Doubly-Iterative Sparsified MMSE Turbo Equalization for OTFS Modulation

Haotian Li, *Student Member, IEEE*, and Qiyue Yu, *Senior Member, IEEE*

Abstract—Currently, orthogonal time frequency space (OTFS) modulation has drawn much attention to reliable communications in high-mobility scenarios. This paper proposes a doubly-iterative sparsified minimum mean square error (DI-S-MMSE) turbo equalizer, which iteratively exchanges the extrinsic information between a soft-input-soft-input (SISO) MMSE estimator and a SISO decoder. Our proposed equalizer does not suffer from short loops and approaches the performance of the near-optimal symbol-wise maximum *a posteriori* (MAP) algorithm. To exploit the inherent sparsity of OTFS system, we resort to graph theory to investigate the sparsity pattern of the channel matrix, and propose two sparsification guidelines to reduce the complexity of calculating the matrix inverse at the MMSE estimator. Then, we apply two iterative algorithms to MMSE estimation, i.e., the Generalized Minimal Residual (GMRES) and Factorized Sparse Approximate Inverse (FSPAI) algorithms. The former is used at the initial turbo iteration, whose global convergence is proven in our equalizer, while the latter is used at the subsequent turbo iterations with the help of our proposed guidelines. Simulation results demonstrate that our equalizer can achieve a good bit-error-rate (BER) performance regardless of the fractional Doppler, and only has a linear order of complexity.

Index Terms—Orthogonal time frequency space (OTFS), minimum mean square error (MMSE), turbo equalization, graph theory, iterative algorithms for sparse linear systems.

I. INTRODUCTION

NEXT generation wireless communication systems are expected to accommodate various wireless applications in high-mobility scenarios, such as vehicle-to-vehicle (V2V), and high speed railway. In these scenarios, the performance of conventional orthogonal frequency division multiplexing (OFDM) is greatly degraded. To combat the Doppler spread, one of the appealing solutions is orthogonal time frequency space (OTFS) [1], [2], which has great potential for supporting reliable communications in high-mobility scenarios.

OTFS multiplexes the information symbols in the delay-Doppler (DD) domain instead of the time-frequency (TF) domain. Through a two-dimensional transform, OTFS spreads each symbol across the entire TF domain so that all symbols experience the same channel gain. Hence, OTFS has the potential of extracting full channel diversity with a well-designed transceiver [3], and greatly outperforms conventional OFDM systems [1], [4], [5], [6], [7]. More importantly, the DD domain wireless channel is inherently sparse and has lower variability than that in the TF domain, which facilitates channel estimation and significantly reduces the overhead [1], [8], [9].

However, these benefits of OTFS are obtained at the expense of prohibitive detection complexity. Therefore, many researchers have developed various low-complexity detectors by exploiting the sparsity of the wireless channel in the DD domain. In [10], the authors have proposed a linear minimum mean square error (LMMSE) receiver with a log-linear complexity order, which utilizes the sparsity and quasi-banded structure of the matrices. To further harvest the diversity gain, the authors of [4] have developed an iterative receiver based on message passing algorithm (MPA) with a linear complexity order, in which the interference is assumed to be Gaussian. Moreover, a hybrid maximum *a posteriori* (MAP) and parallel interference cancellation (PIC) detection (Hybrid-MAP-PIC) algorithm has been proposed in [11], whose performance approaches that of the near-optimal symbol-wise MAP algorithm with low complexity. However, due to the short loops in the factor graph that commonly exist in OTFS system, the MPA and Hybrid-MAP-PIC algorithm may fail to converge, resulting in undesirable performance degradation. To tackle this issue, a variational Bayes approach has been developed in [12], which can converge regardless of the short loops in the factor graph. Besides, the authors of [13] have investigated the design of OTFS detector based on approximate message passing (AMP), which handles short loops well with low complexity, and achieves a comparable bit error rate (BER) performance to the MMSE detection. In [14], a cross domain iterative detection of OTFS system has been proposed, which passes the extrinsic information between the time and DD domain. This detector can approach the performance of the maximum-likelihood (ML) sequence detection at the cost of a cubic complexity order.

The existing studies motivate us to achieve a desirable trade-off between the BER performance and complexity. In this paper, we propose a doubly-iterative sparsified MMSE

(DI-S-MMSE) turbo equalizer, which consists of a sparsified soft-input-soft-output (SISO) MMSE estimator and a SISO decoder. It is observed that the overall computational complexity is dominated by computing the inverse of the covariance matrix of the received sequence for MMSE estimation. To address this problem, our proposed equalizer sparsifies the covariance matrix so that its inverse can be derived with a linear complexity order. More importantly, our equalizer can approach the performance of the near-optimal symbol-wise MAP detector regardless of the fractional Doppler, and does not suffer from the short loops as in the MPA [4] and SPA [11] detectors. The major contributions of this paper are summarized as follows.

1) We propose a doubly-iterative sparsified MMSE turbo equalizer, which iteratively exchanges the extrinsic information between the MMSE estimator and the decoder. This turbo iteration is referred to as the outer iteration. On the other hand, to reduce the detection complexity, we use two iterative algorithms to perform MMSE estimation, i.e., the Generalized Minimal Residual (GMRES) and Factorized Sparse Approximate Inverse (FSPAI) algorithms, which are referred to as inner iteration.

2) To fully exploit the inherent sparsity of the DD domain channel, we resort to graph theory to analyze the sparsity pattern of the channel matrix. Based on this, we propose two guidelines to sparsify the covariance matrix. In addition, considering the randomness of the channel matrix, we define the sparsity level in the form of cumulative distribution function (CDF) to measure how sparse a matrix is. It is validated that the sparsity level of the inverse can be greatly improved by the sparsification guidelines and thus the detection complexity is significantly reduced.

3) At the initial outer iteration, computing the inverse is converted into solving the equivalent sparse linear systems via GMRES. Moreover, we prove the global convergence of GMRES in our system. At the subsequent outer iterations, we employ the proposed guidelines to sparsify the covariance matrix. Then, using the Hermitian positive definiteness of the covariance matrix, we modify FSPAI to approximately derive the Cholesky decomposition of its inverse. Simulation results show that our equalizer greatly reduces the complexity and achieves a great BER performance regardless of the fractional Doppler.

The remainder of this paper is organized as follows. In Section II, the system model of the proposed system is presented. Section III provides a brief overview of the proposed equalizer. Then, we analyze the sparsity pattern of the random sparse matrices and propose two sparsification guidelines based on graph theory in Section IV. Section V presents the iterative algorithms. The theoretical analysis is given in Section VI, and simulation results are presented in Section VII. Finally, some conclusions are drawn in Section VIII.

Notations: Scalars, vectors and matrices are denoted by a, $\mathbf{a}$ and $\mathbf{A}$ respectively. $\mathbb{B}$, $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{R}^+$ and $\mathbb{C}$ represent the set of all binary numbers, natural numbers, integers, real numbers, real positive numbers and complex numbers. $\delta(\cdot)$ is the Dirac delta function. $\lfloor a \rfloor$ denotes the the largest integer less than or equal to a. $\mathrm{Re}\{a\}$ and $\mathrm{Im}\{a\}$ mean the real and imaginary parts of a, and a^* is the conjugate of a. $\|\mathbf{a}\|$ represents the ℓ_2-norm of a vector $\mathbf{a}$. The inner product of two vectors $\mathbf{a}$ and $\mathbf{b}$ is denoted by $(\mathbf{a}, \mathbf{b})$. $\mathbf{A}(i, j)$ denotes the (i, j)-th element of $\mathbf{A}$. The transpose, conjugate, conjugate transpose and inverse of a matrix are represented by $(\cdot)^{\mathrm{T}}$, $(\cdot)^*$, $(\cdot)^{\mathrm{H}}$ and $(\cdot)^{-1}$. $\mathrm{diag}[\mathbf{a}]$ denotes a diagonal matrix whose diagonal elements are $\mathbf{a}$, and $\mathrm{vec}(\cdot)$ represents the column-wise vectorization of a matrix. $\mathcal{CN}(\mu, \sigma^2)$ denotes the circularly symmetric complex Gaussian distribution with mean μ and variance σ^2. The expectation operator is denoted by $\mathbb{E}\{\cdot\}$. The covariance matrix of two vectors is denoted by $\mathrm{Cov}\{\mathbf{x}, \mathbf{y}\} = \mathbb{E}\{\mathbf{x}\mathbf{y}^{\mathrm{H}}\} - \mathbb{E}\{\mathbf{x}\}\mathbb{E}\{\mathbf{y}^{\mathrm{H}}\}$. $\mathcal{O}(\cdot)$ is the order of asymptotic time complexity. $|\mathcal{A}|$ is the cardinality of a set $\mathcal{A}$.

