ and such a VBI method has been widely used in Bayesian inference with great success [15]. Since Problem $\mathscr{A}_{\mathrm{VBI}}$ is non-convex, the uniqueness of the optimal solution may not be guaranteed. Fortunately, the existence of the optimal solution has been proved in [15]. In the following, we aim at finding a stationary solution (denoted by $q^*(\boldsymbol{v})$) of $\mathscr{A}_{\mathrm{VBI}}$, as defined below.

Definition 1 (Stationary Solution): $q^*(\boldsymbol{v}) = \prod_{k\in\mathcal{H}} q^*\left(\boldsymbol{v}^k\right)$ *is called a stationary solution of Problem* $\mathscr{A}_{\mathrm{VBI}}$ *if it satisfies all the constraints in* $\mathscr{A}_{\mathrm{VBI}}$ *and* $\forall k \in \mathcal{H}$,

$$q^*\left(\boldsymbol{v}^k\right) = \arg\min_{q(\boldsymbol{v}^k)} \int \prod_{l\neq k} q^*\left(\boldsymbol{v}^l\right) q\left(\boldsymbol{v}^k\right) \ln \frac{\prod_{l\neq k} q^*\left(\boldsymbol{v}^l\right) q\left(\boldsymbol{v}^k\right)}{\hat{p}(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\xi})}.$$

By finding a stationary solution $q^*(\boldsymbol{v})$ of $\mathscr{A}_{\mathrm{VBI}}$, we could obtain the approximate posterior $q^*\left(\boldsymbol{v}^k\right) \approx p\left(\boldsymbol{v}^k|\boldsymbol{y}, \boldsymbol{\xi}\right)$.

A stationary solution of $\mathscr{A}_{\mathrm{VBI}}$ can be obtained via alternately optimizing each individual density $q\left(\boldsymbol{v}^k\right), k \in \mathcal{H}$, as will be proved by Lemma 1. Specifically, for given $q\left(\boldsymbol{v}^l\right), \forall l \neq k$, the optimal $q\left(\boldsymbol{v}^k\right)$ that minimizes the KLD in $\mathscr{A}_{\mathrm{VBI}}$ is given by [15]

$$q\left(\boldsymbol{v}^k\right) \propto \exp\left(\left\langle \ln p(\boldsymbol{v}, \boldsymbol{y}|\boldsymbol{\xi})\right\rangle_{\prod_{l\neq k} q(\boldsymbol{v}^l)}\right), \tag{36}$$

where $\langle f(x)\rangle_{q(x)} = \int f(x)\, q(x) dx$. Based on (36), the update equations of all variables are given in the subsequent subsections. The derivation can be found in Appendix B. Note that the expectation $\left\langle f\left(\boldsymbol{v}^k\right)\right\rangle_{q(\boldsymbol{v}^k)}$ w.r.t. its own approximate posterior is simplified as $\left\langle f\left(\boldsymbol{v}^k\right)\right\rangle$.

2) Initialization of Sparse VBI: In order to trigger the alternating optimization (AO) algorithm, we use the following initializations for the distribution functions $q(\boldsymbol{s})$ and $q(\boldsymbol{\rho})$.

- In the first outer iteration, initialize $\hat{p}(\boldsymbol{s}) = \prod_{i=1}^{Q} \hat{p}(s_i)$ with $\hat{p}(s_i) = (\pi_i)^{s_i} \times (1-\pi_i)^{1-s_i}$. In the rest outer iterations, initialize $q(\boldsymbol{s}) = \prod_{i=1}^{Q} (\tilde{\pi}_i)^{s_i} (1-\tilde{\pi}_i)^{1-s_i}$, where $\tilde{\pi}_i$ is the posterior probability of $s = 1$ calculated from Module A in the previous iteration.
- Initialize a gamma distribution for $\boldsymbol{\rho}$: $q(\boldsymbol{\rho}) = \prod_{n=1}^{N} \Gamma\left(\rho_n; \tilde{a}_n, \tilde{b}_n\right)$, where $\tilde{a}_n = \pi_i a_n + (1-\pi_i)\overline{a}_n$, $\tilde{b}_n = \pi_i b_n + (1-\pi_i)\overline{b}_n$, $\forall n \in \mathcal{I}_i, \forall i$.

3) Update for $\boldsymbol{x}$: $q(\boldsymbol{x})$ can be derived as

$$q(\boldsymbol{x}) = \mathcal{CN}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{37}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be calculated through

$$\boldsymbol{\Sigma} = \left(\mathrm{diag}\left(\left\langle \frac{\tilde{a}_1}{\tilde{b}_1}, \ldots, \frac{\tilde{a}_N}{\tilde{b}_N}\right\rangle\right) + \boldsymbol{A}(\boldsymbol{\theta})^H \mathrm{diag}(\boldsymbol{\kappa})\, \boldsymbol{A}(\boldsymbol{\theta})\right)^{-1}, \tag{38}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma} \boldsymbol{A}(\boldsymbol{\theta})^H \mathrm{diag}(\boldsymbol{\kappa})\, \boldsymbol{y}. \tag{39}$$

4) Update for $\boldsymbol{\rho}$: $q(\boldsymbol{\rho})$ can be derived as

$$q(\boldsymbol{\rho}) = \prod_{n=1}^{N} \Gamma\left(\rho_n; \tilde{a}_n, \tilde{b}_n\right), \tag{40}$$

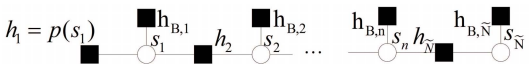


Fig. 6. Support subgraph of the massive MIMO channel estimation problem.

where the approximate posterior parameters are given by:

$$\tilde{a}_n = \langle s_i \rangle a_n + \langle 1 - s_i \rangle \overline{a}_n + 1 = \tilde{\pi}_i a_n + (1 - \tilde{\pi}_i) \overline{a}_n + 1, \tag{41}$$

$$\tilde{b}_n = \left\langle |x_n|^2 \right\rangle + \langle s_i \rangle b_n + \langle 1 - s_i \rangle \overline{b}_n = |\mu_n|^2 + \Sigma_n + \tilde{\pi}_i b_n + (1 - \tilde{\pi}_i) \overline{b}_n, , \forall n \in \mathcal{I}_i, \quad \forall i. \tag{42}$$

where μ_n is the n-th element of $\boldsymbol{\mu}$, and Σ_n is the n-th diagonal element of matrix $\boldsymbol{\Sigma}$.

