Structured Turbo Compressed Sensing for Downlink Massive MIMO-OFDM Channel Estimation

Xiaoyan Kuai, Lei Chen, Xiaojun Yuan, Senior Member, IEEE, and An Liu, Senior Member, IEEE

Abstract—Compressed sensing has been employed to reduce the pilot overhead for channel estimation in wireless communication systems. Particularly, structured turbo compressed sensing (STCS) provides a generic framework for structured sparse signal recovery with reduced computational complexity and storage requirement. In this paper, we consider the problem of massive multiple-input multiple-output (MIMO) orthogonal frequency division multiplexing (OFDM) channel estimation in a frequency division duplexing (FDD) downlink system. By exploiting the structured sparsity in the angle-frequency domain (AFD) and angle-delay domain (ADD) of the massive MIMO-OFDM channel, we represent the channel by using AFD and ADD probability models and design message-passing-based channel estimators under the STCS framework. Several STCS-based algorithms are proposed for massive MIMO-OFDM channel estimation by exploiting the structured sparsity. We show that, compared with other existing algorithms, the proposed algorithms have a much faster convergence speed and achieve competitive error performance under a wide range of simulation settings.

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

I. INTRODUCTION

MASSIVE multiple-input multiple-output (MIMO) techniques can be combined with orthogonal frequency division multiplexing (OFDM) to achieve huge performance gains in both spectrum and energy efficiency. Given that both MIMO and OFDM techniques have been already deployed in the existing commercialized wireless networks, massive MIMO-OFDM has been widely recognized as a high-priority option for future 5G wireless communications [2]–[4].

In massive MIMO-OFDM, the acquisition of accurate channel state information (CSI) is essential for harvesting the capacity and reliability enhancement promised by the system. However, conventional channel estimation approaches require that the pilot length should be at least the same as the number of transmit antennas [5], [6]. This may cause a significant pilot overhead in the downlink of a massive MIMO-OFDM system, where a large number of antennas are deployed at base station (BS). A possible solution to this problem is to assume time division duplexing (TDD), where the CSI only needs to be acquired in the uplink and then the downlink CSI is automatically obtained thanks to the channel reciprocity. However, on one hand, due to limited coherence time and the mismatch of uplink and downlink transmit-receive filters, the CSI acquired in the uplink might be inaccurate for the downlink transmission. On the other hand, it is economically disadvantageous to deploy TDD systems since frequency division duplexing (FDD) dominates the current cellular networks [7]. Therefore, it is of critical importance to reduce the pilot overhead for the downlink of FDD massive MIMO-OFDM systems.

Due to limited local scatterers in physical environments, a massive MIMO-OFDM channel usually exhibits abundant sparsity in certain transformed domains [8], [9]. Compressed sensing (CS) algorithms, such as orthogonal matching pursuit (OMP) [10] and least absolute shrinkage and selection operator (LASSO) algorithm [9], have recently been used to exploit the sparsity in the channel estimation of massive MIMO-OFDM, so as to reduce the pilot overhead. In particular, a burst LASSO algorithm is developed in [11] for clustering the nonzero channel coefficients in the virtual angle domain of a massive MIMO channel. By exploiting temporal correlation of a massive MIMO channel, the CS algorithms in [12] and [13] further reduce the pilot overhead. In [14], an algorithm named distributed sparsity adaptive matching pursuit (DSAMP) was proposed to jointly estimate the channel coefficients of multiple subcarriers based on the common sparsity in the frequency domain. In [7], channel estimation is designed on account of the temporal correlation of the sparsity in the delay domain.

However, the above CS-based channel estimation algorithms, when applied to FDD downlink massive MIMO-OFDM, have their respective drawbacks. For example, OMP...

...
and LASSO do not take into account the structured sparsity in the algorithm design. The burst LASSO algorithm, which considers group sparsity in the virtual angle domain, only works well when non-zero clusters of the channel coefficients have a similar size. The DSAMP algorithm considers the common sparsity between different subcarriers, but unfortunately does not exploit the sparse structure in the angle domain.

Recently, message-passing algorithms for compressed sensing [15]–[23] have attracted much research interest due to their fast convergence and low computational complexity. Among them, the turbo compressed sensing (Turbo-CS) algorithm [17] and its variants [19], [20] have the state-of-the-art performance in both complexity and convergence rate, especially when partial orthogonal sensing matrices are involved. In particular, the authors in [20] proposed a modified Turbo-CS algorithm, termed structured turbo compressed sensing (STCS), in which Turbo-CS is combined with a Markov model to efficiently exploit the clustered sparsity of the massive MIMO channel in the angle domain. In this paper, the main contributions of our work are summarized as follows.

- We extend STCS for the channel estimation of massive MIMO-OFDM by exploiting structured channel sparsity not only in the angle-frequency domain, but also in the angle-delay domain.\(^1\)
- We develop a Markov chain as the probability model to characterize the structured sparsity of massive MIMO-OFDM channels in the angle-frequency domain. The resulting algorithm is referred to as STCS with frequency support (STCS-FS). We further develop a Markov model with two types of hidden state variables to describe the structured sparsity of the massive MIMO-OFDM channel in the angle-delay domain. The resulting algorithm is referred to as STCS with delay support (STCS-DS). We develop state evolution (SE) to accurately predict the performance of the STCS-FS and STCS-DS algorithms.
- We present simulation results that demonstrate the advantages of the STCS-FS and STCS-DS algorithms. In particular, we show that STCS-DS exhibits the fastest convergence rate among all the existing CS-based iterative algorithms, and achieves competitive mean-square error (MSE) performance. Both algorithms have been validated under realistic channel models.

\(^1\)The structured compressed sensing algorithm exploits the sparsity structure in the angle-delay domain of the massive MIMO-OFDM system, while the structured compressed sensing algorithm in [7] exploits the spatial temporal common sparsity of the MIMO-OFDM system.

Second, the probability process in this paper is used to model the clustering property of the channel coefficient support in the angle-frequency domain and angle-delay domain, while the Markov process is used to model the time-varying coefficient support and the time-varying coefficient amplitudes in [22]. Third, the Turbo-CS algorithm is used in our paper, while approximate message passing (AMP) was used to recover the sparse signal in [22]. The turbo-CS algorithm has lower complexity and exhibits faster convergence speed than the AMP algorithm [19].

Compared with the recent work [23], the novelty of our work consists of the following aspects. First, we employ a Markov model to efficiently exploit the clustered sparsity of the massive MIMO channel in the angle-frequency domain and angle-delay domain, while [23] uses the nearest neighbor sparsity pattern learning (NNSPL) algorithm first proposed in [24] to exploit the sparsity structure.\(^2\) Second, STCS-FS achieves a considerably lower mean square error (MSE) performance than the NNSPL algorithm with frequency support, while STCS-DS performs slightly better than the NNSPL algorithm with delay support. Third, the computational complexity of STCS is much lower than that of NNSPL. In this regard, we show that the per-iteration complexity of STCS is lower than that of NNSPL; we further show that STCS converges much faster than NNSPL.

### B. Organization

The rest of the paper is organized as follows. In Section II, we present the massive MIMO-OFDM channel model and the joint sparsity in the angle-frequency and angle-delay domain. In Section III, we review the STCS framework. In Sections IV and V, the details of the angle-frequency domain and angle-delay domain channel support models are elaborated. Based on these models, we design the STCS-based algorithms. Simulation results and conclusions are presented in Sections VI and VII, respectively.

