Structured Turbo Compressed Sensing for Massive MIMO Channel Estimation Using a Markov Prior

Lei Chen, An Liu, *Senior Member, IEEE*, and Xiaojun Yuan, *Senior Member, IEEE*

***Abstract*—Accurate channel estimation with small pilot overhead is vital to improve the capacity and reliability of massive MIMO systems. Recently, compressed sensing has been applied to reduce the pilot overhead in such systems by exploiting the underlying structured channel sparsity. In this paper, we propose a *structured turbo compressed sensing (Turbo-CS)* framework for the design and analysis of structured sparse channel estimation algorithms. In this framework, a Markov prior is used to model the structured sparsity in massive MIMO channels. Then we extend the Turbo-CS algorithm for independent and identically distributed priors to propose a structured Turbo-CS algorithm to solve the resulting sparse channel estimation problem with the Markov chain prior. We also accurately characterize the performance of the algorithm using state evolution. As compared to the existing algorithms, both the state evolution analysis and simulations show that the structured Turbo-CS algorithm can substantially enhance the channel estimation performance.**

***Index Terms*—Massive MIMO, compressed sensing, channel estimation, structured sparsity, message passing.**

I. Introduction

Massive MIMO is a promising technology in 5G wireless communications due to its potential high spectrum and power efficiency. To improve the capacity and reliability of the system, knowledge of channel state information (CSI) is essential. If we use conventional channel estimation methods, such as least square, to estimate the CSI in a frequency-division duplex downlink system, the number of pilot sequences should be at least the same as the number of antennas N at the base station (BS), which is excessively large in massive MIMO systems.

In practice, due to limited scatterers in the environment, massive MIMO channels exhibit structured sparsity in the virtual angular domain [1]–[3]. Motivated by this observation, many compressed sensing algorithms have been applied to estimate the angular domain massive MIMO channel, such as orthogonal matching pursuit (OMP) [4] and compressive sampling matching pursuit (CoSaMP) [4]. Recently, several Structured Sparse Channel Estimation (SSCE) algorithms have been developed to further reduce the pilot sequences by exploiting structured sparsity of massive MIMO channels. For example, in [1], a distributed sparsity adaptive matching pursuit (DSAMP) algorithm is proposed to jointly estimate multiple channels for different subcarriers with consideration of spatially common sparsity; in [5], a spatial-temporal common sparsity of MIMO channels is considered; in [6], the joint sparsity of channel vectors across different antennas is exploited to design more efficient channel estimation algorithms. In recent work [2], it is shown that due to the physical scattering structure, the significant elements in the angular domain massive MIMO channel will appear in bursts, with each burst corresponding to a scattering cluster in the propagation environment. Such a burst-sparse structure is exploited in [2] to design a burst LASSO algorithm which can significantly reduce the required pilots for successful channel recovery.

The existing SSCE algorithms, however, have two drawbacks. First, the SSCE algorithms that exploit joint sparsity or burst sparsity rely on some restrictive assumptions. For example, the burst LASSO algorithm in [2] only works well when all bursts of the channel vector have similar sizes. Second, most existing works focus on the SSCE algorithm design under a specific assumption about the sparse structure and do not provide precise performance analysis. Therefore, there is no systematic framework for the design and analysis of SSCE algorithms. In practice, it is also important to accurately characterize the performance of the SSCE algorithm in order to determine the minimum pilots required to achieve the target channel estimation quality.

In this paper, we develop a structured Turbo-CS framework for the design and analysis of SSCE algorithms. Note that a Turbo-CS algorithm for independent and identically distributed (i.i.d.) priors has been proposed in [7]. However, the algorithm cannot exploit the structured sparsity of massive MIMO channels.

Specifically, we first propose a spatial Markov channel model to capture the structured sparsity for massive MIMO channels without requiring any prior knowledge of channel coefficients, and design a Partial DFT Random Permutation (PDFT-RP) pilot matrix for downlink training, which can achieve better performance than the i.i.d. Gaussian pilot matrix considered in [2], [6]. Then we propose a structured Turbo-CS algorithm to solve the resulting channel estimation problem with a Markov chain prior. Although we focus on the Markov chain prior in this paper, structured Turbo-CS approach can adapt to more general priors, including Markov random field and Markov tree, and thus provides a generic framework for the design and analysis of SSCE algorithms. We show that the performance of the algorithm can be well predicted by the state evolution [8]. Finally, both simulations and analysis show that the proposed structured Turbo-CS algorithm achieves considerable gain over various state-of-the-art baseline algorithms under a realistic MIMO channel model.