5) Update for $\boldsymbol{s}$: $q(\boldsymbol{s})$ can be derived as

$$q(\boldsymbol{s}) = \prod_{i=1}^{Q} (\tilde{\pi}_i)^{s_i} (1 - \tilde{\pi}_i)^{1 - s_i}, \tag{43}$$

where $\tilde{\pi}_i$ is given by

$$\tilde{\pi}_i = \frac{1}{C} \prod_{n \in \mathcal{I}_i} \frac{\pi_i b_n^{a_n}}{\Gamma(a_n)} e^{(a_n - 1)\langle \ln \rho_n \rangle - b_n \langle \rho_n \rangle}, \tag{44}$$

and C is the normalization constant, given by $C = \prod_{n \in \mathcal{I}_i} \frac{\pi_i b_n^{a_n}}{\Gamma(a_n)} e^{(a_n - 1)\langle \ln \rho_n \rangle - b_n \langle \rho_n \rangle} + \prod_{n \in \mathcal{I}_i} \frac{(1 - \pi_i) \overline{b}_n^{\overline{a}_n}}{\Gamma(\overline{a}_n)} e^{(\overline{a}_n - 1)\langle \ln \rho_n \rangle - \overline{b}_n \langle \rho_n \rangle}$, $\langle \ln \rho_n \rangle = \psi(\tilde{a}_n) - \ln\left(\tilde{b}_n\right)$, $\psi(x) = \frac{d}{dx} \ln(\Gamma(x))$ is the digamma function, defined as the logarithmic derivative of the gamma function.

6) Convergence of Sparse VBI: The sparse VBI can be viewed as an AO method [27] to solve $\mathscr{A}_{\text{VBI}}$. It is clear that the sparse VBI can monotonically decreasing the objective value, thus the objective value will converge to a limit. Moreover, for given $q\left(\boldsymbol{v}^l\right), \forall l \neq k$, the optimal $q\left(\boldsymbol{v}^k\right)$ that minimizes the KLD in $\mathscr{A}_{\text{VBI}}$ is unique. Then according to the convergence of AO [27], we have the following convergence theorem.

Lemma 1 (Convergence of Sparse VBI): Every limiting point $q^*(\boldsymbol{v}) = \prod_{k \in \mathcal{H}} q^*\left(\boldsymbol{v}^k\right)$ generated by the sparse VBI is a stationary solution of Problem $\mathscr{A}_{\text{VBI}}$.

The overall algorithm is summarized in Algorithm 1.

Algorithm 1 Turbo-VBI Algorithm

Input: $\boldsymbol{y}$, prior distributions $p(\boldsymbol{\theta})$, $p(\boldsymbol{\phi})$ and $p(\boldsymbol{\kappa})$, $\mathbf{A}(\boldsymbol{\theta})$, maximum iteration number I_m, threshold ε.
Output: $\boldsymbol{\xi}^*$, $\boldsymbol{x}^*$, $\boldsymbol{s}_i^*, \forall i$.

```
Initialize the parameters ξ, and the message π_i.
for k = 1, ..., I_m do
  Turbo-VBI-E Step:
  %Module A: Sparse VBI Estimator
  Initialize the distribution functions q(s) and q(ρ).
  while not converge do
    Update q^k(x;ξ), q^k(ρ;ξ), q^k(s;ξ) .
  end while
  Calculate the extrinsic information of s_i based on (33), send ν_{η_i→s_i}(s_i) to Module B.
  % Module B:
  Perform the SPMP over the support subgraph G_s, send ν_{h→s_i}(s_i) to Module A.
  Turbo-VBI-M Step:
  Construct the surrogate function û in (31) using the output of Module A, i.e., q(v;ξ).
  Update ξ_j alternatively for j = 1, ..., B using (28) for j ∈ J_c and (29) for j ∈ J̄_c.
  if ||μ^(k-1) − μ^(k)|| ≤ ε and ||Σ^(k-1) − Σ^(k)|| ≤ ε then
    break
  end if
  k = k + 1.
end for
Output ξ*, x* = arg max_x q^k(x;ξ) = μ^k and s_i* = arg max_{s_i} q^k(s_i;ξ).
```

V. Applications

A. Massive MIMO Channel Estimation

1) Implementation Details: In this example, we have $Q = \tilde{N}$ and thus we also use n to index the elements in $\boldsymbol{s}$. For given messages $\{\nu_{\eta_n \to s_n}(s_n), \forall n\}$ from Module A, the support subgraph $\mathcal{G}_s$ for this application problem is a Markov chain as shown in Fig. 6. In $\mathcal{G}_s$, the factor nodes

$$h_{B,n}(s_n) \propto \left(\pi_n^{in}\right)^{s_n} \left(1 - \pi_n^{in}\right)^{1 - s_n}, \quad \forall n$$

incorporate the prior information from Module A, where

$$\pi_n^{in} = \frac{\nu_{\eta_n \to s_n}(1)}{\nu_{\eta_n \to s_n}(1) + \nu_{\eta_n \to s_n}(0)}.$$

The factor nodes $h_1(s_1) = p(s_1)$ and

$$h_n(s_{n-1}, s_n) = p(s_n | s_{n-1}), \quad \forall n = 2, \ldots, \tilde{N}$$

incorporate the clustered sparsity in the massive MIMO channel. Note that $h_n, \forall n$ reveal detailed structure of the factor node h associated with the support prior $p(\boldsymbol{s})$.