### II. System Model

#### A. Massive MIMO-OFDM

Consider a typical massive MIMO-OFDM system with one BS serving multiple single-antenna users, where the BS comprises \(N\) antennas and the system employs \(P\) pilot subcarriers. Without loss of generality, we focus on the downlink channel estimation problem at a reference user. To estimate the downlink channel \(\hat{h}_f^{(p)} \in \mathbb{C}^{1 \times 1}\) of the reference user at the \(p\)-th pilot subcarrier in the frequency domain, the BS sends \(M\) training symbols \(x_{m}^{(p)} \in \mathbb{C}^{1 \times 1}, 1 \leq m \leq M\) over successive time slots. Then the received signal at the reference user \(y_{f}^{(p)} \in \mathbb{C}^{1 \times 1}\) can be written as

\[
y_{f}^{(p)} = X^{(p)}\hat{h}_f^{(p)} + w_f^{(p)}, \quad 1 \leq p \leq P, \tag{1}
\]

where \(X^{(p)} = [x_1^{(p)}, \ldots, x_M^{(p)}]^{T}\) is an \(N \times M\) pilot matrix, and \(w_f^{(p)} \sim \mathcal{CN}(0, \sigma^2 I)\) is an additive white Gaussian noise.

\(^2\)Note that [25] uses the NNSPL algorithm to exploit the angle domain sparsity of the channel, while [26] uses the NNSPL algorithm to exploit the delay domain sparsity. [23] presents a comprehensive version of the NNSPL algorithm to jointly handle the angle-delay domain sparsity.
noise (AWGN). From [7], [11], [27], $\tilde{h}^{(p)}_f$ is sparse in the angle domain, i.e., $\tilde{h}^{(p)}_f$ can be expressed as

$$\tilde{h}^{(p)}_f = Bh^{(p)}_f, \quad 1 \leq p \leq P,$$

where $B$ is the transform matrix determined by the geometrical structure of the antenna array, and $h^{(p)}_f$ is a sparse representation of the channel in the transform domain. In this paper, we focus on the half-wavelength uniform linear array (ULA) at BS, where $B$ is the inverse discrete Fourier transform (DFT) matrix [27]. Our discussion can be readily extended to a uniform planar array (UPA) or higher-dimensional antenna array. Substituting (2) and letting $A^{(p)} = X^{(p)}B$, we can rewrite (1) as

$$y^{(p)}_f = A^{(p)}h^{(p)}_f + w^{(p)}_f, \quad 1 \leq p \leq P.$$  \hspace{1cm} (3)

Our goal is to estimate the sparse vectors $h^{(p)}_f$ from the low-dimensional observed signal $y^{(p)}_f$, $1 \leq p \leq P$. This problem can be solved by the existing compressed sensing algorithms [7], [9], [11], [14]–[17], [20]. However, these existing algorithms, if directly applied, can not efficiently exploit the unique sparsity structure of the massive MIMO-OFDM channel, as detailed below.

B. Channel Sparsity

Due to the scattering effect, a massive MIMO-OFDM channel exhibits clustered sparsity. Besides, the scatterers for different subchannels are very similar [27]. Consequently, for a communication system with the bandwidth much smaller than the carrier frequency (e.g., 10 MHz in LTE-A systems with a carrier frequency of 2 GHz), the subchannels $\{h^{(p)}_f\}_{p=1}^P$ have a common support for sparsity [14], i.e.

$$\text{supp}(h^{(1)}_f) = \text{supp}(h^{(2)}_f) = \cdots = \text{supp}(h^{(P)}_f),$$  \hspace{1cm} (4)

where $\text{supp}(h^{(p)}_f)$ returns the positions of the non-zero entries of $h^{(p)}_f$. As an example, in Fig. 1 (a), we generate a massive MIMO-OFDM channel using the spatial channel model (SCM) [28] with carrier frequency at 2 GHz, bandwidth 7.5 MHz and frequency interval 15 kHz. There are 512 subcarriers in total, and 64 of them are chosen as pilot subcarriers. It is clear that massive MIMO-OFDM subchannels have a common support in the frequency domain, and the non-zero elements appear in a clustered manner in the angle domain.

A limited number of scatterers also cause sparsity in the delay domain [7]. We can transform the channel response matrix from the angle-frequency domain to the angle-delay domain with an inverse Fourier transform [29], [30], i.e.

$$H_f F^* = H_d,$$  \hspace{1cm} (5)

where $H_f = [h^{(1)}_f, \cdots, h^{(P)}_f]$, $H_d = [h^{(1)}_d, \cdots, h^{(P)}_d]$, $F$ denotes the $P \times P$ unitary DFT matrix, and $(\cdot)^*$ denotes the conjugate operation. Without loss of generality, let $L$ be the maximum delay spread, implying $h^{(p)}_d = 0$ for

$$p = L + 1, \cdots, P.$$  

Fig. 1. An example of the spatial channel model in [28]. Subfigure (a) plots the channel gains in the angle-frequency domain, and subfigure (b) plots the channel gains in the angle-delay domain. All parameters in the model are set by default, except for ‘NumBsElements’ = 256, ‘NumMsElements’ = 1, and ‘Scenario’ = Urban-macro.

III. STRUCTURED TURBO COMPRESSED SENSING

The goal of this paper is to estimate $H_f$ based on the observed signal $Y_f = [y^{(1)}_f, y^{(2)}_f, \cdots, y^{(P)}_f]$ together with the sparsity of $H_f$ described in Section II-B. We will mainly follow the STCS approach in [17], [19], [20] to solve the above problem. For self-containedness, we present the STCS algorithm in the following. For notational convenience, we drop the subscripts of $H_f$ and $Y_f$, since later we will apply the STCS algorithm to the delay domain representations of $H_f$ and $Y_f$.

The STCS algorithm contains two modules, namely, Module A and Module B. Module A is basically a linear minimum mean square error (LMMSE) estimator based on the observation $Y$ and the messages from Module B. Module B refines the estimate of the channel by combining the messages from Module A and the prior distribution of $H$. The two modules are executed iteratively until convergence.

In Module A, the channel vector $h^{(p)}$ (the $p$-th column of matrix $H$) is estimated based on the observation $y^{(p)}$ (the $p$-th column of matrix $Y$) with a prior distribution...
the probability distribution for non-zero coefficients. In this paper, function $G$ is chosen as $CN(h_{f,n}^{(p)}, 0, (\sigma_f^{(p)})^2)$. Define a vector $s_f = [s_{f,1}, \ldots, s_{f,N}]^T$. Then the clustering effect of non-zeros can be modeled using a Markov chain as

$$p(s_f) = p(s_{f,1}) \prod_{n=2}^{N} p(s_{f,n}|s_{f,n-1}),$$

with the transition and initial probabilities given by

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

and

$$p(s_{f,1}) = (1-\lambda_f)^{1-s_{f,1}}(\lambda_f)^{s_{f,1}},$$

where $\lambda_f = \Pr(s_{f,n} = 1) = (1+p_{01}/p_{10})^{-1}$, the average ratio of the non-zero elements in $s_f$, describes the sparsity of $h_{f,n}^{(p)}$ for all $p$. Such a Markov chain is fully described by parameters $p_{10} = \Pr(s_{f,n} = 1 | s_{f,n-1} = 0)$ and $p_{01} = \Pr(s_{f,n} = 0 | s_{f,n-1} = 1)$. Since $p_{00} = 1-p_{10}$, a smaller $p_{10}$ implies a larger gap between two clusters. Similarly, with $p_{11} = 1-p_{01}$, a smaller $p_{01}$ implies a larger average cluster size.

### B. Message Passing for Module B

In this subsection, we explain the details of Module B for the frequency support model in the angle-frequency domain. First of all, a basic assumption is used to model $h_{f,n}^{(p)}$, the input mean of Module B in (9), as

$$h_{f,n}^{(p)} = h_{f}^{(p)} + n_{f,n}^{(p)}, \quad \forall p,$n

where $n_{f,n}^{(p)} \sim CN(0, (\sigma_f^{(p)})^2)$ is independent of $h_{f}^{(p)}$, and $h_{f,n}^{(p)}$ is the input variance of Module B in (10). Similar assumptions have been used in message-passing-based iterative signal recovery algorithm [15]–[17], [19], [20], [32]. Under this assumption, the factor graph of the joint probability distribution

$$p(H_{B}^{(p)}, H_{f}^{(p)}, s_f) = p(H_{B}^{(p)}|H_{f}^{(p)})p(H_{f}^{(p)}|s_f)p(s_f),$$

$$= \prod_{p=1}^{P} \prod_{n=1}^{N} p(h_{B,n}^{(p)}|h_{f,n}^{(p)}) \prod_{p=1}^{P} \prod_{n=1}^{N} p(h_{f,n}^{(p)}|s_{f,n}),$$

$$\times \prod_{n=2}^{N} p(s_{f,1}) \prod_{n=2}^{N} p(s_{f,n}|s_{f,n-1}),$$

denoted by $G_f$, is shown in Fig. 3, where the factor function of each factor node is listed in Table I.