II. System Model

A. Downlink Training With PDFT-RP Pilot Matrix

Consider a massive MIMO system with one BS serving single-antenna users, where the BS is equipped with a *half-wavelength space* uniform linear array (ULA) comprised of N antennas. To estimate the

Manuscript received June 17, 2017; revised September 8, 2017 and October 12, 2017; accepted October 14, 2017. Date of publication December 27, 2017; date of current version May 14, 2018. The work of A. Liu was supported by the National Science Foundation of China under Project No. 61571383. The work of X. Yuan was supported by the China Recruitment Program of Global Young Experts. The review of this paper was coordinated by Dr. E. K. S. Au. *(Corresponding author: Xiaojun Yuan.)*

L. Chen is with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the School of Information Science and Technology, ShanghaiTech University, Shanghai 200031, China (e-mail: chenlei@shanghaitech.edu.cn).

A. Liu is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: eewendaol@ust.hk).

X. Yuan was with the School of Information Science and Technology, ShanghaiTech University, Shanghai 200031, China. He is now with the National Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 610051, China (e-mail: xjyuan@uestc.edu.cn).

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

Digital Object Identifier 10.1109/TVT.2017.2787708

downlink channel $\tilde{\boldsymbol{h}} \in \mathbb{C}^{N\times 1}$, the BS transmits M training sequences $\boldsymbol{x}_t^H \in \mathbb{C}^{1\times N}, t = 1, ..., M$. Then the received signal $\boldsymbol{y} \in \mathbb{C}^{M\times 1}$ of a user can be written as

$$\boldsymbol{y} = \boldsymbol{X}\tilde{\boldsymbol{h}} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M]^H$ is an $M \times N$ pilot matrix which is known to the receiver and $\boldsymbol{n} \sim \mathcal{CN}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ is the Gaussian noise.

In many existing works, such as [2] and [6], the pilot matrix $\boldsymbol{X}$ is chosen to have i.i.d. Gaussian entries. However, it is shown in [7] that a partial orthogonal sensing matrix achieves better performance than an i.i.d. Gaussian sensing matrix under the Turbo-CS algorithm. Motivated by this observation, the pilot matrix $\boldsymbol{X}$ is chosen to be a PDFT-RP matrix as $\boldsymbol{X} = \boldsymbol{SFPF}$, where $\boldsymbol{S}$ is a selection matrix consisting of randomly selected and reordered rows of the $N \times N$ identity matrix, $\boldsymbol{F}$ is the DFT matrix, and $\boldsymbol{P}$ is a random permutation matrix generated by a randomly reordered $N \times N$ identity matrix. Such a pilot matrix design ensures that the resulting sensing matrix $\boldsymbol{A} = \boldsymbol{SFP}$ is a partial orthogonal matrix and the transmit power of each antenna is similar when transmitting each training sequence. The superior performance of such a design is verified in the simulations. Define the angular domain channel as $\boldsymbol{h} = \boldsymbol{F}\tilde{\boldsymbol{h}}$. Then the received signal can be rewritten as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{F}^H\boldsymbol{h} + \boldsymbol{n} = \boldsymbol{A}\boldsymbol{h} + \boldsymbol{n}. \tag{2}$$

Note that the angular domain channel $\boldsymbol{h}$ is usually sparse due to limited scatterers at the BS [1], [2], [6]. Therefore, (2) is a standard compressed sensing model, with sensing matrix $\boldsymbol{A}$ and sparse channel $\boldsymbol{h}$.

B. Spatial Markov Channel Model

Consider the widely used Bernoulli–Gaussian (BG) model for the angular domain channel $\boldsymbol{h}$ [6]. Each channel coefficient h_n has a conditionally independent distribution expressed as

$$p(h_n|s_n) = \delta(s_n - 1)\mathcal{CN}(h_n; 0, \sigma_h^2) + \delta(s_n)\delta(h_n), \tag{3}$$

where $s_n \in \{0, 1\}$ is a hidden binary state, $\delta(\cdot)$ denotes the unit impulse function, and $\mathcal{CN}(h_n; \bar{h}, \sigma_h^2)$ is the complex Gaussian distribution with mean $\bar{h}$ and variance σ_h^2. For convenience, $\boldsymbol{s} = [s_1, ..., s_N]^T$ is called the *channel support vector* since it indicates the support of the sparse channel vector $\boldsymbol{h}$.