Since $\mathcal{G}_s$ is a Markov chain, we can use the forward-backward message passing algorithm in [26] to update the messages over $\mathcal{G}_s$. Specifically, we first conduct forward message passing. Let $\lambda_1^f = \lambda$, $\lambda_N^b = 1/2$ and update the forward messages as

$$\lambda_n^f = \frac{p_{01}\left(1 - \pi_{n-1}^{in}\right)\left(1 - \lambda_{n-1}^f\right) + p_{11}\pi_{n-1}^{in}\lambda_{n-1}^f}{\left(1 - \pi_{n-1}^{in}\right)\left(1 - \lambda_{n-1}^f\right) + \pi_{n-1}^{in}\lambda_{n-1}^f}. \tag{45}$$

Then we update the backward messages as

$$\lambda_n^b = \frac{p_{10}\left(1 - \pi_{n+1}^{in}\right)\left(1 - \lambda_{n+1}^b\right) + p_{11}\pi_{n+1}^{in}\lambda_{n+1}^b}{p_0\left(1 - \pi_{n+1}^{in}\right)\left(1 - \lambda_{n+1}^b\right) + p_1\pi_{n+1}^{in}\lambda_{n+1}^b}, \tag{46}$$

for $n = \tilde{N} - 1, \ldots, 2, 1$, where $p_0 = p_{00} + p_{10}$ and $p_1 = p_{11} + p_{01}$. After that, the messages $\{\nu_{h \to s_n} = \nu_{h_n \to s_n}\}$ from Module B to Module A is updated as

$$\nu_{h_n \to s_n}(s_n) = (\pi_n)^{s_n} (1 - \pi_n)^{1 - s_n}, \tag{47}$$

for $n = 1, \ldots, \tilde{N}$, where

$$\pi_n = \frac{\lambda_n^f \lambda_n^b}{\left(1 - \lambda_n^f\right)(1 - \lambda_n^b) + \lambda_n^f \lambda_n^b}. \tag{48}$$

In the Turbo-VBI-M step, we have $B = 3$ and $\xi_1 = \lambda$, $\xi_2 = p_{01}$, $\xi_2 = f_d$. The initial values $\xi_1 = \lambda$ and $\xi_2 = p_{01}$ are set to small values ($\xi_1 = \xi_2 = 0.01$) since the massive MIMO channel is expected to be sparse, and the initial value of $\xi_3 = f_d$ is calculated according to the phase rotation of the received signals between the two adjacent pilot symbols. The maximization step to obtain λ and p_{01} has already been studied in [12] and the details are omitted for conciseness. The Doppler offset f_d is updated using the gradient update in (29). The partial gradient $\frac{\partial \hat{u}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)}{\partial f_d}$ is given by

$$\frac{\partial \hat{u}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)}{\partial f_d} = \sum_{i=1}^{N_p}\sum_{n=1}^{N}\sum_{r=1}^{\widetilde{N}} 2Re\left(\tilde{a}'_{i,n}\left(\tilde{\boldsymbol{a}}_i^H\left(f_d,\theta_{R,r}\right)c_r + \bar{\boldsymbol{c}}_{i,r}^H\right)\mathbf{U}_i^H(:,n)\right),$$

where $\tilde{\boldsymbol{a}}_i\left(f_d,\theta\right) = \boldsymbol{a}_R(\theta)e^{j2\pi f_d i cos(\theta)}$, $c_r = -\frac{1}{\sigma^2}\left(|\mu_r|^2 + \Sigma_{r,r}\right)$, $\bar{\boldsymbol{c}}_{i,r} = \frac{1}{\sigma^2}\mu_r^*\boldsymbol{y}_{-i,r} - \frac{1}{\sigma^2}\sum_{j\neq r}\Sigma_{j,r}\tilde{\boldsymbol{a}}_i\left(f_d,\theta_{R,j}\right)$, $\boldsymbol{y}_{-i,r} = \boldsymbol{y}_i - \sum_{j\neq r}\left(\mu_r\tilde{\boldsymbol{a}}_i\left(f_d,\theta_{R,j}\right)\right)$, μ_r is the r-th element of $\boldsymbol{\mu}$, $\Sigma_{j,r}$ is the (j,r)-th element of $\boldsymbol{\Sigma}$, $\mathbf{U}_i^H(:,n)$ is the n-th column of the matrix $\mathbf{U}_i^H$, and $\tilde{a}'_{i,n} = \frac{\partial \tilde{a}_{i,n}(f_d,\theta)}{\partial f_d}$ is the derivative of the array response element $\tilde{a}_{i,n}\left(f_d,\theta\right)$ at f_d.

2) Simulation Results: We compare the normalized mean square error (NMSE) performance [12] for LASSO, conventional VBI [15], AMP, Turbo-AMP [11] and the proposed Turbo-VBI with Markov prior and i.i.d. prior, respectively. The channel parameters are generated according to the clustered delay line (CDL) channel models as specified in Table 7.7.1-2 (CDL-B) in 3GPP TR 38.901 Release 15 [28].