We now give a message passing algorithm based on graph $G_f$. According to the sum-product rule, the message from variable node $h_{f,n}^{(p)}$ to factor node $f_{n}^{(p)}$ is

$$\nu_{h_{n}^{(p)}}^{(p)}(h_{f,n}^{(p)}) = CN(h_{f,n}^{(p)}; h_{B,n}^{(p)}, \nu_{B}^{(p)}),$$

$$\nu_{h_{B,n}^{(p)}}^{(p)}(h_{f,n}^{(p)}) = CN(h_{f,n}^{(p)}; h_{B,n}^{(p)}, \nu_{B}^{(p)}).$$
and the message from factor node $f^{(p)}_{f,n}$ to variable node $s_{f,n}$ is

$$
\nu^{(p)}_{s_{f,n} \rightarrow f_{f,n}}(s_{f,n}) \propto \int h^{(p)}_{f,n}(h^{(p)}_{f,n}, s_{f,n}) \cdot \nu^{(p)}_{h^{(p)}_{f,n} \rightarrow f_{f,n}}
$$

$$
= \frac{-\nu^{(p)}}{\pi_n} s_{f,n} + (1 - \frac{-\nu^{(p)}}{\pi_n})(1 - s_{f,n}),
$$

(18)

where $f^{(p)}_{f,n}$ is given by Table I, and

$$
\frac{-\nu^{(p)}}{\pi_n} = \left( 1 + \frac{\mathcal{CN}(0; h^{(p)}_{B,n}v^{(p)}_{B})}{\mathcal{CN}(0; h^{(p)}_{B,n}v^{(p)}_{B} + (\sigma_f^{(p)})^2)} \right)^{-1}.
$$

(19)

Then forward-backward message passing is performed over the binary Markov chain $s_f$, with the forward and backward messages respectively given by

$$
\nu_{s_{f,n} \rightarrow f_{f,n}} \propto \sum_{s_{f,n-1}} d_{f,n}(s_{f,n}, s_{f,n-1})
$$

$$
\times \nu_{s_{f,n-1} \rightarrow s_{f,n-1}} \prod_{p=1}^P \nu^{(p)}_{f_{f,n} \rightarrow s_{f,n-1}^p}
$$

$$
= \lambda^{(p)}_n s_{f,n} + (1 - \lambda^{(p)}_n)(1 - s_{f,n}),
$$

(20)

and

$$
\nu_{f_{f,n} \rightarrow s_{f,n}} \propto \sum_{s_{f,n+1}} d_{f,n}(s_{f,n+1}, s_{f,n})
$$

$$
\times \nu_{f_{f,n+1} \rightarrow s_{f,n+1}} \prod_{p=1}^P \nu^{(p)}_{f_{f,n} \rightarrow s_{f,n+1}^p}
$$

$$
= \lambda^{(p)}_n s_{f,n} + (1 - \lambda^{(p)}_n)(1 - s_{f,n}),
$$

(21)

where

$$
\lambda^{(p)}_n = \frac{(1 - p_{10}) \sum_{p=1}^P \prod_{p=1}^{\frac{-\nu^{(p)}}{\pi_n}} + p_{10}}{1 - \sum_{p=1}^P \prod_{p=1}^{\frac{-\nu^{(p)}}{\pi_n}} + 1},
$$

(22)

and

$$
\lambda^{b}_n = \frac{(1 - p_{10}) \lambda^{(p)}_n \sum_{p=1}^P \prod_{p=1}^{\frac{-\nu^{(p)}}{\pi_n}} + p_{10}}{1 - \sum_{p=1}^P \prod_{p=1}^{\frac{-\nu^{(p)}}{\pi_n}} + (1 - p_{10} + p_{10})},
$$

(23)

with $\lambda^{(p)}_f = \lambda_f$ and $\lambda^{b}_N = 1/2$.

After that, according to the sum-product rule, the message from variable node $s_{f,n}$ to factor node $f^{(p)}_{f,n}$ is

$$
\nu^{(p)}_{s_{f,n} \rightarrow f^{(p)}_{f,n}}(s_{f,n}) \propto \nu_{d_{f,n} \rightarrow s_{f,n}} \nu_{d_{f,n+1} \rightarrow s_{f,n}} \prod_{p' \neq p} \nu^{(p')}_{f^{(p')}_{f,n} \rightarrow s_{f,n}}
$$

$$
= \frac{-\nu^{(p)}}{\pi_n} s_{f,n} + (1 - \frac{-\nu^{(p)}}{\pi_n})(1 - s_{f,n}),
$$

(24)

where

$$
\frac{-\nu^{(p)}}{\pi_n} = \left( 1 - \lambda^{(p)}_n \sum_{p' \neq p} \prod_{p' \neq p} \frac{-\nu^{(p)}}{\pi_n} \right).
$$

(25)

The message from the factor node $f^{(p)}_{f,n}$ back to variable node $h^{(p)}_{f,n}$ is

$$
\nu^{(p)}_{f^{(p)}_{f,n} \rightarrow h^{(p)}_{f,n}}(h^{(p)}_{f,n}) \propto \sum_{s_{f,n}} f^{(p)}_{f,n}(h^{(p)}_{f,n}, s_{f,n}) \cdot \nu^{(p)}_{s_{f,n} \rightarrow f^{(p)}_{f,n}}
$$

$$
= \frac{-\nu^{(p)}}{\pi_n} \mathcal{CN}(h^{(p)}_{f,n}; 0, (\sigma_f^{(p)})^2)
$$

$$
+ (1 - \frac{-\nu^{(p)}}{\pi_n}) \delta(h^{(p)}_{f,n}).
$$

(26)

The posterior mean and variance can be calculated as

$$
\hat{h}^{\text{post}}_{B,n} = \mathbb{E}(h^{(p)}_{f,n} | H^{\text{prior}}_{B}) = \int h^{(p)}_{f,n} \nu^{(p)}_{h^{(p)}_{f,n} | H^{\text{prior}}_{B}} d h^{(p)}_{f,n},
$$

(27)

and

$$
\text{Var}(h^{(p)}_{f,n} | H^{\text{prior}}_{B}) = \frac{1}{N} \sum_{n=1}^N \text{Var}(h^{(p)}_{f,n} | H^{\text{prior}}_{B})
$$

$$
= \frac{1}{N} \sum_{n=1}^N \int \nu^{(p)}_{h^{(p)}_{f,n} | H^{\text{prior}}_{B}} d h^{(p)}_{f,n} - \hat{h}^{\text{post}}_{B,n} \nu^{(p)}_{h^{(p)}_{f,n} | H^{\text{prior}}_{B}} d h^{(p)}_{f,n},
$$

(28)
where the conditional distribution is
\[
p(h_f^{(p)} | H_B^{(p)}) \propto \nu_f^{(p)} - h_f^{(p)} \nu_f^{(p)} - h_f^{(p)},
\]
with \( \nu_f^{(p)} - h_f^{(p)} = \mathcal{CN}(h_f^{(p)}; h_{B,n}^{(p)} v_f^{(p)}) \).

Based on the derivation in [17], [19], the corresponding extrinsic update can be calculated as
\[
v_A^{\text{post}} = v_A^{\text{ext}} = \left( \frac{1}{v_B^{\text{post}}} - \frac{1}{v_B^{\text{post}}} \right)^{-1},
\]
and
\[
h_A^{\text{post}} = h_A^{\text{ext}} = v_A^{\text{post}} \left( \frac{h_B^{\text{post}}}{v_B^{\text{post}}} - h_A^{\text{post}} \right).
\]
The structured Turbo-CS algorithm with Module B realized by Eqs. (17) to (31) is referred to as structured Turbo-CS with frequency support (STCS-FS). The STCS-FS algorithm is summarized in Algorithm 1.