In practice, only the signals reflected 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. Such a cluster structure can be modeled using a Markov chain as follow [9]

$$p(\boldsymbol{s}) = p(s_1)\prod_{n=2}^{N} p(s_n|s_{n-1}), \tag{4}$$

with the transition probability given by

$$p(s_n|s_{n-1}) = \begin{cases} (1 - p_{01})^{1-s_n} p_{01}^{s_n}, & s_{n-1} = 0; \\ p_{10}^{1-s_n}(1 - p_{10})^{s_n}, & s_{n-1} = 1. \end{cases} \tag{5}$$

Note that the Markov parameters p_{01} and p_{10} determine the average cluster size and the average gap between two clusters in $\boldsymbol{h}$. Specifically, a smaller p_{10} leads to a larger average cluster size, and a smaller p_{01} leads to a larger average gap between two clusters. The initial distribution $p(s_1)$ is set to be the steady state distribution, i.e., $p(s_1 = 1) = \frac{p_{01}}{p_{01}+p_{10}} \triangleq \lambda$. This ensures that the marginal distribution of s_n is $p(s_n) = \lambda, \forall n$, where λ reflects the sparsity of $\boldsymbol{h}$.

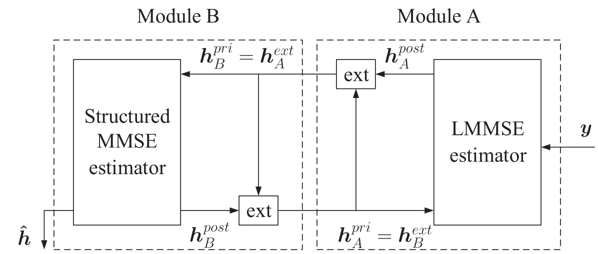

Fig. 1. Modules of the structured Turbo-CS algorithm and message flows between different modules.

III. Structured Turbo-CS Algorithm

A. Outline of the Algorithm

One of the most popular estimators for the linear observation model in (2) is the maximum a *posterior* estimate:

$$\begin{aligned} \hat{\boldsymbol{h}} &= \arg\max_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{y}), \\ &\propto \arg\max_{\boldsymbol{h}} p(\boldsymbol{y}|\boldsymbol{h})p(\boldsymbol{h}), \end{aligned} \tag{6}$$

where

$$p(\boldsymbol{y}|\boldsymbol{h}) = \frac{1}{(\pi\sigma^2)^M} \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}\|_2^2}{\sigma^2}\right), \tag{7}$$

and $p(\boldsymbol{h})$ is the prior distribution depending on the channel structure. Since the complexity of calculating the exact solution of (6) is high, the original Turbo-CS algorithm in [7] aims at calculating an approximate solution of (6). Specifically, Turbo-CS contains two modules: a linear minimum mean square error (LMMSE) estimator (Module A) and a minimum mean square error (MMSE) estimator (Module B), where Module A and Module B are used to handle the likelihood $p(\boldsymbol{y}|\boldsymbol{h})$ and the prior $p(\boldsymbol{h})$, respectively. The extrinsic calculation in the Turbo-CS algorithm is inspired by the extrinsic information in iterative decoding [10]. This process decorrelates the noise between input and output signal for each module [11], which makes the algorithm achieve a better convergence performance. The two modules are executed iteratively until convergence.

The main difference between Turbo-CS and structured Turbo-CS, as illustrated in Fig. 1, is that the assumption on the prior distribution of $\boldsymbol{h}$ is different. Specifically, Turbo-CS assumes an i.i.d. prior, and thus the MMSE estimator for $\boldsymbol{h}$ is given by the *posterior* mean under i.i.d prior. On the other hand, structured Turbo-CS assumes a Markov prior, and thus the structured MMSE estimator is given by the *posterior* mean under the Markov prior, which can be calculated using message passing over the corresponding Markov chain factor graph. The details of structured Turbo-CS are illustrated in Algorithm 1.

B. Details of Structured MMSE Estimator

In this section, we explain the details of structured MMSE estimator. To start with, a basic assumption is to model $\boldsymbol{h}_B^{pri}$ as an additive white Gaussian noise (AWGN) observation, i.e.,

$$\boldsymbol{h}_B^{pri} = \boldsymbol{h} + \boldsymbol{w}, \tag{8}$$

where $\boldsymbol{w} \sim \mathcal{CN}(\boldsymbol{0}, v_B^{pri}\boldsymbol{I})$ is independent of $\boldsymbol{h}$. Similar assumptions have been used in the development of turbo [7], [8] and approximate message passing (AMP) [12], [13] based sparse signal recovery algorithms. Under this assumption, the factor graph of the joint distribution $p(\boldsymbol{h}_B^{pri}, \boldsymbol{h}, \boldsymbol{s})$, denoted by $\mathcal{G}_B$, is a tree-type graph as shown in Fig. 2,

---

Algorithm 1: Structured Turbo-CS algorithm.

Input: received signal $\boldsymbol{y}$, pilot matrix $\boldsymbol{X}$, and number of iterations.
Output: channel state information $\hat{\boldsymbol{h}}$.