In Fig. 7, we consider a half-wavelength space ULA at the user with 128 antennas, and plot the NMSE versus SNR for $N_p = 6$ pilots and $N_b = 16$ RF chains. We also compare the performance with different sparsity levels λ. The Doppler parameter is set $f_d = 3667Hz$. It can be seen that the proposed Turbo-VBI with Markov prior achieves significant gain. In Fig. 8, we consider a 2D planar array at the user with 20×10 antennas, where both the horizontal and vertical inter-antenna spacings are a half wavelength. It can be seen that the proposed Turbo-VBI with Markov prior achieves significant gain. Although Turbo-AMP can exploit the Markov sparse structure of massive MIMO channel, it does not perform well and even diverge at high SNR due to the unstable performance under the correlated sensing matrix in this example. In Fig. 9, we show the convergence of the proposed algorithm with i.i.d prior for the ULA case. The proposed algorithm can converge within about 20 iterations.

B. 5G-based Localization

1) Implementation Details : In this example, for the LOS channel, the support vector $\boldsymbol{s}$ has independent prior as shown in (21). Therefore, in the support subgraph $\mathcal{G}_s$, each variable s_i simply connects to a factor node $h_i\left(s_i\right) = \alpha_i^{s_i}\left(1-\alpha_i\right)^{1-s_i}$ that incorporates the statistical PSI. Note that the factor nodes $h_i, \forall i$ reveal detailed structure of the factor node h associated with the independent support prior $p\left(\boldsymbol{s}\right)$. In this

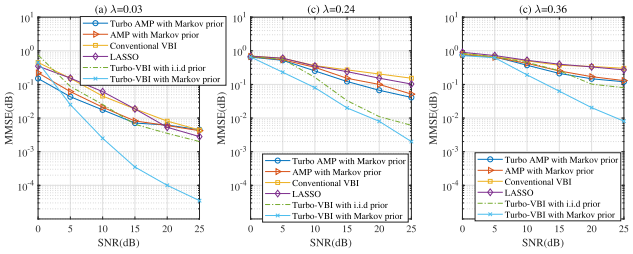


Fig. 7. NMSE of the channel estimate versus the SNR for ULA.

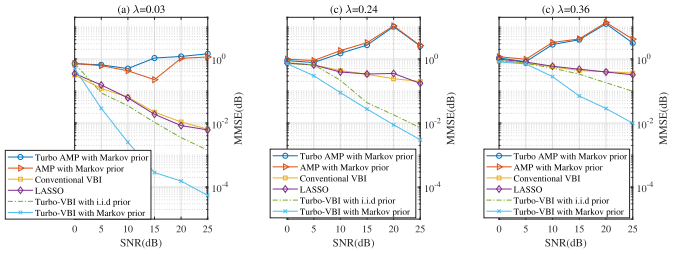


Fig. 8. NMSE of the channel estimate versus the SNR for 2D planar array.

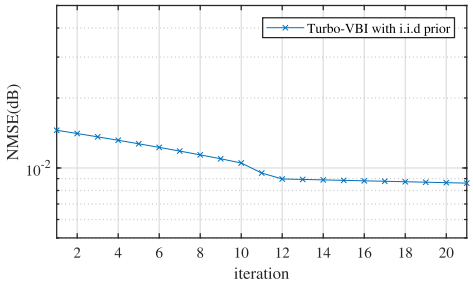


Fig. 9. The convergence of the proposed algorithm for ULA with $\lambda = 0.06$ and SNR $= 15dB$.

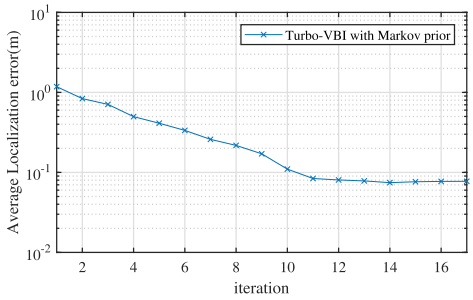


Fig. 10. The convergence speed of the proposed algorithm with $P_T = 15dB$.

case, the messages from Module B to variable nodes $\boldsymbol{s}$ are fixed as $\left\{\nu_{h\to s_i} = \nu_{h_i\to s_i} = \alpha_i^{s_i}\left(1-\alpha_i\right)^{1-s_i}\right\}$ in the E step.

For the NLOS channel, the support vector $\boldsymbol{v}_l$ has Markov prior as shown in (22). Therefore, in the support subgraph $\mathcal{G}_s$, for each l, the variables $v_{l,n}$'s and the associated factor nodes forms a Markov chain similar to that in Fig. 6. For given messages $\left\{\nu_{\eta_{l,n}\to v_{l,n}}\left(v_{l,n}\right), \forall n\right\}$ from Module A,[2] we can use the forward-backward message passing algorithm to update the messages associated with each $\boldsymbol{v}_l$ in Module B

[2]Here, $\eta_{l,n}$ denotes the factor node associated with the conditional probability $p\left(\rho_{l,n}|v_{l,n}\right)$ for the precision $\rho_{l,n}$ of $v_{l,n}$.

and calculate the messages $\left\{\nu_{h\to v_{l,n}}(\cdot)\right\}$ from Module B to Module A, similar to (45) - (48).