**Algorithm 1** Structured Turbo-CS Algorithm With Frequency Support (STCS-FS)

**Input:** received signal \( Y_f = [y_f^{(1)}, \cdots, y_f^{(P)}] \), pilot matrix \( X^{(p)} \) \( \forall p \), and additive noise variance \( \sigma^2 \).

**Output:** channel state information \( \mathcal{H} \).

**Initialize:** \( A^{(p)} = X^{(p)} F^H, h_A^{\text{post}}, v_A^{\text{post}}, \forall p \).

**Module A:**

**% LMMSE estimator**
1: \( h_A^{\text{post}} = h_A^{\text{extr}} + \frac{v_A^{\text{extr}}}{v_A^{\text{extr}} + \sigma^2} A^{(p)H} (y_f^{(p)} - A^{(p)} h_A^{\text{extr}}), \forall p \).
2: \( v_A^{\text{post}} = v_A^{\text{extr}} - \frac{M}{N} (v_A^{\text{extr}})^2 (v_A^{\text{extr}} + \sigma^2), \forall p \).

**% Update extrinsic messages**
3: \( v_A^{\text{extr}} = e_A^{\text{extr}} = \left( \frac{1}{v_A^{\text{post}}} - \frac{1}{v_A^{\text{post}}} \right)^{-1}, \forall p \).
4: \( h_B^{\text{extr}} = h_B^{\text{extr}} = v_A^{\text{extr}} \left( \frac{h_B^{\text{post}}}{v_B^{\text{post}}} - h_A^{\text{extr}} \right), \forall p \).

**Module B:**

**% Structured estimator**
5: \( h_B^{\text{post}} \) and \( v_B^{\text{post}} \), \( \forall p \), are given by (27) and (28).

**% Update extrinsic messages**
6: \( v_A^{\text{extr}} = e_A^{\text{extr}} = \left( \frac{1}{v_B^{\text{post}}} - \frac{1}{v_B^{\text{post}}} \right)^{-1}, \forall p \).
7: \( h_A^{\text{extr}} = h_A^{\text{extr}} = v_A^{\text{extr}} \left( \frac{h_B^{\text{post}}}{v_B^{\text{post}}} - h_A^{\text{extr}} \right), \forall p \).

Repeat Module A and Module B until convergence or the maximum iteration number is exceeded.

**V. DELAY DOMAIN CHANNEL SUPPORT MODEL**

**A. Probability Model**

In this section, we establish a probability model to directly characterize the angle-delay domain sparsity. To start with, we transform the system model in (3) into the angle-delay domain. Define
\[
Y_f = [y_f^{(1)}, \cdots, y_f^{(P)}], \quad H_f = [h_f^{(1)}, \cdots, h_f^{(P)}], \quad W_f = [w_f^{(1)}, \cdots, w_f^{(P)}].
\]

As the pilot matrix can be designed in advance, for simplification, we assume that \( X^{(1)} = \cdots = X^{(P)} \), i.e., the matrix \( X \) is the same for different pilot subcarrier. \( X^{(p)} \) is hence replaced by \( X \) for \( 1 \leq p \leq P \). The received signal in the angle-frequency domain is represented as
\[
Y_f = AH_f + W_f.
\]

We next transform the channel response matrix from the angle-frequency domain to the angle-delay domain with an inverse Fourier transform \( F^\ast \), i.e., \( H_f F^* = H_d \) in (5). Then the received signal in the delay domain can be represented as
\[
Y_f F^* = AH_f F^* + W_f F^*,
\]
or equivalently,
\[
Y_d = AH_d + W_d.
\]

From (34), the \( p \)-th column of \( Y_d \) is given by
\[
y_d^{(p)} = Ah_d^{(p)} + w_d^{(p)}, \quad 1 \leq p \leq P,
\]
where \( y_d^{(p)} \) is the received signal in the delay domain, \( w_d^{(p)} \sim \mathcal{CN}(0, \sigma^2 I) \) is an AWGN with the same variance as \( w_f^{(p)} \), and \( h_d^{(p)} \) is the channel coefficient vector in the delay domain.\(^5\)

In the angle-delay domain, two hidden binary states are introduced to model the non-zero columns and cluster structure of the delay domain channel matrix \( H_d \). Each channel coefficient \( h_{d,n}^{(p)} \) has a conditionally independent distribution expressed as
\[
p(h_{d,n}^{(p)} | s_{d,n}^{(p)}) = (1 - s_{d,n}^{(p)}) \delta(h_{d,n}^{(p)})
+ s_{d,n}^{(p)} \mathcal{CN}(h_{d,n}^{(p)}; 0, (\sigma_d^{(p)})^2),
\]
where \( s_{d,n}^{(p)} \in \{0, 1\} \) is a hidden binary state. What is different from the angle-frequency model is that the binary state \( s_{d,n}^{(p)} \) is also conditioned on another binary state \( t_d^{(p)} \in \{0, 1\} \). Specifically, \( t_d^{(p)} \) indicates whether the \( p \)-th column of \( H_d \) is zero \( (t_d^{(p)} = 0) \) or not \( (t_d^{(p)} = 1) \). Hence, the vector \( t_d^{(p)} = [t_d^{(1)}, \cdots, t_d^{(P)}] \) can be used to capture the channel sparsity in the delay domain and each entry \( t_d^{(p)} \) complies with a Bernoulli distribution
\[
p(t_d^{(p)} = 1 - \gamma_d^{(p)}) i_{d}^{(p)} \gamma_d^{(p)} i_{d}^{(p)},
\]
where \( \gamma_d^{(p)} \) denotes the probability that the \( p \)-th column in \( H_d \) is non-zero. On the other hand, \( s_{d,n}^{(p)} \) indicates whether the \( (n,p) \)-th element of \( H_d \) is zero \( (s_{d,n}^{(p)} = 0) \) or not \( (s_{d,n}^{(p)} = 1) \). Hence, the vector \( s_d^{(p)} = [s_{d,1}, \cdots, s_{d,N}]^T \) can be used to capture the clustered sparsity in the angle domain for the channel vector \( h_d^{(p)} \). Specifically, conditioned on \( t_d^{(p)} \), the cluster structure of \( h_d^{(p)} \) can be modeled using a Markov chain as
\[
p(s_{d,n}^{(p)} | s_{d,n-1}^{(p)}, t_d^{(p)}) = p(s_{d,n}^{(p)} | s_{d,n-1}^{(p)} \cdot \cdot \cdot s_{d,1}^{(p)}),
\]
\(^5\)The STCS algorithms in this paper can be extended to the system model such as (13) in [21]. However, this involves more complicated signal processing since then the path delay taps are mixed in the channel output.
with the transition and initial probabilities given by
\[
p(s_{d,n}^{(p)}|s_{d,n-1}^{(p)},t_{d,n}^{(p)}) = \begin{cases} 
(1 - p_{10}^{(p)}) & s_{d,n-1}^{(p)} = 0, t_{d,n}^{(p)} = 1; \\
(1 - p_{01}^{(p)}) & s_{d,n-1}^{(p)} = 1, t_{d,n}^{(p)} = 1; \\
1 - s_{d,n}^{(p)} & s_{d,n-1}^{(p)} = 0, t_{d,n}^{(p)} = 0; \\
1 - s_{d,n}^{(p)} & s_{d,n-1}^{(p)} = 1, t_{d,n}^{(p)} = 0.
\end{cases}
\] (39)

and
\[
p(s_{d,n}^{(p)}|t_{d,n}^{(p)}) = (1 - \lambda_d^{(p)}) (1 - s_{d,n}^{(p)}) (\lambda_d^{(p)}) s_{d,n}^{(p)},
\]
\[
	imes 1(1 - s_{d,n}^{(p)}) (1 - t_{d,n}^{(p)}) 0^{s_{d,n}^{(p)}} 1^{1-t_{d,n}^{(p)}}.
\] (40)

In other words, when \(t_{d,n}^{(p)} = 0\), we must have \(s_{d,n}^{(p)} = 0\). When \(t_{d,n}^{(p)} = 1\), \(s_{d,n}^{(p)}\) is a binary Markov chain similar to \(s_{f}\) but with different transition probabilities. The probability model is illustrated as a factor graph in Fig. 4.