Initialize: $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{F}^H$, $\boldsymbol{h}_A^{pri} = \boldsymbol{0}$, $v_A^{pri} = \lambda\sigma_h^2$, $t = 0$.

Module A:

% LMMSE estimator

1: $\boldsymbol{h}_A^{post} = \boldsymbol{h}_A^{pri} + \frac{v_A^{pri}}{v_A^{pri}+\sigma^2}\boldsymbol{A}^H(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{h}_A^{pri})$

2: $v_A^{post} = v_A^{pri} - \frac{M}{N}\cdot\frac{(v_A^{pri})^2}{v_A^{pri}+\sigma^2}$

% Update extrinsic

3: $v_B^{pri} = v_A^{ext} = (\frac{1}{v_A^{post}} - \frac{1}{v_A^{pri}})^{-1}$

4: $\boldsymbol{h}_B^{pri} = \boldsymbol{h}_A^{ext} = v_B^{pri}(\frac{\boldsymbol{h}_A^{post}}{v_A^{post}} - \frac{\boldsymbol{h}_A^{pri}}{v_A^{pri}})$

Module B:

% Structured MMSE estimator

5: Update the messages $\nu_{h_n\to f_n}(h_n)$, $\nu_{f_n\to s_n}(s_n)$ using (9) and (10).

6: $\lambda_1^f = \lambda$, $\lambda_N^b = \frac{1}{2}$

for $n = 2, \ldots, N$

7: $\lambda_n^f = \frac{p_{01}(1-\pi_{n-1}^{out})(1-\lambda_{n-1}^f)+p_{11}\pi_{n-1}^{out}\lambda_{n-1}^f}{(1-\pi_{n-1}^{out})(1-\lambda_{n-1}^f)+\pi_{n-1}^{out}\lambda_{n-1}^f}$

end for

for $n = N-1, \ldots, 2, 1$

8: $\lambda_n^b = \frac{p_{10}(1-\pi_{n+1}^{out})(1-\lambda_{n+1}^b)+(1-p_{10})\pi_{n+1}^{out}\lambda_{n+1}^b}{(p_{00}+p_{10})(1-\pi_{n+1}^{out})(1-\lambda_{n+1}^b)+(p_{11}+p_{01})\pi_{n+1}^{out}\lambda_{n+1}^b}$

end for

9: Update the messages $\nu_{s_n\to f_n}(s_n)$, $\nu_{f_n\to h_n}(h_n)$ using (12) and (14).

10: $h_{B,n}^{post} = \mathrm{E}(h_n|\boldsymbol{h}_B^{pri}) = \int_{h_n} h_n p(h_n|\boldsymbol{h}_B^{pri})$

11: $v_{B,n}^{post} = \mathrm{Var}(h_n|\boldsymbol{h}_B^{pri})$
$= \int_{h_n} |h_n - \mathrm{E}(h_n|\boldsymbol{h}_B^{pri})|^2 p(h_n|\boldsymbol{h}_B^{pri})$ where the conditional distribution

$$p(h_n|\boldsymbol{h}_B^{pri}) = \nu_{f_n\to h_n}(h_n)\mathcal{CN}(h_n; h_{B,n}^{pri}, v_{B,n}^{pri})/C$$

with prior $\nu_{f_n\to h_n}(h_n)$ and normalization factor C.

12: $v_B^{post} = \frac{1}{N}\sum_{n=1}^N v_{B,n}^{post}$

% Update extrinsic

13: $v_A^{pri} = v_B^{ext} = (\frac{1}{v_B^{post}} - \frac{1}{v_B^{pri}})^{-1}$

14: $\boldsymbol{h}_A^{pri} = \boldsymbol{h}_B^{ext} = v_A^{pri}(\frac{\boldsymbol{h}_B^{post}}{v_B^{post}} - \frac{\boldsymbol{h}_B^{pri}}{v_B^{pri}})$

15: **Let** $t = t+1$. If $t = T$, terminate and output $\hat{\boldsymbol{h}} = \boldsymbol{h}_B^{post}$. Otherwise, return to Line 1.

---

where the factor nodes $\{f_n(h_n, s_n) = p(h_n|s_n)|\forall n\}$ are the conditional priors in (3), and $\{g_n(h_{B,n}^{pri}, h_n) = \mathcal{CN}(h_n; h_{B,n}^{pri}, v_B^{pri})|\forall n\}$, with $h_{B,n}^{pri}$ denoting the n-th element of $\boldsymbol{h}_B^{pri}$. Therefore, a structured MMSE estimator (Lines 5–12) can be interpreted as a message passing algorithm over $\mathcal{G}_B$ to calculate the *posterior* distributions $\{p(h_n|\boldsymbol{h}_B^{pri})|\forall n\}$.