II. SYSTEM MODEL

A. OTFS Transmitter

The diagram of OTFS system is shown in Fig. 1. The bitstream $\mathbf{u} \in \mathbb{B}^{K_1}$ is encoded by an encoder with a generator matrix $\mathbf{G}$ whose code rate is R_c, through which an encoded vector $\mathbf{c} = \mathbf{G}\mathbf{u} \in \mathbb{B}^{K_2}$ is obtained, where $K_1 = K_2 \cdot R_c$. Then, $\mathbf{c}$ is interleaved to $\mathbf{d} \in \mathbb{B}^{K_2}$, and mapped to the symbol sequence $\mathbf{x} \in \mathbb{C}^{K_3}$ from the symbol alphabet $\mathcal{S}$, where $K_3 = K_2/\log_2(|\mathcal{S}|)$. Then $\mathbf{x}$ is arranged into an $M \times N$ DD domain matrix $\mathbf{X}_{\mathrm{DD}}$ column by column with $K_3 = MN$, where the delay and Doppler dimensions contain M and N samples respectively.

Next, the inverse symplectic finite Fourier transform (ISFFT) performs an M-point FFT along the delay dimension and an N-point IFFT along the Doppler dimension to derive the TF domain transmit symbols matrix $\mathbf{X}_{\mathrm{TF}} \in \mathbb{C}^{M \times N}$ as [15]

$$\mathbf{X}_{\mathrm{TF}} = \mathbf{F}_M \mathbf{X}_{\mathrm{DD}} \mathbf{F}_N^{\mathrm{H}}, \tag{1}$$

where $\mathbf{F}_M$ denotes the M-point DFT matrix and $\mathbf{F}_N^{\mathrm{H}}$ denotes the N-point IDFT matrix. Here, NT and $M\Delta f$ represent the time duration and bandwidth respectively, where the subcarrier spacing is Δf (Hz) and the time interval is T (seconds).

After the ISFFT, the Heisenberg transform performs an M-point IFFT along each column of $\mathbf{X}_{\mathrm{TF}}$ and applies the pulse shaping waveform $g_{\mathrm{tx}}(t)$ of duration $[0, T]$. Then, the obtained time domain transmit matrix $\mathbf{X}_{\mathrm{T}}$ can be denoted by [15]

$$\mathbf{X}_{\mathrm{T}} = \mathbf{G}_{\mathrm{tx}}\mathbf{F}_M^{\mathrm{H}}\mathbf{X}_{\mathrm{TF}} = \mathbf{G}_{\mathrm{tx}}\mathbf{X}_{\mathrm{DD}}\mathbf{F}_N^{\mathrm{H}}, \tag{2}$$

where the diagonal elements of $\mathbf{G}_{\mathrm{tx}}$ are samples of $g_{\mathrm{tx}}(t)$ with sampling interval T/M, i.e., $\mathbf{G}_{\mathrm{tx}} = \mathrm{diag}\,[g_{\mathrm{tx}}(0), g_{\mathrm{tx}}(T/M), \ldots, g_{\mathrm{tx}}((M-1)T/M)] \in \mathbb{C}^{M \times M}$. Without loss of generality, this paper only considers the rectangular waveform, where $\mathbf{G}_{\mathrm{tx}}$ is simplified as the identity matrix $\mathbf{I}_M$. Nonetheless, some other well-designed windows in [16] can also be applied. Then, the transmit signal can be vectorized as $\mathbf{x}_{\mathrm{T}} = (\mathbf{F}_N^{\mathrm{H}} \otimes \mathbf{G}_{\mathrm{tx}})\mathbf{x} = (\mathbf{F}_N^{\mathrm{H}} \otimes \mathbf{I}_M)\mathbf{x}$, where $\mathbf{x}_{\mathrm{T}} = \mathrm{vec}(\mathbf{X}_{\mathrm{T}}) \in \mathbb{C}^{MN}$ and $\otimes$ represents the kronecker product. Such a vector of length MN is termed an OTFS
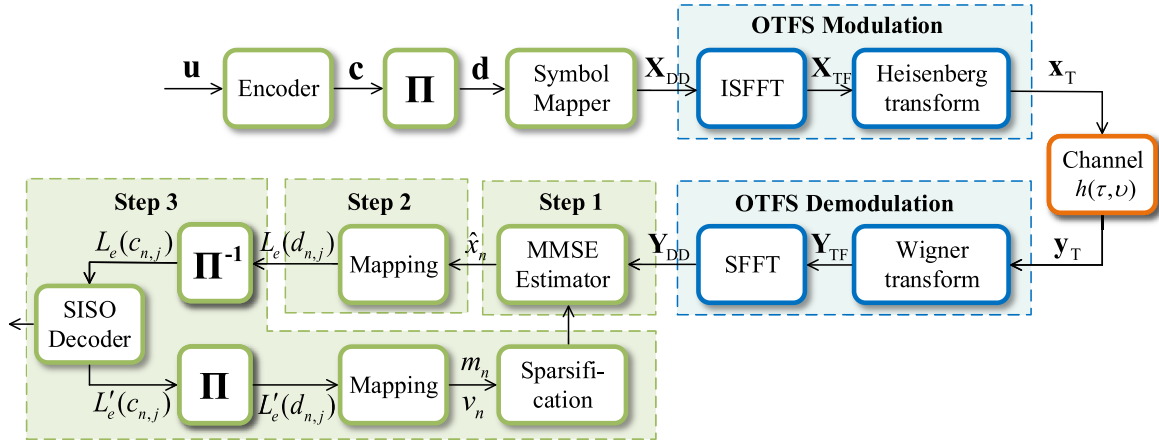
Fig. 1. A diagram of OTFS transceiver, where $\mathbf{\Pi}$ and $\mathbf{\Pi}^{-1}$ are the interleaver and deinterleaver respectively. Each iteration of the proposed equalizer consists of three steps: 1) perform MMSE estimation using the *a priori* information; 2) map the estimated symbols to the extrinsic information for the decoder; 3) after decoding, update means and variances, and sparsify the covariance matrix of the received sequence for the MMSE estimator.

frame [1]. A cyclic prefix (CP) of length L is appended to each OTFS frame, and after digital to analog conversion, the time domain signal $x_{\mathrm{T}}(t)$ is transmitted into the time-varying channel.

B. Channel Model

The channel impulse response can be characterized as $h(\tau, \nu)$ with delay τ and Doppler ν

$$h(\tau, \nu) = \sum_{p=1}^{P} h_p \delta\left(\tau - \tau_p\right) \delta\left(\nu - \nu_p\right), \qquad (3)$$

where h_p, τ_p and ν_p represent the channel gain, delay and Doppler shift of the p-th path for $p = 1, \ldots, P$, and P is the number of paths [4]. Here, the delay and Doppler shift are given by

$$\tau_p = \frac{l_p}{M\Delta f}, \quad \nu_p = \frac{k_p + \kappa_p}{NT}, \qquad (4)$$

where the delay indices l_p and Doppler shift indices k_p are integers, i.e., $k_p, l_p \in \mathbb{Z}$, and the term $\kappa_p \in [-1/2, 1/2]$ is the fractional Doppler. For conventional wide-band systems, the sampling time $1/(M\Delta f)$ is sufficiently small so that the delay taps can be approximated as integers [4]. However, due to the limit of latency, N cannot be chosen arbitrarily large. Therefore, the fractional Doppler may exist in practical systems. In the following, we assume N is sufficiently large so that the fractional Doppler can be neglected. This facilitates us to understand the sparsity characteristics of the channel matrix. Then in the simulation part, we will demonstrate that our proposed equalizer can also handle the fractional Doppler well with low complexity.