B. Message Passing for Module B

Similarly to (15), we assume
\[
h_{d,n}^{\text{pri}(p)} = h_d^{(p)} + n_d^{(p)}, \quad \forall p,
\] (41)

where \(n_d^{(p)} \sim \mathcal{CN}(0, \nu_d^{(p)} I)\) is independent of \(h_d^{(p)}\). The factor graph of the joint distribution
\[
p(H_d^{\text{pri}}, H_d, S_d, t_d)
\]
\[
= \prod_{p=1}^{P} p(h_d^{\text{pri}(p)}, h_d^{(p)}, s_d^{(p)}, t_d^{(p)})
\]
\[
= \prod_{p=1}^{P} \prod_{n=1}^{N} p(h_d^{\text{pri}(p)}|h_d^{(p)}) \prod_{n=1}^{N} p(h_d^{(p)}|s_d^{(p)})
\]
\[
\cdot p(s_d^{(p)}|s_{d,n-1}^{(p)}, t_d^{(p)}) p(t_d^{(p)})
\] (42)
denoted by \(G_d\), is shown in Fig. 4, where the function of each factor node is listed in Table II. The probability model in (42) assumes that the columns of \(H_d\) are independent of each other. This is justified by the fact that in practical scenarios, the channel coefficients for different delay taps usually experience significantly different channel fading.

We now derive the message passing algorithm on graph \(G_d\). Note that the functions of \(d_{d,n}^{(p)}(s_d^{(p)}, s_{d,n-1}^{(p)})\) and \(d_{d,n}^{(p)}(s_d^{(p)}, s_{d,n-1}^{(p)}, t_d^{(p)})\) are modified by replacing 1 and 0 with \(1 - \varepsilon\) and \(\varepsilon\), where \(\varepsilon\) is a small constant. Such a modification is used in [22] to avoid improper probability distribution functions and make the algorithm more robust.

We start with message passing from variable \(t_d^{(p)}\) to factor node \(d_{d,n}^{(p)}(s_d^{(p)}, s_{d,n-1}^{(p)}, t_d^{(p)})\) or \(d_{d,n}^{(p)}(s_d^{(p)}, s_{d,n-1}^{(p)}, t_d^{(p)})\):
\[
V_{d,n}^{(p)}(s_d^{(p)}|s_{d,n-1}^{(p)}, t_d^{(p)}) \propto V_{d,n}^{(p)}(s_d^{(p)}|s_{d,n-1}^{(p)}, t_d^{(p)}) \prod_{n' \neq n} V_{d,n'}^{(p)}(s_d^{(p)}|s_{d,n'-1}^{(p)}, t_d^{(p)})
\]
\[
\times \left[ \gamma_d^{(p)} + (1 - \gamma_d^{(p)}) (1 - t_d^{(p)}) \right] \prod_{n' \neq n} \left[ \theta_{n'} s_d^{(p)} + (1 - \theta_{n'}) \right] t_d^{(p)}
\]
\[
= \theta_n^{(p)} s_d^{(p)} + (1 - \theta_n^{(p)}) t_d^{(p)},
\] (43)

where
\[
\theta_n^{(p)} = \frac{\gamma_d^{(p)} \prod_{n' \neq n} \theta_n^{(p)}}{\gamma_d^{(p)} \prod_{n' \neq n} \theta_n^{(p)} + (1 - \gamma_d^{(p)}) \prod_{n' \neq n} (1 - \theta_n^{(p)})}.
\] (44)
The message from factor node $f_{d,n}^{(p)}$ to variable node $s_{d,n}^{(p)}$ is given by
\[
\nu_{f_{d,n}^{(p)} \rightarrow s_{d,n}^{(p)}}(s_{d,n}^{(p)}) \propto \int_{h_{d,n}^{(p)}} f_{d,n}^{(p)}(h_{d,n}^{(p)}, s_{d,n}^{(p)}) \cdot \nu_{h_{d,n}^{(p)} \rightarrow f_{d,n}^{(p)}}(s_{d,n}^{(p)})
\]
\[
= \frac{\pi_n}{\pi_n} s_{d,n}^{(p)} + (1 - \frac{\pi_n}{\pi_n})(1 - s_{d,n}^{(p)}).
\] (45)

Then the forward-backward message passing can be applied in the Markov chains according to the sum-product rule. Note that $s_{d,n}^{(p)}$ and $t_{d,n}^{(p)}$ are binary variables. The messages passed between $\{s_{d,n}^{(p)}\}$ and $\{t_{d,n}^{(p)}\}$ are given by
\[
\nu_{d,n}^{(p)} \rightarrow s_{d,n}^{(p)} = \lambda_n^{f_{d,n}^{(p)}} s_{d,n}^{(p)} + (1 - \lambda_n^{f_{d,n}^{(p)}})(1 - s_{d,n}^{(p)}),
\] (46)
\[
\nu_{d,n}^{(p)} \rightarrow d_{n+1}^{(p)} = \lambda_n^{f_{d,n}^{(p)}} s_{d,n}^{(p)} + (1 - \lambda_n^{f_{d,n}^{(p)}})(1 - s_{d,n}^{(p)}),
\] (47)
\[
\nu_{d,n}^{(p)} \rightarrow s_{d,n}^{(p)} = \lambda_n^{f_{d,n}^{(p)}} s_{d,n}^{(p)} + (1 - \lambda_n^{f_{d,n}^{(p)}})(1 - s_{d,n}^{(p)}),
\] (48)
and
\[
\nu_{d,n}^{(p)} \rightarrow d_{n+1}^{(p)} = \lambda_n^{f_{d,n}^{(p)}} s_{d,n}^{(p)} + (1 - \lambda_n^{f_{d,n}^{(p)}})(1 - s_{d,n}^{(p)}).
\] (49)

After that, we calculate the messages going out of the Markov chains and the message back to factor node $f_{d,n}^{(p)}$. The message from variable node $s_{d,n}^{(p)}$ to factor node $f_{d,n}^{(p)}$ is
\[
\nu_{s_{d,n}^{(p)} \rightarrow f_{d,n}^{(p)}}(s_{d,n}^{(p)}) \propto \nu_{s_{d,n}^{(p)} \rightarrow s_{d,n}^{(p)}}(s_{d,n}^{(p)}) \cdot \nu_{s_{d,n}^{(p)} \rightarrow s_{d,n}^{(p)}}(s_{d,n}^{(p)})
\]
\[
= \frac{\pi_n}{\pi_n} s_{d,n}^{(p)} + (1 - \frac{\pi_n}{\pi_n})(1 - s_{d,n}^{(p)}),
\] (50)
with
\[
\frac{\pi_n}{\pi_n} = \frac{\lambda_n^{f_{d,n}^{(p)}} \lambda_n^{f_{d,n}^{(p)}}}{(1 - \lambda_n^{f_{d,n}^{(p)}})(1 - \lambda_n^{f_{d,n}^{(p)}}) + \lambda_n^{f_{d,n}^{(p)}} \lambda_n^{f_{d,n}^{(p)}}}.
\] (51)

The message from factor node $f_{d,n}^{(p)}$ to variable node $h_{d,n}^{(p)}$ is
\[
\nu_{f_{d,n}^{(p)} \rightarrow h_{d,n}^{(p)}}(h_{d,n}^{(p)}) \propto \sum_{s_{d,n}^{(p)}} f_{d,n}^{(p)}(h_{d,n}^{(p)}, s_{d,n}^{(p)}) \cdot \nu_{s_{d,n}^{(p)} \rightarrow f_{d,n}^{(p)}}(s_{d,n}^{(p)})
\]
\[
= \frac{\pi_n}{\pi_n} \mathcal{CN}(h_{d,n}^{(p)}; 0, (\sigma_d^{(p)}))^2
\]
\[
+ (1 - \frac{\pi_n}{\pi_n}) \delta(h_{d,n}^{(p)}).
\] (52)