Since $\mathcal{G}_B$ is a tree-type graph, the exact *posterior* distribution $\{p(h_n|\boldsymbol{h}_B^{pri})|\forall n\}$ can be calculated using the sum-product message passing from Lines 5 to 12, as elaborated below.

According to the sum-product rule, the message from variable node h_n to factor node f_n is

$$\nu_{h_n\to f_n}(h_n) = \mathcal{CN}(h_n; h_{B,n}^{pri}, v_B^{pri}), \tag{9}$$

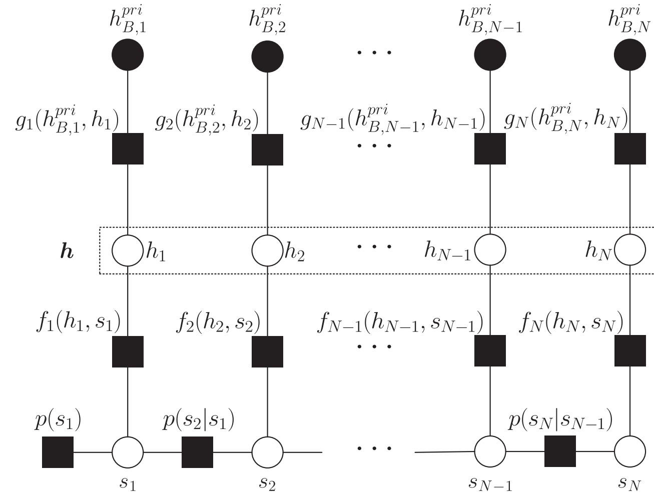

Fig. 2. Factor graph $\mathcal{G}_B$ of structured MMSE estimator in Fig. 1.

and the message from factor node f_n to variable node s_n is

$$\begin{aligned}\nu_{f_n\to s_n}(s_n) &= \frac{\int f_n(h_n, s_n)\nu_{h_n\to f_n}(h_n)dh_n}{\sum_{s_n'=0,1}\int f_n(h_n, s_n')\nu_{h_n\to f_n}(h_n)dh_n}\\ &= \pi_n^{out}\delta(s_n - 1) + (1-\pi_n^{out})\delta(s_n),\end{aligned} \tag{10}$$

where

$$\pi_n^{out} = \left(1 + \frac{\mathcal{CN}(0; h_{B,n}^{pri}, v_B^{pri})}{\int \mathcal{CN}(h_n; 0, \sigma_n^2)\mathcal{CN}(h_n; h_{B,n}^{pri}, v_B^{pri})dh_n}\right)^{-1}. \tag{11}$$

From Lines 6 to 8, a standard forward-backward message passing algorithm is performed over the Markov chain $\boldsymbol{s}$. After that, according to the sum-product rule, the message from variable node s_n to factor node f_n is

$$\nu_{s_n\to f_n}(s_n) = \pi_n^{in}\delta(s_n - 1) + (1-\pi_n^{in})\delta(s_n), \tag{12}$$

where

$$\pi_n^{in} = \frac{\lambda_n^f\lambda_n^b}{(1-\lambda_n^f)(1-\lambda_n^b) + \lambda_n^f\lambda_n^b}. \tag{13}$$

Then the message from the factor node f_n to variable node h_n is

$$\begin{aligned}\nu_{f_n\to h_n}(h_n) &= \frac{\sum_{s_n=0,1} f_n(h_n, s_n)\nu_{s_n\to f_n}(s_n)}{\int\sum_{s_n=0,1} f_n(h_n', s_n)\nu_{s_n\to f_n}(s_n)dh_n'}\\ &= \pi_n^{in}\mathcal{CN}(h_n; 0, \sigma_h^2) + (1-\pi_n^{in})\delta(h_n).\end{aligned} \tag{14}$$

In Line 10 and 11, with the prior distribution $p(\boldsymbol{h}) = \prod_n \nu_{f_n\to h_n}(h_n)$ (messages from the channel support vector from (12) to (14)), the nonlinear estimations of $\boldsymbol{h}$ are given by $\boldsymbol{h}_B^{post}$ and v_B^{post}.

Compared with Turbo-CS, structured Turbo-CS requires additional calculations from Line 5 to Line 9 in Algorithm 1, which only includes component-wise multiplication with the computational complexity $\mathcal{O}(N)$. Therefore, the proposed structured Turbo-CS algorithm has the same order of computational complexity $\mathcal{O}(N\log N)$ as Turbo-CS [7].