The aforementioned length of the CP should be larger than the maximum delay, i.e., $L > \tau_{\max} M\Delta f$. According to [1], the received signal $y_{\mathrm{T}}(t)$ is

$$y_{\mathrm{T}}(t) = \iint h(\tau, \nu) e^{j2\pi\nu(t-\tau)} x_{\mathrm{T}}(t - \tau) \mathrm{d}\tau \mathrm{d}\nu + z_{\mathrm{T}}(t), \quad (5)$$

where $z_{\mathrm{T}}(t)$ is the additive white Gaussian noise (AWGN) in the time domain, which is distributed as $\mathcal{CN}(0, N_0)$. After CP

is removed, $y_{\mathrm{T}}(t)$ is sampled with interval T/M. By substituting (3), the discrete received signal is obtained as [15]

$$y_{\mathrm{T}}(n) = \sum_{p=1}^{P} h_p e^{j2\pi \frac{k_p(n - l_p)}{MN}} x_{\mathrm{T}}([n - l_p]_{MN}) + z_{\mathrm{T}}(n), \quad (6)$$

where $n = 1, \ldots, MN$ and $[\cdot]_{MN}$ represents the modulo-MN operation. From (6), the time domain effective channel matrix $\mathbf{H}_{\mathrm{T}} \in \mathbb{C}^{MN \times MN}$ can be written as [15]

$$\mathbf{H}_{\mathrm{T}} = \sum_{p=1}^{P} h_p \mathbf{\Pi}^{l_p} \mathbf{\Delta}^{k_p}, \qquad (7)$$

where $\mathbf{\Pi}$ is a permutation matrix (forward cyclic shift) and represents the effect of delay

$$\mathbf{\Pi}_{MN} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 1 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}_{MN \times MN}, \qquad (8)$$

and $\mathbf{\Delta}$ models the Doppler effect and introduces an extra frequency shift, which is an $MN \times MN$ diagonal matrix as $\mathbf{\Delta} = \mathrm{diag}[\omega^0, \omega^1, \cdots, \omega^{MN-1}]$ with $\omega = e^{\frac{j2\pi}{MN}}$. In (7), each path incurs an l_p-step cyclic time shift, represented by $\mathbf{\Pi}^{l_p}$, along with a frequency shift k_p of the transmit signal $\mathbf{x}_{\mathrm{T}}$, represented by $\mathbf{\Delta}^{k_p}$. In this way, the received signal $\mathbf{y}_{\mathrm{T}} \in \mathbb{C}^{MN}$ of (6) is given by

$$\mathbf{y}_{\mathrm{T}} = \mathbf{H}_{\mathrm{T}} \mathbf{x}_{\mathrm{T}} + \mathbf{z}_{\mathrm{T}}, \qquad (9)$$

where $\mathbf{z}_{\mathrm{T}} \in \mathbb{C}^{MN}$ is the AWGN vector.

C. OTFS Receiver

At the receiver, $\mathbf{y}_{\mathrm{T}}$ is rearranged into an $M \times N$ matrix $\mathbf{Y}_{\mathrm{T}}$ column by column. Similarly, we consider rectangular received pulse shaping waveform. Then, an M-point FFT operation is applied to each column of $\mathbf{Y}_{\mathrm{T}}$ to obtain the TF domain received signal $\mathbf{Y}_{\mathrm{TF}} \in \mathbb{C}^{M \times N}$ as

$$\mathbf{Y}_{\mathrm{TF}} = \mathbf{F}_M \mathbf{G}_{\mathrm{rx}} \mathbf{Y}_{\mathrm{T}} = \mathbf{F}_M \mathbf{I}_M \mathbf{Y}_{\mathrm{T}}. \qquad (10)$$

Afterwards, the SFFT is performed, which consists of an M-point IFFT of columns and an N-point FFT of rows. Thus, by substituting (10), the DD domain received matrix $\mathbf{Y}_{\mathrm{DD}}$ is yielded as $\mathbf{Y}_{\mathrm{DD}} = \mathbf{F}_M^{\mathrm{H}}\mathbf{Y}_{\mathrm{TF}}\mathbf{F}_N = \mathbf{I}_M\mathbf{Y}_{\mathrm{T}}\mathbf{F}_N$, whose vectorized form is $\mathbf{y} = \mathrm{vec}(\mathbf{Y}_{\mathrm{DD}}) = (\mathbf{F}_N\otimes\mathbf{I}_M)\mathbf{y}_{\mathrm{T}}$. Then, the transmitted DD domain vector $\mathbf{x}$ and the received DD domain vector $\mathbf{y}$ are related as

$$\begin{aligned}\mathbf{y} &= (\mathbf{F}_N\otimes\mathbf{I}_M)\,\mathbf{H}_{\mathrm{T}}\left(\mathbf{F}_N^{\mathrm{H}}\otimes\mathbf{I}_M\right)\mathbf{x} + (\mathbf{F}_N\otimes\mathbf{I}_M)\,\mathbf{z}_{\mathrm{T}}\\&= \mathbf{H}_{\mathrm{DD}}\mathbf{x} + \mathbf{z},\end{aligned}\tag{11}$$

where $\mathbf{z}$ is the DD domain noise vector, and the effective DD domain channel matrix is

$$\mathbf{H}_{\mathrm{DD}} = (\mathbf{F}_N\otimes\mathbf{I}_M)\mathbf{H}_{\mathrm{T}}(\mathbf{F}_N^{\mathrm{H}}\otimes\mathbf{I}_M) = \sum_{p=1}^{P} h_p\mathbf{H}_{\mathrm{DD}}^{(p)},\tag{12}$$

where the p-th component $\mathbf{H}_{\mathrm{DD}}^{(p)} \in \mathbb{C}^{MN\times MN}$ is caused by the p-th path. From [15], $\mathbf{H}_{\mathrm{DD}}^{(p)}$ has been derived in (13) as shown at the bottom of the next page, where $0 \leq a, b \leq MN - 1$, $k_a = \lfloor a/M \rfloor$ and $l_a = a - k_a M$. Apparently, $\mathbf{H}_{\mathrm{DD}}^{(p)}$ contains only one nonzero element in each row and column, and thus $\mathbf{H}_{\mathrm{DD}}$ is a sparse matrix with P nonzero elements in each row and column. In what follows, the received sequence $\mathbf{y}$ is fed into the turbo equalizer to recover the information bits. In practice, the size of an OTFS frame, i.e., MN is usually very large. Therefore, a low-complexity equalizer is necessary.

III. DOUBLY-ITERATIVE SPARSIFIED MMSE TURBO EQUALIZER

In this section, we propose a doubly-iterative sparsified MMSE (DI-S-MMSE) turbo equalizer, which is composed of a SISO MMSE estimator and a SISO decoder. The proposed equalizer has a doubly iterative structure, i.e., the outer and inner iteration. The outer iteration is operated by passing soft information between the MMSE estimator and the decoder iteratively; the inner iteration is performed inside the MMSE estimator to reduce the computational cost by means of two iterative algorithms, i.e., the GMRES algorithm at the initial outer iteration and the FSPAI algorithm at the subsequent outer iterations. This section mainly focuses on the outer iteration, and the following sections will investigate the sparsification methods and the iterative algorithms.

A. An Overview of the Outer Iteration

As shown in Fig. 1, the outer iteration of our proposed equalizer consists of three steps: 1) perform MMSE estimation using the *a priori* information; 2) calculate the extrinsic information for the decoder; 3) after decoding, update means and variances, and sparsify the covariance matrix for the MMSE estimator. In this paper, we use quadrature phase shift keying (QPSK) with alphabet $\mathcal{S} = \{s_1 = (1+i)/\sqrt{2}, s_2 = (1-i)/\sqrt{2}, s_3 = (-1+i)/\sqrt{2}, s_4 = (-1-i)/\sqrt{2}\}$. Here, s_k for $k = 1, \ldots, 4$ represents the k-th constellation, whose corresponding two bits $\mathbf{s}_k = [s_{k,1}, s_{k,2}]$ are $[0,0], [0,1], [1,0]$ and $[1,1]$ respectively.

The first step is to estimate the transmitted symbols from the received sequence $\mathbf{y}$ by using the MMSE estimator. The input

of the MMSE estimator consists of the mean $m_n = \mathbb{E}\{x_n\}$ and the variance $v_n = \mathbb{E}\{|x_n - m_n|^2\}$ of each symbol x_n for $n = 1, \ldots, MN$. To minimize the mean square error (MSE) $\mathbb{E}\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\}$, the estimator derives the estimated symbol $\hat{x}_n$ from $\mathbf{y}$ as [17]

$$\hat{x}_n = m_n + \mathrm{Cov}\{x_n, \mathbf{y}\}\mathrm{Cov}\{\mathbf{y}, \mathbf{y}\}^{-1}(\mathbf{y} - \mathbb{E}\{\mathbf{y}\}).\tag{14}$$

For notational convenience, the covariance matrix of the received sequence $\mathbf{y}$ is denoted by $\mathbf{A} = \mathrm{Cov}\{\mathbf{y}, \mathbf{y}\}$, and computing its inverse largely dominates the complexity of our equalizer.