In the M step, we have $B=3$ and $\boldsymbol{\xi}_1=\boldsymbol{\kappa}$, $\boldsymbol{\xi}_2=\triangle\boldsymbol{r}$, $\boldsymbol{\xi}_3=\triangle\boldsymbol{\vartheta}$. The initial value of $\boldsymbol{\xi}_1=\boldsymbol{\kappa}$ is set to be the estimated noise precision. Considering the true position and AoA are around their nearest grid points, the initial values of $\boldsymbol{\xi}_2=\triangle\boldsymbol{r},\boldsymbol{\xi}_3=\triangle\boldsymbol{\vartheta}$ are set as zero vectors (since we usually have no prior knowledge about the offset vectors). Let $q\left(\boldsymbol{x}_l;\dot{\boldsymbol{\xi}}\right)=\mathcal{CN}\left(\boldsymbol{x}_l;\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l\right)$ denote the approximate posterior of $\boldsymbol{x}_l$ obtained using the sparse VBI estimator with $\dot{\boldsymbol{\xi}}$. The maximization of the surrogate function $\hat{u}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ in (31) over the noise precision $\boldsymbol{\kappa}$ has a closed-form solution given by

$$\kappa_l=\frac{M+a_\kappa}{b_\kappa+\mathrm{tr}(\boldsymbol{A}_l\left(\triangle\dot{\boldsymbol{r}}\right)\boldsymbol{\Sigma}_l\boldsymbol{A}_l\left(\triangle\dot{\boldsymbol{r}}\right)^H)+\left\|\boldsymbol{y}_l-\boldsymbol{A}_l\left(\triangle\dot{\boldsymbol{r}}\right)\boldsymbol{\mu}_l\right\|^2}.$$

On the other hand, $\triangle\boldsymbol{r}$ and $\triangle\boldsymbol{\vartheta}$ are updated using the gradient update in (29). Let $\triangle\boldsymbol{r}_i=[\triangle r_i^x,\triangle r_i^y]$ and $\boldsymbol{r}_i=[r_i^x,r_i^y]$, where $\triangle r_i^x,\triangle r_i^y$ (r_i^x,r_i^y) are the x- and y-coordinates of $\triangle\boldsymbol{r}_i$ $(\boldsymbol{r}_i)$ respectively. Then the partial gradient $\frac{\partial\hat{u}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)}{\partial\triangle r_i^x}$ is given by

$$\frac{\partial\hat{u}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)}{\partial\triangle r_i^x}=\sum_{l=1}^{L}2Re\left(\boldsymbol{a}_{l,i}'^H\left(\boldsymbol{a}_l\left(\theta_l\left(\triangle\boldsymbol{r}_i\right)\right)c_{l,i}+\bar{\boldsymbol{c}}_{l,i}\right)c_{l,i}^x\right),$$

where $c_{l,i}=-\frac{1}{\sigma^2}\left(\left|\mu_{l,i}\right|^2+\Sigma_{l,i,i}\right)$, $\bar{\boldsymbol{c}}_{l,i}=\frac{1}{\sigma^2}\left(\mu_{l,i}^*\boldsymbol{y}_{-l,i}-\sum_{j\neq i}\Sigma_{l,j,i}\boldsymbol{a}_l\left(\theta_l\left(\triangle\boldsymbol{r}_j\right)\right)\right)$, $\boldsymbol{y}_{-l,i}=\boldsymbol{y}_l-\sum_{j\neq i}\left(\mu_{l,j}\boldsymbol{a}_l\left(\theta_l\left(\triangle\boldsymbol{r}_j\right)\right)\right)$, $\mu_{l,i}$ is the i-th element of $\boldsymbol{\mu}_l$, $\Sigma_{l,j,i}$ is the (j,i)-th element of $\boldsymbol{\Sigma}_l$, $c_{l,i}^x=-\left(r_i^y+\triangle r_i^y-\tilde{p}_l^y\right)/\left\|\boldsymbol{r}_i+\triangle\boldsymbol{r}_i-\tilde{\boldsymbol{p}}_l\right\|^2$, and $\boldsymbol{a}_{l,i}'=\frac{\partial\boldsymbol{a}_l(\theta)}{\partial\theta}|_{\theta=\theta_l(\triangle\boldsymbol{r}_i)}$ is the derivative of the array response vector $\boldsymbol{a}_l\left(\theta\right)$ at $\theta=\theta_l\left(\triangle\boldsymbol{r}_i\right)$, where $\theta_l\left(\triangle\boldsymbol{r}_i\right)$ is an abbreviation for $\theta_l\left(\boldsymbol{r}_i+\triangle\boldsymbol{r}_i\right)$ defined in (17). The expression for the partial gradient $\frac{\partial\hat{u}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)}{\partial\triangle r_i^y}$ is similar but with $c_{l,i}^x$ replaced by $c_{l,i}^y=\left(r_i^x+\triangle r_i^x-\tilde{p}_l^x\right)/\left\|\boldsymbol{r}_i+\triangle\boldsymbol{r}_i-\tilde{\boldsymbol{p}}_l\right\|^2$. Finally, $\frac{\partial\hat{u}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)}{\partial\triangle\vartheta_l^m}$ is given by

$$\frac{\partial\hat{u}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)}{\partial\triangle\vartheta_l^m}=2Re\left(\boldsymbol{a}_l'\left(\vartheta_l^m\right)^H\left(\boldsymbol{a}_l\left(\vartheta_l^m\right)c_{l,m}+\bar{\boldsymbol{c}}_{l,m}\right)\right),$$

where $\vartheta_l^m=\vartheta_m+\triangle\vartheta_l^m$, $c_{l,m}=-\frac{1}{\sigma^2}\left(\left|\mu_{l,m}^y\right|^2+\Sigma_{l,m,m}\right)$, $\bar{\boldsymbol{c}}_{l,m}=\frac{1}{\sigma^2}\mu_{l,i}^*\boldsymbol{y}_{-l,i}-\frac{1}{\sigma^2}\sum_{m'\neq m}\Sigma_{l,m',m}\boldsymbol{a}_l\left(\vartheta_l^{m'}\right)\boldsymbol{y}_{-l,i}=\boldsymbol{y}_l-\sum_{j\neq i}\left(\mu_{l,j}\boldsymbol{a}_l\left(\theta_l\left(\triangle\boldsymbol{r}_j\right)\right)\right)$, $\mu_{l,i}$ is the i-th element of $\boldsymbol{\mu}_l$, $\Sigma_{l,j,i}$ is the (j,i)-th element of $\boldsymbol{\Sigma}_l$, $c_{l,i}^x=-\left(r_i^y+\triangle r_i^y-\tilde{p}_l^y\right)/\left\|\boldsymbol{r}_i+c-\tilde{\boldsymbol{p}}_l\right\|^2$.