IV. Performance Analysis Based on State Evolution

The performances of AMP and Turbo-CS can be characterized by simple scalar recursions called state evolution [7], [8], [11]. A similar technique can be applied to structured Turbo-CS by tracking the input variances v_A^{pri} and v_B^{pri} of Module A and Module B where

$$v_A^{pri} \triangleq \frac{1}{N}\mathrm{E}\left[|\boldsymbol{h}_A^{pri} - \boldsymbol{h}|^2\right] = \left(\frac{1}{\mathrm{MMSE}(v_B^{pri})} - \frac{1}{v_B^{pri}}\right)^{-1}, \quad (15)$$

and

$$v_B^{pri} \triangleq \frac{1}{N}\mathrm{E}\left[|\boldsymbol{h}_B^{pri} - \boldsymbol{h}|^2\right] = \frac{N}{M}(v_A^{pri} + \sigma^2) - v_A^{pri}, \quad (16)$$

where $\mathrm{MMSE}(v_B^{pri})$ in (15) is the MMSE achieved by the *posterior* mean estimation for the AWGN observation model $\boldsymbol{h}_B^{pri} = \boldsymbol{h} + \boldsymbol{w}$ in (8). Note that (15) is obtained from [7, (28b)], and (16) [7, (28a)] can be obtained by substituting Line 2 into Line 3 in Algorithm 1.

For the Markov prior $p(\boldsymbol{h})$ in (3) and (4), $\mathrm{MMSE}(v_B^{pri})$ can be written as

$$\mathrm{MMSE}(v_B^{pri}) = \frac{1}{N}\sum_{n=1}^{N}\mathrm{E}\left[|h_n - \mathrm{E}(h_n|\boldsymbol{h} + \boldsymbol{w})|^2\right]. \quad (17)$$

As a comparison, the state evolution of Turbo-CS in [7] is given by (15) and (16), with $\mathrm{MMSE}(v_B^{pri})$ replaced by[1]

$$\mathrm{MSE}(v_B^{pri}) = \frac{1}{N}\sum_{n=1}^{N}\mathrm{E}\left[\left|h_n - \mathrm{E}\left[h_n|h_n + w_n\right]\right|^2\right], \quad (18)$$

where $h_n \sim p(h_n) = \lambda\mathcal{CN}(h_n; 0, \sigma_h^2) + (1-\lambda)\delta(h_n), \forall n$. Note that estimator $\mathrm{E}(h_n|h_n + w_n)$ in (18) can be obtained from $\mathrm{E}(h_n|\boldsymbol{h} + \boldsymbol{w})$ by dropping observed data $\{h_i + w_i | i \neq n\}$ in the condition. Since dropping away observed data cannot improve the estimation accuracy, we generally have $\mathrm{MMSE}(v_B^{pri}) \leq \mathrm{MSE}(v_B^{pri})$.

In Figs. 3 and 4, we compare the state evolution of different algorithms, where the NMSE is defined as $\mathrm{NMSE} = \|\hat{\boldsymbol{h}} - \boldsymbol{h}\|_2^2/\|\boldsymbol{h}\|_2^2$. We also plot a lower bound (red dashed line), which is obtained by a genie-aided version of the proposed algorithm with knowledge of positions of non-zero elements. It is known that the state evolution analysis for both the AMP and Turbo-CS algorithms agrees well with simulation when N is large. For a relatively small N, the performance of message-passing based algorithm, including AMP, Turbo-CS and structured Turbo-CS, can not be accurately predicted by the state evolution [11], as illustrated in Fig. 4. However, the gap between the structured Turbo-CS algorithm and its state evolution is much smaller than those of Turbo-CS and AMP, which implies the robustness of the proposed algorithm even for a small N. The proposed algorithm also achieves faster convergence under a high SNR, and nearly 5 dB NMSE gain when the SNR is low. In practice, the number of antennas at the BS is usually a few hundred and the SNR is not very high. Therefore, AMP and Turbo-CS cannot work well for practical massive MIMO systems, and the structured Turbo-CS algorithm is more suitable for the channel estimation problem.

V. Test for a Realistic Channel Model

We compare the performance of the proposed algorithm with various baseline algorithms under the spatial channel model (SCM) [14] developed in 3 GPP. The parameters of the SCM used in simulation are listed in Table I. To apply the structured Turbo-CS algorithm in practical channel estimation, we update the statistical parameters $\{p_{01}, p_{10}, \sigma_n^2\}$ by using the expectation-maximization (EM) framework [13].

Figs. 5 and 6 compare the average NMSE performance for the OMP [4], CoSaMP [4], DSAMP [1], L1 LASSO [15], burst LASSO [2], EM-BG-AMP [13], Turbo-CS [7] and the proposed structured Turbo-CS

[1] This is because Turbo-CS assumes i.i.d. prior, while the true prior is the Markov prior.