Next, the estimated symbol $\hat{x}_n$ is mapped to its corresponding log-likelihood ratio (LLR) for the decoder. To harvest the potential full channel diversity gain, the *a posteriori* probability of each bit with respect to the entire received sequence $\mathbf{y}$ is desirable, i.e., $P(d_{n,j}|\mathbf{y})$, where $\mathbf{d}_n = [d_{n,1}, d_{n,2}]$ denotes the interleaved bits corresponding to the symbol $x_n \in \mathcal{S}$ and $j = 1, 2$. However, computing $P(d_{n,j}|\mathbf{y})$ is extremely time-consuming, especially when M and N are large. Therefore, [18] inventively introduces the *a posteriori* probability with respect to the estimated symbol $P(d_{n,j}|\hat{x}_n)$ to approximate $P(d_{n,j}|\mathbf{y})$. The LLR of $P(d_{n,j}|\hat{x}_n)$ is defined as

$$\begin{aligned}L(d_{n,j}|\hat{x}_n) &\triangleq \ln\frac{P(d_{n,j}=0|\hat{x}_n)}{P(d_{n,j}=1|\hat{x}_n)}\\&= \ln\frac{\sum_{\forall\mathbf{d}_n:d_{n,j}=0} p(\hat{x}_n|\mathbf{d}_n)P(\mathbf{d}_n)}{\sum_{\forall\mathbf{d}_n:d_{n,j}=1} p(\hat{x}_n|\mathbf{d}_n)P(\mathbf{d}_n)}\\&= \ln\frac{\sum_{\forall\mathbf{d}_n:d_{n,j}=0} p(\hat{x}_n|\mathbf{d}_n)\prod_{\forall j':j'\neq j} P(d_{n,j'})}{\sum_{\forall\mathbf{d}_n:d_{n,j}=1} p(\hat{x}_n|\mathbf{d}_n)\prod_{\forall j':j'\neq j} P(d_{n,j'})}\\&\quad+\ln\frac{P(d_{n,j}=0)}{P(d_{n,j}=1)},\end{aligned}\tag{15}$$

where $j' = 1, 2$. The first term of (15) is defined as the extrinsic LLR $L_e(d_{n,j})$, and the second term is defined as the *a priori* LLR $L(d_{n,j})$, which is derived from the decoder. It is worth noting that $L_e(d_{n,j})$ and $L(d_{n,j})$ should be strictly independent.

Last, the estimator outputs the extrinsic information, which serves as the *a priori* information of the decoder. Thereafter, the decoder further mitigates the interference, and feeds the extrinsic information back to the estimator, from which new means and variances are obtained. Before we start another outer iteration, the covariance matrix $\mathbf{A}$ is sparsified to reduce the detection complexity. In the following, we will give a detailed description of these three steps.

B. Step 1: MMSE Estimation Using the a Priori Information

At the initial iteration, it is assumed that $m_n = 0$ and $v_n = 1$, while at the subsequent iterations, means and variances are provided by the decoder. By substituting (11), we have

$$\mathrm{Cov}\{\mathbf{x}, \mathbf{x}\} = \mathrm{diag}[v_1, \ldots, v_{MN}] \triangleq \mathbf{V},\tag{16a}$$

$$\mathrm{Cov}\{\mathbf{y}, \mathbf{y}\} = \mathbf{A} = \mathbf{H}_{\mathrm{DD}}\mathbf{V}\mathbf{H}_{\mathrm{DD}}^{\mathrm{H}} + N_0\mathbf{I},\tag{16b}$$

$$\mathrm{Cov}\{\mathbf{z}, \mathbf{z}\} = (\mathbf{F}_N\otimes\mathbf{I}_M)\,\mathrm{Cov}\{\mathbf{z}_{\mathrm{T}}, \mathbf{z}_{\mathrm{T}}\}\left(\mathbf{F}_N^{\mathrm{H}}\otimes\mathbf{I}_M\right) = N_0\mathbf{I},\tag{16c}$$

where (16a) is obtained due to the assumption that the transmitted symbols are independent, which approximately holds due to the interleaver; (16c) is derived from $\text{Cov}\{\mathbf{z}_{\text{T}}, \mathbf{z}_{\text{T}}\} = N_0\mathbf{I}$ and $(\mathbf{F}_N \otimes \mathbf{I}_M)(\mathbf{F}_N^{\text{H}} \otimes \mathbf{I}_M) = \mathbf{I}$ where the subscript of $\mathbf{I}_{MN}$ can be omitted without ambiguity. Moreover, we derive $\text{Cov}\{x_n, \mathbf{y}\} = \text{Cov}\{x_n, \mathbf{x}\}\mathbf{H}_{\text{DD}}^{\text{H}} = v_n\mathbf{h}_n^{\text{H}}$, where $\mathbf{h}_n$ represents the n-th column of $\mathbf{H}_{\text{DD}}$. By substituting (16b) and $\text{Cov}\{x_n, \mathbf{y}\}$ into (14), the estimated symbol $\hat{x}_n$ is derived as

$$\hat{x}_n = m_n + v_n\mathbf{h}_n^{\text{H}}\mathbf{A}^{-1}(\mathbf{y} - \mathbf{H}_{\text{DD}}\mathbf{m}), \tag{17}$$

where $\mathbf{m} = [m_1, \ldots, m_{MN}]^{\text{T}}$ denotes the vector of means.

Note that the derivation of $\hat{x}_n$ relies on the *a prior* knowledge $L(d_{n,j})$, so that the resultant $L_e(d_{n,j})$ derived from $\hat{x}_n$ is not independent of $L(d_{n,j})$. Thus, (17) is modified as [18]

$$\hat{x}_n = 0 + 1 \cdot \mathbf{h}_n^{\text{H}}\left(\mathbf{A} + (1 - v_n)\mathbf{h}_n\mathbf{h}_n^{\text{H}}\right)^{-1}(\mathbf{y} \\ -\mathbf{H}_{\text{DD}}\mathbf{m} + m_n\mathbf{h}_n), \tag{18}$$

where $\hat{x}_n$ excludes its own *a priori* knowledge by setting $m_n = 0$ and $v_n = 1$. To avoid computing the inverse in (18) repeatedly for each $\hat{x}_n$, the Woodbury matrix identity [19] is utilized as

$$\mathbf{h}_n^{\text{H}}\left(\mathbf{A} + (1 - v_n)\mathbf{h}_n\mathbf{h}_n^{\text{H}}\right)^{-1} \\ = \mathbf{h}_n^{\text{H}}\mathbf{A}^{-1} - \mathbf{h}_n^{\text{H}}\mathbf{A}^{-1}\mathbf{h}_n\left[(1 - v_n)^{-1} + \mathbf{h}_n^{\text{H}}\mathbf{A}^{-1}\mathbf{h}_n\right]^{-1} \\ \mathbf{h}_n^{\text{H}}\mathbf{A}^{-1} = [1 + (1 - v_n)\xi_n]^{-1}\mathbf{h}_n^{\text{H}}\mathbf{A}^{-1}, \tag{19}$$

where $\xi_n = \mathbf{h}_n^{\text{H}}\mathbf{A}^{-1}\mathbf{h}_n$. Then, by substituting (19) into (18), the estimated symbol is

$$\hat{x}_n = \frac{\mathbf{h}_n^{\text{H}}\mathbf{A}^{-1}(\mathbf{y} - \mathbf{H}_{\text{DD}}\mathbf{m}) + m_n\xi_n}{1 + (1 - v_n)\xi_n}. \tag{20}$$

From (20), we observe that $\mathbf{A}^{-1}$ is used twice, i.e., $\mathbf{h}_n^{\text{H}}\mathbf{A}^{-1}(\mathbf{y} - \mathbf{H}_{\text{DD}}\mathbf{m})$ and ξ_n, which dominates the complexity of MMSE estimation.

C. Step 2: Calculate the Extrinsic Information for the Decoder

To calculate $L_e(d_{n,j})$ in (15), $p(\hat{x}_n|\mathbf{d}_n = \mathbf{s}_k)$ can be assumed to be Gaussian distribution as $p(\hat{x}_n|\mathbf{d}_n = \mathbf{s}_k) = p(\hat{x}_n|x_n = s_k) = 1/(\pi\sigma_{n,k}^2)\exp(-|\hat{x}_n - \mu_{n,k}|^2/\sigma_{n,k}^2)$. Here, $\mu_{n,k} \triangleq \mathbb{E}\{\hat{x}_n|x_n = s_k\}$ and $\sigma_{n,k}^2 \triangleq \mathbb{E}\{|\hat{x}_n - \mu_{n,k}|^2 \mid x_n = s_k\}$ are the mean and variance of the estimation error $e_{n,k} = \hat{x}_n - s_k$. According to (20), $\mu_{n,k}$ and $\sigma_{n,k}^2$ are given by

$$\mu_{n,k} = \frac{\left[\mathbf{h}_n^{\text{H}}\mathbf{A}^{-1}\left(\mathbb{E}\{\mathbf{y}|x_n = s_k\} - \mathbf{H}_{\text{DD}}\mathbf{m}\right) + m_n\xi_n\right]}{1 + (1 - v_n)\xi_n} \\ = \frac{1}{1 + (1 - v_n)\xi_n}\mathbf{h}_n^{\text{H}}\mathbf{A}^{-1}\mathbf{h}_n s_k = \frac{\xi_n s_k}{1 + (1 - v_n)\xi_n}, \tag{21}$$

$$\sigma_{n,k}^2 = \frac{\mathbf{h}_n^{\text{H}}\mathbf{A}^{-1}\text{Cov}\{\mathbf{y}, \mathbf{y}|x_n = s_k\}(\mathbf{A}^{-1})^{\text{H}}\mathbf{h}_n}{[1 + (1 - v_n)\xi_n]^2} \\ = \frac{\xi_n^* - v_n\xi_n\xi_n^*}{[1 + (1 - v_n)\xi_n]^2} \overset{(a)}{=} \frac{\xi_n(1 - v_n\xi_n)}{[1 + (1 - v_n)\xi_n]^2}, \tag{22}$$

where (a) follows from the fact that $\mathbf{A}$ is Hermitian, and thus ξ_n is a real number. By substituting (21) and (22) into the extrinsic LLR of (15), $L_e(d_{n,j})$ for $j = 1, 2$ are derived as [18]

$$L_e(d_{n,1}) = \frac{\sqrt{8}[1 + (1 - v_n)\xi_n]\text{Re}\{\hat{x}_n\}}{1 - v_n\xi_n}, \tag{23a}$$

$$L_e(d_{n,2}) = \frac{\sqrt{8}[1 + (1 - v_n)\xi_n]\text{Im}\{\hat{x}_n\}}{1 - v_n\xi_n}. \tag{23b}$$

Then, $L_e(c_{n,j})$ is deinterleaved into $L_e(c_{n,j})$, and serves as the *a priori* information of the decoder.