2) Simulation Results: As in [23]. The source is positioned randomly within an area of size 90 × 90 m. Four BSs are positioned at the corners. Every BS is equipped with a ULA of 32 antennas, with the inter antenna spacing equals to half wavelength. The NLOS component is generated according to the clustered delay line (CDL) channel models as specified in Table 7.7.1-2 (CDL-B) in 3GPP TR 38.901 Release 15 [28]. We compare the localization errors for DiSouL [23], conventional VBI [15], the proposed Turbo-VBI with statistical PSI and without statistical PSI (i.e., $\alpha_i=\frac{1}{Q}$). The statistical PSI is generated from a Gaussian localization error with standard derivation $\sigma_p=2$ meters.

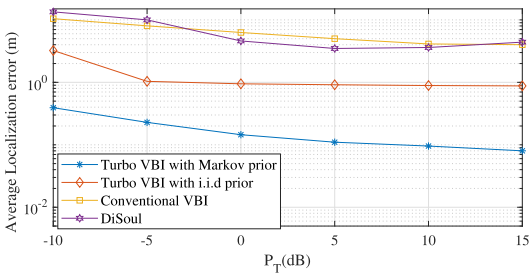


Fig. 11. Average localization error versus the transmit power.

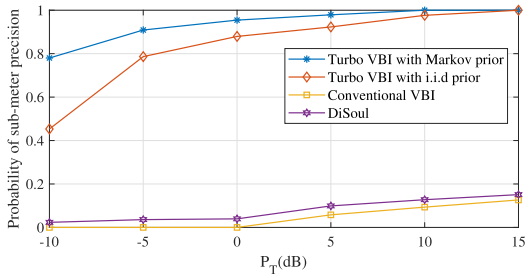


Fig. 12. Probability of sub-meter precision versus the transmit power.

In Fig. 10, we show the convergence of the proposed algorithm with Markov prior for a typical case with $P_T=15dB$. The proposed algorithm can converge within about 15 iterations. In Fig. 11, we plot the average localization error versus the transmit power. It can be seen that the proposed Turbo-VBI without statistical PSI can already achieve a smaller localization error compared to the baselines. With the statistical PSI, the proposed Turbo-VBI can achieve an even larger gain over the baselines. DiSouL [23] becomes worse for higher transmit powers because as P_T increases, the effective noise power (i.e., the mismatch between the measurements and the signal) caused by the offset is enlarged, but the constraint bound ϵ in [23, (17b)] only depends on the AWGN power. In Fig. 12, we plot the probability of sub-meter precision versus the transmit power. Again, the proposed Turbo-VBI achieves the highest probability of sub-meter precision.

VI. Conclusion

We propose a novel Turbo-VBI framework for robust recovery of structured sparse signals under uncertain (and possibly correlated) sensing matrices. To capture various structured sparsities, we propose a new 3LHS sparse prior model which is not only more flexible than existing commonly used sparse models, but tractable for low complexity algorithm design. To handle the 3LHS sparse prior model and uncertain/correlated sensing matrix, we propose a Turbo-VBI algorithm which approximately calculates the marginal posteriors of the sparse signal by combining the message passing and VBI approaches via the turbo framework, and use an in-exact block MM method to find an approximate MAP estimator for the uncertain parameters in the sensing matrix. We further establish the convergence of the in-exact

block MM and sparse VBI components in the Turbo-VBI framework. Finally, we apply the proposed Turbo-VBI to solve two application problems in high-mobility massive MIMO channel estimation and 5G-based Localization. The proposed Turbo-VBI algorithm is shown in the simulations to achieve significant gains over baseline algorithms.

APPENDIX

A. Proof of Theorem 1

Using the property of surrogate function and gradient update, we have $\ln p(\boldsymbol{y}, \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}) = u\left(\boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right) \leq u\left(\boldsymbol{\xi}_j^{(i+1)}, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right) \leq \ln p(\boldsymbol{y}, \boldsymbol{\xi}_j^{(i+1)}, \boldsymbol{\xi}_{-j}^{(i)}), \forall j \in \overline{\mathcal{J}}_c$, where the equality holds only when the gradient w.r.t. $\boldsymbol{\xi}_j$ is zero. It is clear that the update in (28) also strictly increases the surrogate function and the original objective function whenever $\frac{\partial u\left(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right)}{\partial \boldsymbol{\xi}_j} \neq 0$. Therefore, the objective value will keep increasing until converging to a certain value p^*, and we must have

$$\lim_{i\to\infty} \frac{\partial u\left(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right)}{\partial \boldsymbol{\xi}_j} = 0, \quad \forall j. \tag{49}$$