The posterior mean and variance can be calculated as
\[
h_{B,n}^{post(p)} = \mathbb{E}(h_{d,n}^{(p)} | h_{B,n}^{pri(p)}),
\] (53)
and
\[
v_{B,n}^{post(p)} = \frac{1}{N} \sum_{n=1}^{N} \text{Var}(h_{d,n}^{(p)} | h_{B,n}^{pri(p)}).
\] (54)

Then, the mean and variance are updated using (31) and (30). The structured Turbo-CS algorithm with Module B realized by Eqs. (43) to (54) is referred to as structured Turbo-CS with delay support (STCS-DS), summarized in Algorithm 2. Note that both STCS-FS and STCS-DS are approximate algorithms to exploit the sparsity of the massive MIMO-OFDM channel. Though, it is difficult to tell which algorithm has better performance in theory, we will show numerically in the next section that STCS-DS makes more efficient usage of the delay-domain channel sparsity and hence considerably outperforms STCS-FS. We can also use a single factor graph to represent the two modules of the proposed STCS-based schemes, e.g., Appendix shows the factor-graph representation of the channel model in the angle-frequency domain.

VI. PERFORMANCE COMPARISONS

A. Pilot Design

The Turbo-CS algorithm and its variants are designed as a low-complexity and near-optimal solution to handle orthogonal measurements, i.e., the sensing matrix is a partial orthogonal matrix. In [17], the sensing matrix is chosen as the partial DFT matrix, which works well for the Turbo-CS algorithm when the unknown variables are i.i.d.. However, as shown in Fig. 5, the partial DFT sensing matrix does not work well here, since the support of the channel exhibits a clustered structure, rather than an i.i.d. structure as in [17].

In this work, we decorrelate the sparse signal by using random permutation (RP). The corresponding sensing matrix, referred to as a partial DFT-RP sensing matrix, is given by
\[
A = SFR,
\] (55)
where $S$ is a selection matrix consisting of randomly selected and reordered rows of the $N \times N$ identity matrix, and $R$ is a random permutation matrix. Then the corresponding pilot matrix is $X = SFRF$. With such a pilot design, the algorithm only needs to store the permutation orders specified by $S$ and $R$, rather than to store the whole sensing matrix, which relieves the storage burden at user side. Moreover, the

Algorithm 2: Structured Turbo-CS Algorithm With Delay Support (STCS-DS)

**Input:** received signal $Y_f = [y_f^{(1)}, \ldots, y_f^{(P)}]$, pilot matrix $X^{(p)} \forall p$, and additive noise variance $\sigma^2$.

**Output:** channel state information $\hat{\mathbf{H}}$.

**Initialize:** $Y_f = Y_f F_r$, $A = X F H$, $h_A^{pri(p)}$, $v_A^{pri(p)} \forall p$.

**Module A:**

1. $h_A^{post(p)} = h_A^{pri(p)} + \frac{\nu_A^{pri(p)}}{\nu_A^{pri(p)} + \sigma^2} A H (y_f^{(p)} - A h_A^{pri(p)}) \forall p$
2. $v_A^{post(p)} = v_A^{pri(p)} - M \sum_{d,n}^{P} \frac{(v_A^{pri(p)})(v_A^{post(p)})}{v_A^{pri(p)} + \sigma^2} \forall p$

**Update extrinsic messages**

3. $v_B^{pri(p)} = v_B^{ext(p)} = \left(\frac{1}{v_B^{post(p)}} - \frac{1}{v_B^{ext(p)}}\right)^{-1}$, $\forall p$
4. $h_B^{pri(p)} = h_B^{ext(p)} = v_B^{ext(p)} \left(\frac{h_B^{post(p)}}{v_B^{ext(p)}} - \frac{h_B^{pri(p)}}{v_B^{ext(p)}}\right)$, $\forall p$

**Module B:**

**% Structured estimator**

5. $h_B^{post(p)}$ and $v_B^{post(p)}$, $\forall p$, are given by (53) and (54).

**% Update extrinsic messages**

6. $v_A^{pri(p)} = v_B^{ext(p)} = \left(\frac{1}{v_B^{post(p)}} - \frac{1}{v_B^{ext(p)}}\right)^{-1}$, $\forall p$
7. $h_B^{pri(p)} = h_B^{ext(p)} = v_B^{ext(p)} \left(\frac{h_B^{post(p)}}{v_B^{ext(p)}} - \frac{h_B^{pri(p)}}{v_B^{ext(p)}}\right)$, $\forall p$

Repeat Module A and Module B until convergence or the maximum iteration number is exceeded.
matrix multiplication involving $A$ can be realized by the Fast Fourier Transform (FFT) algorithm for complexity reduction.

In simulation, we consider a massive MIMO-OFDM system with $N = 256$ antennas at BS. Pilot subcarriers are uniformly allocated in the frequency band. The total number of pilot subcarriers is $32$. The realizations of the delay taps $H_d$ are generated by using the following parameter setting. The states $\{s^{(p)}_{d,n}\}$ are generated with transition probability $p_{01}^{(p)} = 1/16$ and $p_{10}^{(p)} = 1/240$. Given $\{s^{(p)}_{d,n}\}$, $H_d$ is generated by following (11). The maximum delay length $L$ is $16$. Once $H_d$ is generated, $H_f$ can be obtained from (5). The training length is $M = 103 \approx 0.4N$. In Fig. 5, Turbo-CS, STCS-FS, and STCS-DS are tested, where the normalized mean square error (NMSE) is defined as NMSE = $\|H - H_0\|_2^2 / \|H\|_2^2$. From Fig. 5, we see that all the algorithms converge when a partial DFT-RP sensing matrix is used; however, the algorithm diverges when a partial DFT sensing matrix is used. The simulation results in later subsections always use partial DFT-RP sensing matrices for pilot design. In addition, following the model in Section V-A, we always assume that a common pilot matrix is used for all subcarriers unless otherwise specified.

B. Storage and Computational Complexity

In original Turbo-CS, the sensing matrix is chosen as a partial DFT matrix, which means the matrix multiplication can be substituted by using FFT. The storage complexity for sensing matrix and computational complexity for each iteration are $O(1)$ and $O(N\log N + N)$. For STCS in this paper, we need additional storage $O(N)$ for permutation matrix and some additional calculations caused by the permutation with computational complexity $O(N)$. In addition, STCS involves $P$ measurements. Therefore, the proposed STCS-FS and STCS-DS have per-iteration complexity $O(PN\log N + PN)$. This per-iteration complexity is lower than that of AMP-NNSPL-DD in [23] (with per-iteration complexity $O(MPN + N^2 + P\log P)$). That is, STCS is more efficient in both storage and per-iteration complexity than AMP-NNSPL-DD. Later, we will further show that STCS also exhibits the fastest convergence rate among all the existing algorithms.

C. State Evolution

The performance of Turbo-CS can be characterized by simple scalar recursions called state evolution [17]–[19]. We apply a similar technique to STCS by tracking the input variance $\tau_A$ and $\tau_B$ of Module A and Module B. Specifically, the relation of $\tau_A$ and $\tau_B$ can be described by $\tau_B = f(\tau_A)$ and $\tau_A = g(\tau_B)$, where $f(\cdot)$ and $g(\cdot)$ correspond to the operations of Module A and B respectively. Then, the fixed point $\tau = g(f(\tau))$ can be used to predict the output mean square error of the STCS algorithm. In this paper, by following [19], $f(\cdot)$ and $g(\cdot)$ are given by

$$\tau_A = g(\tau_B) = \frac{1}{NP}E \left[\|D_B(H + \tau_B E) - H\|^2_F\right]$$

(56) and

$$\tau_B = f(\tau_A) = \frac{N}{M} (\tau_A + \sigma^2) - \tau_A,$$

(57)

where $D_B$ in (56) is the input output function of Module B with the input $H + \tau_B E$, and each element of matrix $E$ obeys a circularly complex Gaussian distribution with zero mean and unit variance. Note that function $D_B$ includes not only the structured estimator but also the extrinsic update step. Also note that $g(\cdot)$ in (56) does not have a simple analytical expression. This function can be numerically evaluated by simulation.