D. Step 3: Update Means and Variances and Sparsify the Covariance Matrix

After decoding, the decoder outputs new extrinsic information $L_e'(c_{n,j})$. Through interleaver, $L_e'(d_{n,j})$ is obtained and fed back to the estimator. Considering the definition of LLR, $P(x_n = s_k)$ can be readily derived as

$$P(x_n = s_k) = \prod_{j=1}^{2} P(d_{n,j} = s_{k,j}) = \prod_{j=1}^{2} \\ \times \frac{1}{2} \cdot \left[1 - \text{sgn}(s_{k,j} - \frac{1}{2}) \cdot \tanh(\frac{L_e'(d_{n,j})}{2})\right], \tag{24}$$

where $\text{sgn}(\cdot)$ is the sign function. From $\mathcal{S}$, new means and variances are obtained as

$$m_n = \sum_{k=1}^{4} s_k \cdot P(x_n = s_k) \\ = \frac{1}{\sqrt{2}}\left[\tanh(\frac{L_e'(d_{n,1})}{2}) + i \cdot \tanh(\frac{L_e'(d_{n,2})}{2})\right],$$

$$v_n = \sum_{k=1}^{4} |x_n - m_n|^2 P(x_n = s_k) = 1 - |m_n|^2.$$

At the end of an outer iteration, we expect to sparsify $\mathbf{A}$ for ease of computing $\mathbf{A}^{-1}$. To this end, we resort to graph theory to propose some guidelines as shown in the next section.

IV. SPARSIFICATION OF THE COVARIANCE MATRIX BASED ON GRAPH THEORY

As mentioned before, high complexity of the proposed equalizer is mainly caused by computing $\mathbf{A}^{-1}$ in MMSE estimation, since $\mathbf{A}$ is an $MN \times MN$ random matrix, and MN is typically large. For example, if we have $M = 64$ and $N = 32$, directly calculating $\mathbf{A}^{-1}$ takes roughly

$$\mathbf{H}_{\text{DD}}^{(p)}(a, b) = \begin{cases} \omega^{k_p[l_a - l_p]_M - k_a M}, & \text{if } b = [l_a - l_p]_M + M[k_a - k_p]_N \text{ and } l_a < l_p \\ \omega^{k_p[l_a - l_p]_M}, & \text{if } b = [l_a - l_p]_M + M[k_a - k_p]_N \text{ and } l_a \geq l_p \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

$(MN)^3 = 2^{33}$ complex multiplications. According to (16b), $\mathbf{A}$ is a function of the random sparse matrix $\mathbf{H}_{DD}$ so that $\mathbf{A}$ is also a random sparse matrix. Unfortunately, the sparsity of $\mathbf{A}$ cannot guarantee that $\mathbf{A}^{-1}$ is also sparse. On the contrary, the inverse $\mathbf{A}^{-1}$ is dense in most cases. Hence, in this section, we resort to graph theory to provide some insight into the sparsity pattern of $\mathbf{A}$, and propose some guidelines to sparsify $\mathbf{A}$ so that $\mathbf{A}^{-1}$ can be approximated as a sparse matrix and thus the detection complexity can be greatly reduced.

A. Sparsification Guidelines Based on Graph Theory

Without loss of generality, we first introduce an arbitrary random sparse matrix $\mathbf{S}$, and present two important definitions to consider its sparsity and randomness respectively. Then using graph theory, we propose two sparsification guidelines to sparsify $\mathbf{S}$. To begin with, we define the sparsity pattern of $\mathbf{S}$ to describe the locations of nonzero elements.

Definition 1: (Sparsity pattern) For an arbitrary random sparse matrix $\mathbf{S} \in \mathbb{C}^{N_S \times N_S}$, its sparsity pattern is denoted by a Boolean matrix $\mathbf{\Upsilon_S} \in \mathbb{B}^{N_S \times N_S}$ to extract the locations of nonzero elements, where $\mathbf{\Upsilon_S}(i,j) = 1$ if and only if $\mathbf{S}(i,j) \neq 0$.

In our system, we only focus on symmetric sparsity pattern. Therefore, we assume $\mathbf{\Upsilon_S}$ is symmetric in the following. Here, $\mathbf{\Upsilon_S}$ can be easily illustrated by its graph, denoted by $\mathcal{G}(\mathcal{N}, \mathcal{E})$, which consists of a set of nodes $\mathcal{N}$ and a set of edges $\mathcal{E}$ between nodes [20]. In this graph, the node set $\mathcal{N}$ contains N_S nodes that represent the corresponding row or column indices of $\mathbf{S}$; the edge set $\mathcal{E}$ includes all the undirected lines connecting from node i to node j if $\mathbf{S}(i,j)$ and $\mathbf{S}(j,i)$ are nonzero. Besides, the number of edges connected to a node is defined as the degree, denoted by D.

On the other hand, we need to take the randomness of $\mathbf{S}$ into consideration. Since $\mathbf{S}$ is a random sparse matrix, degree of its nodes can be denoted by a discrete random variable φ_S, where $0 \leq \varphi_S \leq N_S - 1$. Based on this, the sparsity level of $\mathbf{S}$ is statistically measured as follows.

Definition 2: (Sparsity level) For an arbitrary random sparse matrix $\mathbf{S}$, let $F_{\varphi_S}(D) = P(\varphi_S \leq D)$ denote the CDF of φ_S, where D denotes a specific value of degrees and $D = 0, 1, \ldots, N_S - 1$. This CDF $F_{\varphi_S}(D)$ is defined as the sparsity level of $\mathbf{S}$.

With the help of the above definition, we can easily compare the sparsity level of two random sparse matrices. For two random sparse matrices $\mathbf{S}_1$ and $\mathbf{S}_2$, let φ_{S_1} and φ_{S_2} denote the node degrees of $\mathbf{S}_1$ and $\mathbf{S}_2$ respectively. If their sparsity levels satisfy $F_{\varphi_{S_1}}(D) > F_{\varphi_{S_2}}(D)$ for any D, more nodes of $\mathbf{S}_1$ have small degrees than those of $\mathbf{S}_2$. Therefore, $\mathbf{S}_1$ is sparser than $\mathbf{S}_2$. As will be shown in Section VI-B, the average complexity of our proposed equalizer is largely dictated by the sparsity level of $\mathbf{A}^{-1}$.

Next, based on graph theory, we propose two guidelines to sparsify $\mathbf{S}$ from the edge and node aspects respectively so that the complexity of deriving $\mathbf{S}^{-1}$ can be greatly reduced.

1) **Guideline 1** (Sparsify $\mathbf{S}$ from edge constraint): Normalize the diagonal elements of $\mathbf{S}$ through Jacobi scaling as $\widetilde{\mathbf{S}} = \mathbf{J}^{-\frac{1}{2}} \mathbf{S} \mathbf{J}^{-\frac{1}{2}}$, where $\mathbf{J}$ is a diagonal matrix and

$\mathbf{J}(i,i) = |\mathbf{S}(i,i)|$ for $i = 1, \ldots, N_S$ [21]. Then, if an edge satisfies $|\widetilde{\mathbf{S}}(i,j)| \leq \epsilon_A$, then this edge is neglected and $\mathbf{S}(i,j)$ is set to be zero. Here, ϵ_A is an appropriate threshold, and it is preferable to let $\epsilon_A \ll 1$ in case of significant performance degradation;

2) **Guideline 2** (Sparsify $\mathbf{S}$ from node constraint): If the degree of a node is smaller than a threshold ϵ_D, i.e., $D \leq \epsilon_D$, this node can be disconnected from other nodes. To avoid incurring too much imprecision, ϵ_D should be much smaller than the maximum degree.