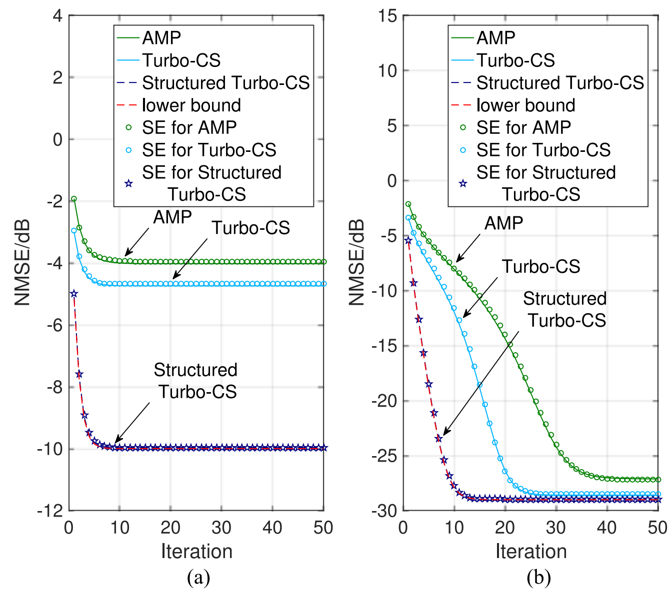

Fig. 3. Comparison of state evolution and simulation results under SNR = 10 dB in (a) and 30 dB in (b). Set $N = 2000$, $M = 800$, $p_{01} = 1/750$ and $p_{10} = 1/250$ (i.e., $\lambda = 0.25$) for simulation. i.i.d. complex Gaussian matrix is used for simulation of the AMP algorithm.

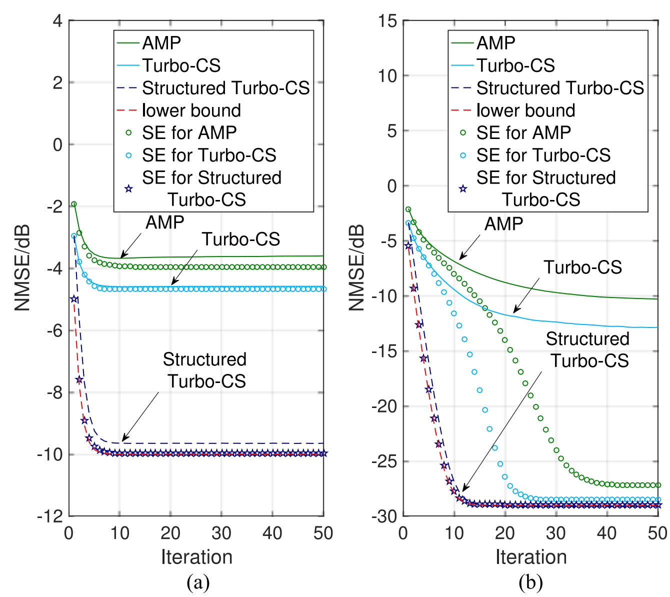

Fig. 4. Comparison of state evolution and simulation results under SNR = 10 dB in (a) and 30 dB in (b). Set $N = 128$, $M = 51$, $p_{01} = 1/96$ and $p_{10} = 1/32$ (i.e., $\lambda = 0.25$) for simulation. i.i.d. complex Gaussian matrix is used for simulation of the AMP algorithm.

TABLE I
Parameter Setting for Spatial Channel Model

Parameter name	Value	Parameter name	Value
NumBsElements	128	NumPaths	6
NumMsElements	1	Scenario	suburban macro
CenterFrequency	2E9 Hz		

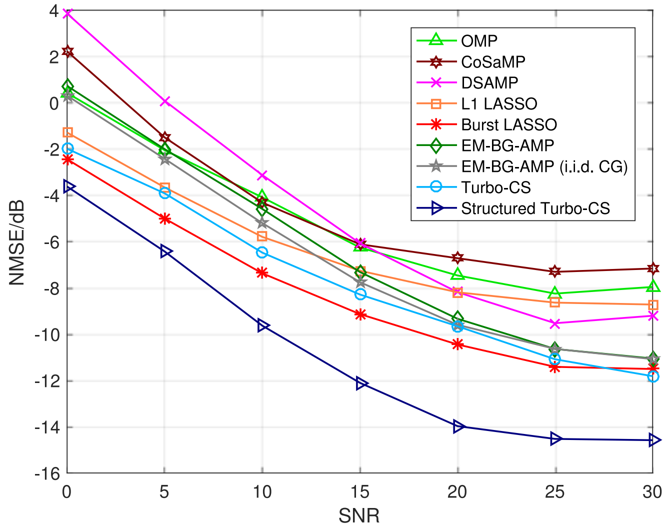

Fig. 5. NMSE of different algorithms versus SNR under the spatial channel model [14] with $N = 128$ antennas and $M = 45$ pilots. The results are averaged by 200 realizations.