(otherwise, the objective value will keep increasing to infinity, which contradicts with the fact that $\ln p(\boldsymbol{y}, \boldsymbol{\xi}_j^{(i+1)}, \boldsymbol{\xi}_{-j}^{(i)})$ must be bounded above). Then according to (49) and the property of gradient update, we must have $\lim_{i\to\infty} \left\| \boldsymbol{\xi}_j^{(i+1)} - \boldsymbol{\xi}_j^{(i)} \right\| = 0, \forall j \in \overline{\mathcal{J}}_c$. Moreover, it follows from (49) and the strong convexity of $u\left(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right)$ w.r.t. $\boldsymbol{\xi}_j, \forall j \in \mathcal{J}_c$ that $\lim_{i\to\infty} \left\| \boldsymbol{\xi}_j^{(i+1)} - \boldsymbol{\xi}_j^{(i)} \right\| = 0, \forall j \in \mathcal{J}_c$. Therefore, we have

$$\lim_{i\to\infty} \left\| \boldsymbol{\xi}_j^{(i+1)} - \boldsymbol{\xi}_j^{(i)} \right\| = 0, \quad \forall j. \tag{50}$$

It follows from (50) that all the B sequences $\left\{\boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right\}, j = 0, 1, \ldots, B-1$ have the same set of limiting points. Let $\left\{\boldsymbol{\xi}_j^{(i_t)}, \boldsymbol{\xi}_{-j}^{(i_t)}, t = 1, 2, \ldots\right\}$ denote a subsequence that converges to a limiting point $\boldsymbol{\xi}^*$. Suppose $\boldsymbol{\xi}^*$ is not a stationary point of $\ln p(\boldsymbol{y}, \boldsymbol{\xi})$, then $\frac{\partial \ln p(\boldsymbol{y}, \boldsymbol{\xi}^*)}{\partial \boldsymbol{\xi}} \neq 0$ and it follows from (50) that $\lim_{t\to\infty} \frac{\partial u\left(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{-j}^{(i_t)}; \boldsymbol{\xi}_j^{(i_t)}, \boldsymbol{\xi}_{-j}^{(i_t)}\right)}{\partial \boldsymbol{\xi}_j} \neq 0$ must hold at least for some j, which contradicts with (49). Therefore, every limiting point $\boldsymbol{\xi}^*$ must be a stationary point of $\ln p(\boldsymbol{y}, \boldsymbol{\xi})$.

B. Derivation of (37)-(44)

Based on (36), $q(\boldsymbol{x})$ in (37) can be obtained as

$$\begin{aligned}
\ln q(\boldsymbol{x}) &\propto \langle \ln p(\boldsymbol{x}|\boldsymbol{\rho}) \rangle_{\boldsymbol{\rho}} + \ln p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\xi}) \\
&\propto -\boldsymbol{x}^H \operatorname{diag}(\langle \widetilde{\boldsymbol{\rho}} \rangle) \boldsymbol{x} - \left\| \operatorname{diag}^{1/2}(\boldsymbol{\kappa}) (\boldsymbol{y} - \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}) \right\|^2 \\
&\propto -(\boldsymbol{x} - \boldsymbol{\mu})^H \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}).
\end{aligned}$$

$q(\boldsymbol{\rho})$ in (40) can be obtained as

$$\begin{aligned}
\ln q(\boldsymbol{\rho}) &\propto \langle \ln p(\boldsymbol{x}|\boldsymbol{\rho}) \rangle_{\boldsymbol{x}} + \langle \ln p(\boldsymbol{\rho}|\boldsymbol{s}) \rangle_{\boldsymbol{s}} \\
&\propto \sum_{i=1}^{Q} \sum_{n \in \mathcal{I}_i} (\langle s_i \rangle a_n + \langle 1 - s_i \rangle \overline{a}_n) \ln \rho_n \\
&\quad - \left(\left\langle |x_n|^2 \right\rangle + \langle s_i \rangle b_n + \langle 1 - s_i \rangle \overline{b}_n \right) \rho_n.
\end{aligned}$$

$q(\boldsymbol{s})$ in (43) can be obtained as

$$\begin{aligned}
&\ln q(\boldsymbol{s}) \\
&\propto \langle \ln p(\boldsymbol{s}|\boldsymbol{\rho}) \rangle_{\boldsymbol{\rho}} + \ln \hat{p}(\boldsymbol{s}) \\
&\propto \sum_{i=1}^{Q} \sum_{n \in \mathcal{I}_i} s_i (\ln b_n^{a_n} + (a_n - 1) \langle \ln \rho_n \rangle - b_n \langle \rho_n \rangle - \ln \Gamma(a_n)) \\
&\quad + (1 - s_i) \left(\ln \overline{b}_n^{\overline{a}_n} + (\overline{a}_n - 1) \langle \ln \rho_n \rangle - \overline{b}_n \langle \rho_n \rangle - \ln \Gamma(\overline{a}_n) \right) \\
&\quad + \sum_{i=1}^{Q} (s_i \ln \pi_i + (1 - s_i) \ln (1 - \pi_i)) \\
&\propto \sum_{i=1}^{Q} \sum_{n \in \mathcal{I}_i} s_i \ln \frac{\pi_i b_n^{a_n}}{\Gamma(a_n)} e^{(a_n - 1)\langle \ln \rho_n \rangle - b_n \langle \rho_n \rangle} \\
&\quad + (1 - s_i) \ln \frac{(1 - \pi_i) \overline{b}_n^{\overline{a}_n}}{\Gamma(\overline{a}_n)} e^{(\overline{a}_n - 1)\langle \ln \rho_n \rangle - \overline{b}_n \langle \rho_n \rangle} \\
&\propto \ln \prod_{i=1}^{Q} (\tilde{\pi}_i)^{s_i} (1 - \tilde{\pi}_i)^{1 - s_i}.
\end{aligned}$$

REFERENCES

[1] L. He and L. Carin, "Exploiting structure in wavelet-based Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 57, no. 9, pp. 3488–3497, Sep. 2009.