The first guideline eliminates edges whose magnitude is negligible compared with the corresponding diagonal element, since these edges have little impact on the matrix operations of $\mathbf{S}$. Here, we use the Jacobi scaling to normalize the diagonal elements for ease of choosing ϵ_A. When the diagonal elements of $\mathbf{S}$ have larger magnitude than most of the off-diagonal ones, guideline 1 can effectively sparsify $\mathbf{S}$. Moreover, the second guideline further enhances the sparsity level by isolating the nodes with small degrees, since these nodes are hardly related with other nodes.

According to the guidelines, a smaller ϵ_A and a larger ϵ_D result in a sparser $\mathbf{S}$. Once $\mathbf{S}$ is well sparsified, the computational complexity of calculating $\mathbf{S}^{-1}$ can be greatly reduced. However, this complexity reduction is achieved at the cost of performance degradation. Hence, ϵ_A and ϵ_D will be carefully chosen from considerable simulations in Section VII to strike a balance between the complexity of calculating $\mathbf{S}^{-1}$ and performance degradation.

B. Apply the Sparsification Guidelines to Our Equalizer

Now, we apply the above definitions to $\mathbf{A}$, and leverage the sparsification guidelines to reduce the complexity of our equalizer. From (12) and (13), $\mathbf{H}_{DD}$ is the sum of $\mathbf{H}_{DD}^{(p)}$, and $\mathbf{H}_{DD}^{(p)}$ contains only one nonzero element in each row and column, whose sparsity pattern is given by

$$\mathbf{\Upsilon}_{\mathbf{H}_{DD}^{(p)}} = \mathrm{sp}(\mathbf{H}_{DD}^{(p)}) = \mathbf{\Pi}_N^{k_p} \otimes \mathbf{\Pi}_M^{l_p}, \tag{25}$$

where $\mathrm{sp}(\cdot)$ denotes the sparsity pattern of a matrix. Then, the sparsity pattern of $\mathbf{A}$ is

$$\begin{aligned}
\mathbf{\Upsilon_A} &= \mathrm{sp}(\mathbf{H}_{DD}\mathbf{V}\mathbf{H}_{DD}^{\mathrm{H}} + N_0\mathbf{I}) \\
&= \mathrm{sp}\left(\sum_{p=1}^{P} \mathbf{\Pi}_N^{k_p} \otimes \mathbf{\Pi}_M^{l_p} \cdot \sum_{p'=1}^{P} \mathbf{\Pi}_N^{-k_{p'}} \otimes \mathbf{\Pi}_M^{-l_{p'}} \right) \\
&= \mathrm{sp}\left(\sum_{p=1}^{P} \sum_{p'=1}^{P} \mathbf{\Pi}_N^{k_p - k_{p'}} \otimes \mathbf{\Pi}_M^{l_p - l_{p'}} \right) \\
&= \sum_{(\delta_k, \delta_l) \in \mathcal{D}} (\mathbf{\Pi}_N^{\delta_k} \otimes \mathbf{\Pi}_M^{\delta_l}), \tag{26}
\end{aligned}$$

where the identity matrix is dropped, since it is readily proven that the diagonal elements of $\mathbf{H}_{DD}\mathbf{V}\mathbf{H}_{DD}^{\mathrm{H}}$ are positive. Here, $\delta_k = k_p - k_{p'}, \delta_l = l_p - l_{p'}$ for $p, p' = 1, \ldots, P$, and $\mathcal{D}$ denotes the set of pairs (δ_k, δ_l). We observe that if $(\delta_k, \delta_l) \in \mathcal{D}$, then $(-\delta_k, -\delta_l) \in \mathcal{D}$, which is consistent with the fact that $\mathbf{\Upsilon_A}$ is symmetric. Besides, the degree of each node φ_A is the number

(a) Sparsity pattern of $\mathbf{H}_{\mathrm{DD}}$ (b) Sparsity pattern of $\mathbf{A}$
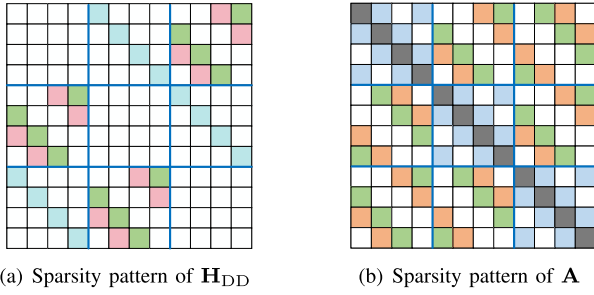
Fig. 2. An example to show the sparsity patterns of $\mathbf{H}_{\mathrm{DD}}$ and $\mathbf{A}$ with $M = 4$ and $N = 3$, where the delay and Doppler indices are $[0, 1, 2]$ and $[-1, 1, 1]$ respectively.

(a) complete graph   (b) partially split graph   (c) fully split graph

(d) $\mathbf{\Upsilon}_{\mathbf{A}^{-1}}$ of (a)   (e) $\mathbf{\Upsilon}_{\mathbf{A}^{-1}}$ of (b)   (f) $\mathbf{\Upsilon}_{\mathbf{A}^{-1}}$ of (c)
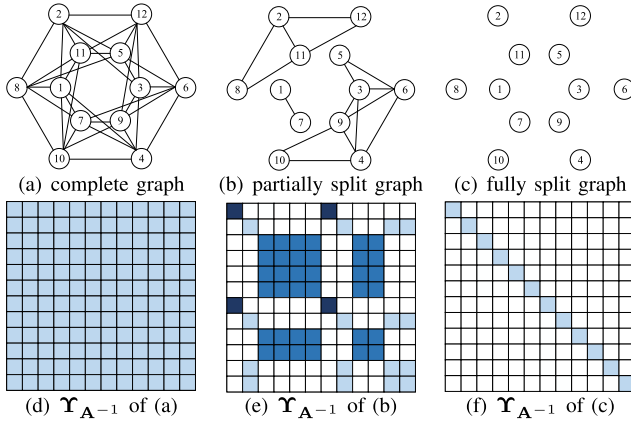
Fig. 3. Complete and split graphs of $\mathbf{A}$ with their corresponding sparsity patterns of $\mathbf{A}^{-1}$, where the blank square units represent zero elements.

of nonzero off-diagonal elements in this column. According to (26), all the node degrees satisfy $\varphi_A = |\mathcal{D}| - 1$.

An example is presented to illustrate the sparsity patterns of $\mathbf{H}_{\mathrm{DD}}$ and $\mathbf{A}$ as shown in Fig. 2. Assume there are $P = 3$ paths with the delay indices $[0, 1, 2]$ and Doppler indices $[-1, 1, 1]$, and a small OTFS frame with $M = 4$ and $N = 3$ is used for convenience of analysis. In Fig. 2 (a), different colors of the square units denote the nonzero elements caused by different paths, i.e., $\mathbf{H}_{\mathrm{DD}}^{(p)}$ for $p = 1, 2, 3$. In Fig. 2 (b), different kinds of the square units denote the nonzero elements caused by different pairs in $\mathcal{D}$, and the square units of pairs (δ_k, δ_l) and $(-\delta_k, -\delta_l)$ share the same color but differ in texture. The graph of sparse matrix $\mathbf{A}$ in Fig. 2 is shown in Fig. 3 (a). Obviously, the nodes are connected intricately where each node has degree $D = 6$. The resultant $\mathbf{A}^{-1}$ turns out to be a dense matrix whose sparsity pattern is shown in Fig. 3 (d), and each node of $\mathbf{A}^{-1}$ has degree $D = MN - 1$, i.e., $F_{\varphi_{A^{-1}}}(MN - 2) = 0$ and $F_{\varphi_{A^{-1}}}(MN - 1) = 1$. This is the worst case that needs to be avoided.

Here, we give a special case to illustrate the effect of the guidelines. As shown in Fig. 3 (b), after eliminating all the trivial edges and isolating all the nodes with small degree, the graph is split into three separate subgraphs. In this way, $\mathbf{A}^{-1}$ can be derived by computing the inverse of these subgraphs individually. In Fig. 3 (e), we use different color shades to illustrate the inverse of these subgraphs. Obviously, the sparsity level of $\mathbf{A}^{-1}$ is enhanced and thus can be derived with less complexity. As will be shown shortly, even if the

graph is not ideally split into several disconnected subgraphs, the complexity can be significantly reduced with the help of the guidelines. In an extreme case, if all the nodes are fully disconnected, $\mathbf{A}$ is perfectly reduced to a diagonal matrix whose graph is exhibited in Fig. 3 (c), and the sparsity pattern of $\mathbf{A}^{-1}$ is also diagonal as shown in Fig. 3 (f). This is the best case, where $F_{\varphi_{A^{-1}}}(0) = 1$ and $\mathbf{A}^{-1}$ can be directly obtained with complexity $\mathcal{O}(MN)$.