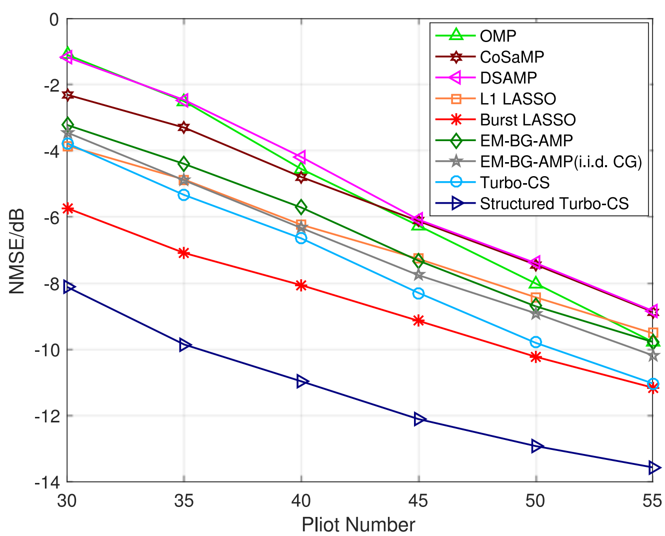

Fig. 6. NMSE of different algorithms versus the number of pilots M under the spatial channel model [14] with $N = 128$ antennas and SNR $= 15$ dB. The results are averaged by 200 realizations.

algorithms. In the simulations, for burst LASSO, we set the number of non-zero clusters as 6 and the average cluster size as 5 empirically based on the channel data generated by the SCM. For the message passing based algorithms, such as EM-BG-AMP and structured Turbo-CS, the required channel statistical parameters are automatically learned by the EM framework. When neither Turbo-CS nor structured Turbo-CS algorithms are used, the sensing matrix is still chosen as PDFT-RP matrix for a reasonable comparison. The performance of AMP with i.i.d. complex Gaussian (i.i.d. CG) sensing matrix is added to verify that the performance gain of the proposed algorithm is not only due to the choice of sensing matrix, but due to the fact that the design has exploited the specific sparse Markov structure. It can be seen that the proposed structured Turbo-CS achieves significant gain over all baseline algorithms, which demonstrates that the proposed algorithm combined with the EM framework is a powerful method for accurate channel estimation in practical massive MIMO systems.

VI. Conclusion

We use a Markov prior to model the structured sparsity of massive MIMO channels and propose a structured Turbo-CS algorithm to improve the channel estimation performance by incorporating the Markov prior information. Moreover, we derive the state evolution of the structured Turbo-CS algorithm, which can accurately predict the performance. The state evolution analysis shows that the proposed algorithm can achieve a significant gain over the state-of-the-art algorithms in terms of a larger operation region and lower channel estimation NMSE. Finally, we verify the superior performance of the proposed algorithm under a realistic channel model, which demonstrates that the Markov prior can indeed capture the structured sparsity of practical massive MIMO channels.

References

[1] 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.

[2] A. Liu, V. K. 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.

[3] Y. Han, J. Lee, and D. J. Love, "Compressed sensing-aided downlink channel training for FDD massive MIMO systems," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 2852–2862, Jul. 2017.

[4] M. F. Duarte and Y. C. Eldar, "Structured compressed sensing: From theory to applications," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4053–4085, Sep. 2011.

[5] Z. Gao, L. Dai, W. Dai, B. Shim, and Z. Wang, "Structured compressive sensing-based spatio-temporal joint channel estimation for FDD massive MIMO," *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 601–617, Feb. 2016.

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

[7] J. Ma, X. Yuan, and L. Ping, "Turbo compressed sensing with partial DFT sensing matrix," *IEEE Signal Process. Lett.*, vol. 22, no. 2, pp. 158–161, Feb. 2015.

[8] 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.

[9] F. R. Kschischang, B. J. 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.

[10] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[11] J. Ma and L. Ping, "Orthogonal AMP," *IEEE Access*, vol. 5, pp. 2020–2033, 2017.

[12] 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, 2009.

[13] J. P. Vila and P. Schniter, "Expectation-maximization gaussian-mixture approximate message passing," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, Oct. 2013.

[14] J. Salo *et al.*, "MATLAB implementation of the 3GPP spatial channel model (3GPP TR 25.996)," Jan. 2005. [Online]. Available: http://www.tkk.fi/Units/Radio/scm/

[15] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.