Fig. 6 illustrates the NMSE performances of Turbo-CS and the various STCS-based algorithms proposed in this paper, together with the predictions by the state evolution. In simulation, $N = 256$ and SNR = 10 dB and 30 dB. We see that all the STCS-based algorithms agree well with the state evolution. However, there is a gap for Turbo-CS between simulation and state evolution at SNR = 30 dB. The reason is that the original Turbo-CS algorithm is designed for i.i.d. unknowns, and does not work well for unknowns with clustered sparsity.

D. EM Learning

The STCS based algorithms require the prior knowledge of the channel distribution. However, the parameters of the
channel distribution are usually unknown in practice. In [16],
the expectation maximization (EM) algorithm is combined
with the approximate message passing (AMP) algorithm [15]
to learn the distribution parameters. A similar EM algorithm
can be applied to STCS. For STCS-FS, recall that
the expectation \( \lambda_f \) is taken over the output distribution
\( H_f \). More details of the EM
is taken over the output distribution
\( H_f \). Then, in each iteration, the parameters are updated by
\[
q_f^{(t+1)} = \arg \max_{q_f} \mathbb{E} \{ \| y_f \|^2 - 2 \mathbb{E} \{ H_f Y_f q_f \} | Y_f; q_f^{(t)} \},
\]
(58)
where the expection \( \mathbb{E} \) is taken over the output distribution
of \( H_f \) in the \( t \)-th EM iteration. More details of the EM
algorithm can be found, e.g., in [16]. Similarly, the EM
parameter learning scheme can also be applied to STCS-DS.
The parameters of the channel distribution under consideration. Then, in each iteration, the parameters are updated by
\[
q_d^{(p),(t+1)} = \arg \max_{q_d^{(p)}} \mathbb{E} \{ \| y_d^{(p)} \|^2 - 2 \mathbb{E} \{ H_d^{p,r}(q_d^{(p)}) \} | y_d^{(p)}; q_d^{(p),(t)} \},
\]
(59)
for all \( p \).

E. Noisy Channel Estimation

In this subsection, we compare the performance of the proposed STCS-FS and STCS-DS with various baseline
algorithms using the channel generated in Subsection A. The parameters of the channel are learned by the EM framework.
For frequency support algorithm, the parameters are initialized by
\( \lambda_f = 0.3, \sigma_f = 2N \| y_f \|^2 / M \| A(p) \|^2, \forall p \), and
\( p_{01} = 0.1 \). For delay support algorithm, the parameters are initialized by
\( \lambda_d = 0.3, \sigma_d = 2N \| y_d \|^2 / M \| A(p) \|^2, \forall p \), and
\( p_{01} = 0.1 \) and \( \gamma_d = 0.1, \forall p \). In Fig. 7, we compare the average NMSE performance of OMP [9], DSAMP [14],
L1 LASSO [9], EM-BG-AMP [16], AMP-NNSPL-FD [23],
AMP-NNSPL-DD [23], STCS-FS, and STCS-DS under a wide
range of SNR and number of measurements. A 21 \times 20 grid of
each algorithm is constructed from SNR in [−10, 30] dB and
\( M \in [0.05N, N] \). The performance is averaged over
100 independent trials at each grid point. From Fig. 7, we see that the proposed STCS based algorithms, especially
STCS-DS, can achieve a considerable gain over all baseline
algorithms under various system settings.

F. Test for More Realistic Channel Data

We compare the performance of the proposed STCS based
algorithms with various baseline algorithms under two real-
istic channel models: the spatial channel model (SCM) [28]
developed in 3GPP/3GPP2 for low frequency band (less than
6 GHz), and the millimeter-wave statistical spatial channel
model (mm-SSCM) proposed in [33] for high frequency band
(28-73 GHz). The SCM has been widely used to evaluate
TABLE III
PARAMETER SETTINGS FOR THE CHANNEL MODEL

| Parameter name | Value | Parameter name | Value |
|----------------|-------|----------------|-------|
| NumBSElements  | 256   | Subcarriers    | 512   |
| NumMSElements  | 1     | Subcarrier spacing | 15 kHz |
| CenterFrequency| 2 GHz | NumPaths       | 6     |

Fig. 8. The NMSEs of various algorithms versus the number of measurements $M$ under SCM, with $N = 256$ and SNR = 0 dB. The algorithms are tested under different scenarios. (a) Urban macro. (b) Suburban macro. (c) Urban micro. The results are averaged over 50 independent realizations.

The channel estimation performance of Massive MIMO-OFDM systems; see, e.g. [11], [12], [20], [34]. In the following, we will use simulations to verify that the proposed STCS algorithms can achieve superior performance over the state-of-the-art baseline algorithms in realistic channel models under different scenarios, which implies that the proposed probabilistic channel models are flexible and work well for realistic channels.

The parameters of SCM used in the simulations are listed in Table III. Note that different pilot matrices are used for different subcarriers in the simulations of the AMP-NNSPL-FD and AMP-NNSPL-DD algorithms [23]. In Fig. 8 and Fig. 9, the simulation results are given for SNR = 0 dB and SNR = 10 dB, respectively. From Fig. 8, we see that the proposed STCS-FS and STCS-DS significantly outperform OMP [9], DSAMP [14], L1 LASSO [9], EM-BG-AMP [16], Turbo-CS [17], AMP-NNSPL-FD, and AMP-NNSPL-DD [23] algorithms. Meanwhile, we observe that the NNSPL-based algorithms cannot work well in the low SNR regime. Fig. 9 has similar trends as those in Fig. 8 except for the AMP-NNSPL-DD algorithm. Specifically, the AMP-NNSPL-DD algorithm [23] outperforms the proposed STCS-DS algorithm in the range of $M < 50$, due to the diversity from different pilot matrices on different pilot subcarriers [14]. As the increase of $M$, the AMP-NNSPL-DD algorithm and the proposed STCS-DS algorithm almost work the same. Fig. 10 shows the NMSE performances of the STCS-based algorithms and the NNSPL-based algorithms [23] as a function of iteration number at SNR = 10 dB, with $N = 256$ and $\frac{M}{N} = 0.4$ in (a), $\frac{M}{N} = 0.6$ in (b), and $\frac{M}{N} = 0.8$ in (c). The results are averaged over 50 independent realizations.

Fig. 9. The NMSEs of various algorithms versus the number of measurements $M$ under SCM, with $N = 256$ and SNR = 10 dB. The algorithms are tested under different scenarios. (a) Urban macro. (b) Suburban macro. (c) Urban micro. The results are averaged over 50 independent realizations.

Fig. 10. The NMSEs versus the iteration number of various algorithms under urban macro scenario at SNR = 10 dB, with $N = 256$ and $\frac{M}{N} = 0.4$ in (a), $\frac{M}{N} = 0.6$ in (b), and $\frac{M}{N} = 0.8$ in (c). From Fig. 10, we see that the STCS-based algorithms converge much faster than the NNSPL-based algorithms.

TABLE IV
PARAMETER SETTINGS FOR THE MM-WAVE MODEL

| Parameter name | Value | Parameter name | Value |
|----------------|-------|----------------|-------|
| Number of TX   | 256   | Subcarriers    | 800   |
| Number of RX   | 1     | Pilot subcarriers | 40   |
| Frequency      | 28 GHz| Subcarrier spacing | 1 MHz |
| Separation Distance | 100 m | Scenario | UMa  |
that the proposed algorithms still considerably outperform the state-of-the-art baseline algorithms in the mm-SSCM. This shows the advantage and robustness of the proposed algorithms in practical massive MIMO-OFDM systems.