As illustrated in the example, $\mathbf{A}$ can be computed with much less complexity after applying these two guidelines. When E_b/N_0 is low, $\mathbf{A} = \mathbf{H}_{\mathrm{DD}}\mathbf{V}\mathbf{H}_{\mathrm{DD}}^{\mathrm{H}} + N_0\mathbf{I}$ is dominated by the second term $N_0\mathbf{I}$, so that the diagonal elements of $\mathbf{A}$ have much larger magnitude than the off-diagonal elements. Hence, after sparsifying $\mathbf{A}$ according to the guidelines, most off-diagonal elements can be eliminated. Unfortunately, at high E_b/N_0 the off-diagonal elements are likely to have the same order of magnitude as the diagonal elements. In this case, we should avoid computing $\mathbf{A}^{-1}$ directly. Nevertheless, with the help of the decoder, the *a priori* information is highly reliable at the second iteration, so that the updated variances are close to zero, which diminish the impact of the first term $\mathbf{H}_{\mathrm{DD}}\mathbf{V}\mathbf{H}_{\mathrm{DD}}^{\mathrm{H}}$ on $\mathbf{A}$. As a result, the second term $N_0\mathbf{I}$ dominates again and the guidelines can be utilized to sparsify $\mathbf{A}$. In summary, reducing the complexity of MMSE estimation can be decomposed into two subproblems. The first subproblem is to perform MMSE estimation without deriving $\mathbf{A}^{-1}$ directly at the initial iteration when the covariance matrix $\mathbf{A}$ can hardly be sparsified by the guidelines; the second one is to derive $\mathbf{A}^{-1}$ at the subsequent iterations when $\mathbf{A}$ has been sparsified according to the proposed guidelines.

V. LOW-COMPLEXITY ALGORITHMS FOR MMSE ESTIMATION

In this section, we present two low-complexity algorithms to compute $\mathbf{A}^{-1}$, where one of them is used at the initial outer iteration and the other is applied at the subsequent iterations. The former resorts to the GMRES algorithm to convert the problem of computing $\mathbf{A}^{-1}$ into solving the equivalent sparse linear systems, while the latter leverages the FSPAI algorithm to derive an approximation of $\mathbf{A}^{-1}$ after applying the proposed sparsification guidelines.

A. GMRES at the Initial Outer Iteration

At the initial iteration, we have $\mathbf{A} = \mathbf{H}_{\mathrm{DD}}\mathbf{H}_{\mathrm{DD}}^{\mathrm{H}} + N_0\mathbf{I}$ due to the lack of *a priori* information. In this case, the off-diagonal elements of $\mathbf{A}$ has a comparable order to the diagonal elements, so that the sparsity level of $\mathbf{A}^{-1}$ can only be slightly enhanced. Hence, directly deriving $\mathbf{A}^{-1}$ is not a good option. As described in Section III, $\mathbf{A}^{-1}$ is needed in two places when MMSE estimation is performed, namely $\mathbf{A}^{-1}(\mathbf{y} - \mathbf{H}_{\mathrm{DD}}\mathbf{m})$ of (20) and $\xi_n = \mathbf{h}_n^{\mathrm{H}}\mathbf{A}^{-1}\mathbf{h}_n$ of (20) and (23a), which are discussed respectively in the following.

With regard to the former, we observe that $\mathbf{A}^{-1}(\mathbf{y} - \mathbf{H}_{\mathrm{DD}}\mathbf{m})$ is a common part of all the estimated symbol $\hat{x}_n$, where $\mathbf{A}^{-1}(\mathbf{y} - \mathbf{H}_{\mathrm{DD}}\mathbf{m}) = \mathbf{A}^{-1}\mathbf{y}$ at the initial iteration since $\mathbf{m}$ is of all zeros. Therefore, we can solve only one equivalent sparse linear system $\mathbf{A}\mathbf{f}_1 = \mathbf{y}$ for all $\hat{x}_n$'s and avoid directly computing $\mathbf{A}^{-1}$, where $\mathbf{f}_1$ is the vector of unknowns.
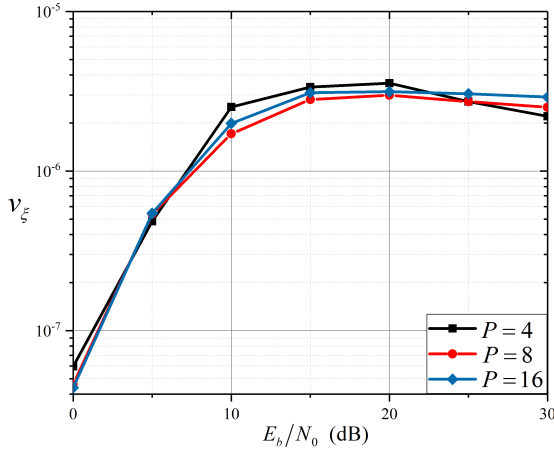
Fig. 4. The variance of ξ_n versus E_b/N_0, where $M = 64$, $N = 32$ and $P = 4, 8, 16$. At each E_b/N_0 we average over 1000 channel realizations.

As for computing ξ_n, all ξ_n's for $n = 1, \ldots, MN$ are almost the same for a given $\mathbf{H}_{DD}$ as shown in Fig. 4, where $v_\xi = \mathbb{E}\{|\xi_n - \frac{1}{MN}\sum_{n=1}^{MN}\xi_n|^2\}$ is the variance of ξ_n. It is found that v_ξ converges to zero when E_b/N_0 is extremely low or high. According to $\xi_n = \mathbf{h}_n^H \mathbf{A}^{-1} \mathbf{h}_n$, ξ_n is actually the diagonal element of $\mathbf{B} = \mathbf{H}_{DD}^H (\mathbf{H}_{DD}\mathbf{H}_{DD}^H + N_0\mathbf{I})^{-1}\mathbf{H}_{DD}$. If E_b/N_0 is low enough, $\mathbf{B}$ can be approximated as $\mathbf{B} \approx N_0^{-1}\mathbf{H}_{DD}^H\mathbf{H}_{DD}$ whose diagonal elements are all equal to $N_0^{-1}\sum_{p=1}^{P}|h_p|^2$ and $v_\xi \approx 0$. If E_b/N_0 is sufficiently high, we have $\mathbf{B} \approx \mathbf{H}_{DD}^H(\mathbf{H}_{DD}\mathbf{H}_{DD}^H)^{-1}\mathbf{H}_{DD} = \mathbf{I}$ and thus $v_\xi \approx 0$. Moreover, as observed in Fig. 4, even the maximum value of v_ξ is smaller than 10^{-5}. Hence, it is reasonable to assume that ξ_n's are the same for $n = 1, \ldots, MN$. Then, deriving ξ_n's can be converted to solving another sparse linear system $\mathbf{A}\mathbf{f}_2 = \mathbf{h}_n$ for an arbitrary n, where $\mathbf{f}_2 = \mathbf{A}^{-1}\mathbf{h}_n$ is the vector of unknowns. Thereafter, little computation is required to obtain ξ_n through $\xi_n = \mathbf{h}_n^H \mathbf{f}_2$, since $\mathbf{h}_n$ is sparse with only P nonzero elements. In this way, instead of directly calculating $\mathbf{A}^{-1}$, we can solve two equivalent sparse linear systems, i.e., $\mathbf{A}\mathbf{f}_1 = \mathbf{y}$ and $\mathbf{A}\mathbf{f}_2 = \mathbf{h}_n$ at the initial iteration.

We use an iterative algorithm called GMRES to solve the sparse linear systems, since GMRES has no requirements for the coefficient matrix $\mathbf{A}$ of a linear system $\mathbf{A}\mathbf{f} = \mathbf{b}$. Besides, GMRES takes less computation than some other candidate iterative algorithms, such as the Generalized Conjugate Residual (GCR) algorithm [22]. More importantly, the global convergence of GMRES applied in our system can be guaranteed with limited iterations (as will be presented in Section VI-A). Note that if MMSE estimation is only performed once without subsequent outer iterations, the iterative equalizer is reduced to a conventional LMMSE equalizer. Hence, GMRES can be readily extended to any other systems with LMMSE equalizer. Next, we will briefly introduce how to apply GMRES at the initial iteration.

For a linear system $\mathbf{A}\mathbf{f} = \mathbf{b}$, GMRES uses an arbitrary initial guess $\mathbf{f}_0$, e.g., a zero vector, to extract an approximate solution $\mathbf{f}_j = \mathbf{f}_0 + \Delta\mathbf{f}_j$ iteratively, where $\Delta\mathbf{f}_j$ is obtained from a subspace $\mathcal{K}_j \in \mathbb{C}^j$ to minimize the current residual

norm $\|\mathbf{r}_j\| = \|\mathbf{b} - \mathbf{A}\mathbf{f}_j\|$ [22]. Here, $\mathcal{K}_j$ is the j-th Krylov subspace $\mathcal{K}_j = span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{j-1}\mathbf{r}_0\}$ with the initial residual $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{f}_0$. For convenience, we set the norm $\|\mathbf{r}_0\| = \rho$. Then the least-squares (LS) problem can be written as

$$\min_{\Delta\mathbf{f}_j \in \mathcal{K}_j} \|\mathbf{b} - \mathbf{A}(\mathbf{f}_0 + \Delta\mathbf{f}_j)\| = \min_{\Delta\mathbf{f}_j \in \mathcal{K}_j} \|\mathbf{r}_0 - \mathbf{A}\Delta\mathbf{f}_j\|. \quad (27)$$

When the residual norm is smaller than a drop tolerance ϵ_g, the algorithm stops. Otherwise, the subspace needs to be expanded to $\mathcal{K}_{j+1}$, from which the approximate solution is updated to $\mathbf{f}_{j+1}$ to derive a more precise solution. This update process is an inner iteration.