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

[3] A. Liu, V. K. N. Lau, and W. Dai, "Exploiting burst-sparsity in massive MIMO with partial channel support information," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7820–7830, Nov. 2016.

[4] J. A. del Peral-Rosado, J. A. Lopez-Salcedo, S. Kim, and G. Seco-Granados, "Feasibility study of 5G-based localization for assisted driving," in *Proc. Int. Conf. Localization GNSS (ICL-GNSS)*, Jun. 2016. pp. 1–6.

[5] C. Thrampoulidis, A. Panahi, and B. Hassibi, "Asymptotically exact error analysis for the generalized equation-LASSO," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 2021–2025.

[6] M. Stojnic, F. Parvaresh, and B. Hassibi, "On the reconstruction of block-sparse signals with an optimal number of measurements," *IEEE Trans. Signal Process.*, vol. 57, no. 8, pp. 3075–3085, Aug. 2009.

[7] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.

[8] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 45, pp. 18914–18919, Nov. 2009.

[9] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2011, pp. 2168–2172.

[10] J. Ma, X. Yuan, and L. Ping, "On the performance of turbo signal recovery with partial DFT sensing matrices," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1580–1584, Oct. 2015.

[11] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a Markov-tree prior," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3439–3448, Jul. 2012.

[12] L. Chen, A. Liu, and X. Yuan, "Structured turbo compressed sensing for massive MIMO channel estimation using a Markov prior," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4635–4639, May 2018.

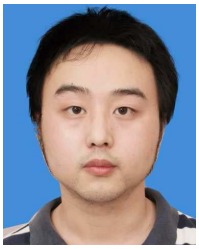
[13] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.

[14] 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, Jan. 2015.

[15] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 131–146, Nov. 2008.

[16] C. Liu, "Maximum likelihood estimation from incomplete data via em-type algorithms," in *Proc. Adv. Med. Statist.*, Nov. 2003, pp. 1051–1071.

[17] M. Crouse, R. Nowak, and R. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 886–902, Apr. 1998.
[18] P. Schniter, "Turbo reconstruction of structured sparse signals," in *Proc. 2010 44th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2010, pp. 1–6.
[19] L. Lian, A. Liu, and V. K. N. Lau, "Weighted LASSO for sparse recovery with statistical prior support information," *IEEE Trans. Signal Process.*, vol. 66, no. 6, pp. 1607–1618, Mar. 2018.
[20] D. L. Donoho, A. Maleki, and A. Montanari, "The noise-sensitivity phase transition in compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6920–6941, Oct. 2011.
[21] W. Guo, W. Zhang, P. Mu, F. Gao, and H. Lin, "High-mobility wideband massive MIMO communications: Doppler compensation, analysis and scaling laws," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3177–3191, Jun. 2019.
[22] D. Tse and P. Viswanath, *Fundamentals Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[23] N. Garcia, H. Wymeersch, E. G. Larsson, A. M. Haimovich, and M. Coulon, "Direct localization for massive MIMO," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2475–2487, May 2017.
[24] J. Dai, A. Liu, and V. K. N. Lau, "FDD massive MIMO channel estimation with arbitrary 2D-array geometry," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2584–2599, May 2018.
[25] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1995.
[26] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
[27] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss–Seidel method under convex constraints," *Oper. Res. Lett.*, vol. 26, no. 3, pp. 127–136, Apr. 2000.
[28] *Study on Channel Model for Frequencies From 0.5 to 100 GHz*, document TR 38.901 V15.0.0, 3GPP, Jun. 2018.



An Liu (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Peking University, China, in 2004 and 2011, respectively. From 2008 to 2010, he was a Visiting Scholar with the Department of ECEE, University of Colorado at Boulder. He has been a Post-Doctoral Research Fellow from 2011 to 2013, a Visiting Assistant Professor in 2014, and a Research Assistant Professor from 2015 to 2017, with the Department of ECE, HKUST. He is currently a Distinguished Research Fellow with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests include wireless communications, stochastic optimization, and compressive sensing.

Guanying Liu received the B.Eng. degree in information science and technology from Southwest Jiaotong University in 2017. She is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University. Her research interests include signal processing and compressive sensing.

Lixiang Lian (Student Member, IEEE) received the B.Eng. degree in information and communication engineering from Zhejiang University in 2014 and the Ph.D. degree from the Department of ECE, The Hong Kong University of Science and Technology (HKUST), in 2020. Her research interests include wireless communication and compressive sensing.

Vincent K. N. Lau (Fellow, IEEE) received the B.Eng. degree (Hons.) from The University of Hong Kong in 1992 and the Ph.D. from Cambridge University in 1997. He was with Bell Labs from 1997 to 2004 and the Department of ECE, The Hong Kong University of Science and Technology (HKUST), in 2004. He is currently a Chair Professor and the Founding Director of the Huawei-HKUST Joint Innovation Lab, HKUST. His current research interests include robust and delay-optimal cross layer optimization for MIMO/OFDM wireless systems, interference mitigation techniques for wireless networks, massive MIMO, M2M, and network control systems.

Min-Jian Zhao (Member, IEEE) received the M.Sc. and Ph.D. degrees in communication and information systems from Zhejiang University, Hangzhou, China, in 2000 and 2003, respectively. He was a Visiting Scholar with the University of York (UK) in 2010. He is currently a Professor and the Deputy Director of the College of Information Science and Electronic Engineering, Zhejiang University. His research interests include modulation theory, channel estimation and equalization, MIMO, signal processing for wireless communications, anti-jamming technology for wireless transmission and networking, and communication SOC chip design.