VII. CONCLUSIONS

In this paper, we apply the structured Turbo-CS framework to improve the estimation accuracy of the massive MIMO-OFDM channel by exploiting its sparsity structure in the angle-frequency domain and angle-delay domains. We show that the proposed STCS based algorithms can be well predicted by the state evolution even for a relatively small $N$. Finally, STCS-FS and STCS-DS are tested for realistic spatial channel models. We show that the proposed algorithms have much faster convergence speed and achieve competitive NMSE performance under a wide range of simulation settings. This demonstrates the merit of our channel estimation approach in practical massive MIMO-OFDM systems.

APPENDIX

We first present the factor graph representation of the channel model in the angle-frequency domain. Recall that $y_f = A_f h_f + w_f \in \mathbb{C}^{M \times 1}$, $1 \leq p \leq P$ in (3) and the probability model of channel in (11)-(14). To integrate the two modules of the proposed scheme into a factor graph, we first write the joint probability density of $Y_f$, $H_f$, and $s_f$ given by

$$p(Y_f, H_f, s_f) = \prod_{p=1}^{P} \prod_{n=1}^{N} p(y_f^p | h_f^p) \prod_{n=1}^{N} p(h_f^p | s_f, n) \prod_{n=2}^{N} p(s_f, n | s_f, n-1).$$

The factor graph of (60) is shown in Fig. 12. The variable nodes include $\{h_f^p\}$ and $\{s_f, n\}$. The check nodes include $p(y_f^p | h_f^p)$, $p(h_f^p | s_f, n)$, and $p(s_f, n | s_f, n-1)$. We can divide the whole factor graph into two parts, where the left part and right part correspond to Module A and Module B, respectively, as illustrated in Fig. 12. The message passing on the factor graph $\mathcal{G}_f$ is as follows:

- **Left-part message passing:** The message from node $p(y_f^p | h_f^p)$ to variable node $\{h_f^p, n\}$ is given by

$$p_{\text{post}}(h_f^p | h_f^p) = p(h_f^p | y_f^p, h_f^p) \times p_{\text{post}}(y_f^p | h_f^p) \times \mathcal{CN}(h_f^p ; h_f^p, \text{ext}(p) ; f_f, n),$$

where $C$ is a normalization factor, $p(y_f^p | h_f^p)$ and $p(h_f^p | h_f^p)$ are Gaussian messages. Hence, $p_{\text{post}}(h_f^p | h_f^p)$ is also Gaussian. Then the message $p_{\text{post}}(y_f^p | h_f^p)$ in (61) satisfies the constraint in (8), i.e.

$$p_{\text{post}}(h_f^p | h_f^p) = p(h_f^p | y_f^p, h_f^p) \times p_{\text{post}}(y_f^p | h_f^p) \times \mathcal{CN}(h_f^p ; h_f^p, \text{ext}(p) ; f_f, n),$$

where the posteriori mean and variance of $p_{\text{post}}(h_f^p)$ are given by (6) and (7), respectively. Correspondingly, the mean and variance of $p_{\text{post}}(y_f^p | h_f^p)$ in (61) can be given by (9) and (10), respectively.
• The message passing for the right-part is identical to the operation of Module B of Turbo-CS.

Similarly, we can present the factor graph representation of the channel model in the angle-delay domain. We omit the details for brevity.

REFERENCES

[1] L. Chen and X. Yuan, “Massive MIMO-OFDM channel estimation via structured turbo compressed sensing,” in Proc. IEEE Int. Conf. Commun. (ICC), May 2018, pp. 1–6.

[2] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, “An overview of massive MIMO: Benefits and challenges,” IEEE J. Sel. Topics Signal Process., vol. 8, no. 5, pp. 742–758, Oct. 2014.

[3] H. Bölcskei, “MIMO-OFDM wireless systems: Basics, perspectives, and challenges,” IEEE Wireless Commun., vol. 13, no. 4, pp. 31–37, Aug. 2006.

[4] C.-X. Wang et al., “Cellular architecture and key technologies for 5G wireless communication networks,” IEEE Commun. Mag., vol. 52, no. 2, pp. 122–130, Feb. 2014.

[5] X. Yuan, C. Fan, and Y. J. Zhang, “Fundamental limits of training-based multiuser MIMO systems,” vol. 15, 2015, arXiv:1511.08977. [Online]. Available: https://arxiv.org/abs/1511.08977

[6] B. Hassibi and B. M. Hochwald, “How much training is needed in multiple-antenna wireless links?” IEEE Trans. Inf. Theory, vol. 49, no. 4, pp. 951–963, Apr. 2003.

[7] Z. Gao, L. Dai, W. Dui, B. Shin, and Z. Wang, “Structured compressive, coded-based, spatio-temporal joint channel estimation for FDD massive MIMO,” IEEE Trans. Commun., vol. 64, no. 2, pp. 601–617, Feb. 2016.

[8] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, “Compressed channel sensing: A new approach to estimating sparse multipath channels,” Proc. IEEE, vol. 98, no. 6, pp. 1058–1076, Jun. 2010.

[9] C. R. Berger, Z. Wang, J. Huang, and S. Zhou, “Application of compressive sensing to sparse channel estimation,” IEEE Commun. Mag., vol. 48, no. 11, pp. 164–174, Nov. 2010.

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

[11] 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.

[12] X. Yao and V. K. N. Lau, “Compressed sensing with prior support quality information and application to massive MIMO channel estimation with temporal correlation,” IEEE Trans. Signal Process., vol. 63, no. 18, pp. 4914–4924, Sep. 2015.

[13] 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.

[14] Y. Sun, J. Zhou, and J. Xiao, “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.

[15] 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.

[16] J. P. Vila and P. Schniter, “Expectation-maximization Gaussian-mixture approximate message passing,” IEEE Trans. Signal Process., vol. 61, no. 19, Oct. 2013.

[17] X. Liu, 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.

[18] X. Liu, X. Yuan, and L. Ping, “On the performance of turbo signal recovery with partial DFT sensing matrices,” IEEE Trans. Signal Process., vol. 22, no. 10, pp. 1580–1584, Oct. 2013.

[19] Z. Xue, J. Ma, and X. Yuan, “Denoising-based turbo compressed sensing,” IEEE Access, vol. 5, pp. 7193–7204, 2017.

[20] 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.

[21] J. Mo, P. Schniter, and R. W. Heath, Jr., “Channel estimation in broadband millimeter wave MIMO systems with few-bit ADCs,” IEEE Trans. Signal Process., vol. 66, no. 5, pp. 1141–1154, Mar. 2018.
Xiaojun Yuan (S’04–M’09–SM’15) received the Ph.D. degree in electrical engineering from the City University of Hong Kong in 2008. From 2009 to 2011, he was a Research Fellow with the Department of Electronic Engineering, City University of Hong Kong. He was also a Visiting Scholar with the Department of Electrical Engineering, University of Hawaii at Manoa, in 2009 and 2010. From 2011 to 2014, he was a Research Assistant Professor with the Institute of Network Coding, The Chinese University of Hong Kong. From 2014 to 2017, he was an Assistant Professor with the School of Information Science and Technology, ShanghaiTech University. He is currently a Professor with the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, supported by the Thousand Youth Talents Plan in China. His research interests include statistical signal processing, machine learning, and information theory, including but not limited to multi-antenna and cooperative communications, sparse and structured signal recovery, Bayesian approximate inference, and network coding. He has published over 110 peer-reviewed research papers in the leading international journals and conferences in the related areas. He was a co-recipient of the Best Paper Award of the IEEE International Conference on Communications (ICC) 2014 and the Best Journal Paper Award of the IEEE Technical Committee on Green Communications and Computing (TCGCC) 2017. He has served on a number of technical programs for international conferences. He has been an Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS since 2017 and also an Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS since 2018.

An Liu (S’07–M’09–SM’17) 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 was 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.