However, with the expansion of the subspace, the computational cost increases exponentially. Hence, we let j_m be the maximum dimension of the Krylov subspace. After the j_m-th inner iteration, if the drop tolerance ϵ_g is still not reached, then compute the current approximate solution $\mathbf{f}_{j_m}$ and restart GMRES with a new initial guess $\mathbf{f}_0' = \mathbf{f}_{j_m}$. These j_m inner iterations is called a restart cycle, and the restarting step can control the overall complexity of GMRES.

Considering that the LS problem of (27) dominates the complexity, we expect to reduce (27) into a smaller problem [23]. We first use the modified Gram-Schmidt algorithm to yield the ℓ_2-orthonormal basis $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_j\}$ of $\mathcal{K}_j$ as summarized in lines 4-8 of Algorithm 1 [24], where $\mathbf{p}_i \in \mathbb{C}^{MN}$ for $i = 1, \ldots, j$. From the modified Gram-Schmidt algorithm, $\mathbf{A}\mathbf{p}_j$ is rewritten as

$$\mathbf{A}\mathbf{p}_j = \sum_{i=1}^{j+1} t_{i,j}\mathbf{p}_i, \quad (28)$$

where $t_{i,j} = (\mathbf{A}\mathbf{p}_j, \mathbf{p}_i)$. Therefore, we have $\mathbf{A}\mathbf{P}_j = [\mathbf{A}\mathbf{p}_1, \ldots, \mathbf{A}\mathbf{p}_j] = \mathbf{P}_{j+1}\mathbf{T}_j$ where $\mathbf{T}_j$ is an upper Hessenberg matrix

$$\mathbf{T}_j = \begin{bmatrix} t_{1,1} & \cdots & t_{1,j} \\ t_{2,1} & \cdots & t_{2,j} \\ & \ddots & \vdots \\ & & t_{j+1,j} \end{bmatrix}_{(j+1)\times j}.$$

Besides, since $\Delta\mathbf{f}_j$ is derived from $\mathcal{K}_j$, $\Delta\mathbf{f}_j$ can be denoted by a linear combination of the orthonormal basis as $\Delta\mathbf{f}_j = \mathbf{P}_j\mathbf{k}_j$ where $\mathbf{P}_j = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_j]$ and $\mathbf{k}_j \in \mathbb{C}^j$. By substituting $\Delta\mathbf{f}_j = \mathbf{P}_j\mathbf{k}_j$ into (27), the LS problem reduces to

$$\min_{\mathbf{k}_j} \|\rho\mathbf{p}_1 - \mathbf{A}\mathbf{P}_j\mathbf{k}_j\| = \min_{\mathbf{k}_j} \left\|\rho\mathbf{P}_{j+1}\mathbf{e}_1^{(j)} - \mathbf{P}_{j+1}\mathbf{T}_j\mathbf{k}_j\right\|$$
$$= \min_{\mathbf{k}_j} \left\|\rho\mathbf{e}_1^{(j)} - \mathbf{T}_j\mathbf{k}_j\right\|, \quad (29)$$

where $\mathbf{e}_i^{(j)}$ is the i-th column of the identity matrix $\mathbf{I}_{j+1}$. The computation of solving (29) is trivial since $\mathbf{T}_j$ is a $j \times (j+1)$ matrix, where j has a maximum value j_m and is generally much smaller than MN.

In GMRES, whether to start the next iteration of GMRES is determined by the residual norm from (29). To avoid solving (29) repeatedly, we can solve the LS problem of (29) by transforming $\mathbf{T}_j$ into an upper triangular matrix $\mathbf{U}_j$ column

by column using the rotation matrix

$$\mathbf{\Psi}_i^{(j)} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & \varsigma_i & \vartheta_i & & \\ & & -\vartheta_i & \varsigma_i & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}_{(j+1)\times(j+1)}, \quad (30)$$

where $\varsigma_i = t_{i,i}/(t_{i,i}^2 + t_{i+1,i}^2)^{1/2}$ and $\vartheta_i = t_{i+1,i}/(t_{i,i}^2 + t_{i+1,i}^2)^{1/2}$ are in the i-th and $(i+1)$-th rows for $i = 1, 2, \ldots, j$. Using $\mathbf{\Psi}_i^{(j)}$, we derive the upper triangular matrix as

$$\mathbf{U}_j = \mathbf{\Psi}_j^{(j)} \ldots \mathbf{\Psi}_2^{(j)} \mathbf{\Psi}_1^{(j)} \mathbf{T}_j = \mathbf{\Xi}_j \mathbf{T}_j = \begin{bmatrix} u_{1,1} & \cdots & u_{1,j} \\ & \ddots & \vdots \\ & & u_{j,j} \\ & & 0 \end{bmatrix},$$

$$(31)$$

where $\mathbf{\Xi}_j = \mathbf{\Psi}_j^{(j)} \ldots \mathbf{\Psi}_2^{(j)} \mathbf{\Psi}_1^{(j)}$. Then, the LS problem is rewritten as

$$\min_{\mathbf{k}_j} \left\| \mathbf{\Xi}_j (\rho \mathbf{e}_1^{(j)} - \mathbf{T}_j \mathbf{k}_j) \right\| = \min_{\mathbf{k}_j} \| \mathbf{w}_j - \mathbf{U}_j \mathbf{k}_j \|, \quad (32)$$

where $\mathbf{w}_j = \rho \mathbf{\Xi}_j \mathbf{e}_1^{(j)}$. Now we can easily solve the LS problem using backward substitution. Since the last row of $\mathbf{U}_j$ is of all zeros, all rows except the last one in $\mathbf{w}_j - \mathbf{U}_j \mathbf{k}_j$ can be canceled out by choosing $\mathbf{k}_j$. Thus, the current residual is readily derived from the last element of $\mathbf{w}_j$ without any additional computation, i.e., $\|\mathbf{r}_j\| = |\mathbf{w}_j(j+1)|$. In this way, the LS problem is solved only when the current restart cycle finishes or the algorithm stops.

The complete algorithm is summarized in Algorithm 1. Initially, we set $\mathbf{U}$ as a $(j_m + 1) \times j_m$ zero matrix, and transform $\mathbf{T}_{j_m}$ to $\mathbf{U}_{j_m}$ progressively in j_m inner iterations. At the j-th inner iteration ($j = 1, \ldots, j_m$), we use the modified Gram-Schmidt algorithm and obtain $\{t_{1,j}, t_{2,j}, \ldots, t_{j+1,j}\}$ as shown in lines 4-8. The new column $[t_{1,j}, t_{2,j}, \ldots, t_{j+1,j}]^{\mathrm{T}}$ is first multiplied by all the previous $\mathbf{\Psi}_i^{(j_m)}$ for $i = 1, 2, \ldots, j - 1$, since the previous rotation matrices also affect this newly produced column. Then, $\mathbf{\Psi}_j^{(j_m)}$ is used to eliminate $t_{j+1,j}$. Meanwhile, $\mathbf{w}$ is multiplied by $\mathbf{\Psi}_j^{(j_m)}$ according to (32). These steps are shown in lines 10-11. If the current residual $|\mathbf{w}_j(j+1)| > \epsilon_g$, another iteration is performed. At the end of a restart cycle, if norm $\|\mathbf{r}_{j_m}\|$ is still larger than ϵ_g, then restart the algorithm and go to line 3.

B. FSPAI at the Subsequent Outer Iterations

After the first outer iteration, we have obtained new means and variances, which serve as the *a priori* information to start the next outer iteration. However, since the variances matrix $\mathbf{V}$ is no longer an identity matrix, ξ_n's have completely different values for $n = 1, \ldots, MN$. As a result, GMRES needs to be used MN times per outer iteration to derive each ξ_n, which is not acceptable for the subsequent outer iterations. However, with the help of the decoder, most *a prior* information of the estimator is reliable enough, so that the corresponding new variances of the estimated symbols approach zero. Hence, the second term $N_0\mathbf{I}$ of $\mathbf{A}$ dominates,

---

Algorithm 1 Restarted GMRES

1 **Initialization:** $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{f}_0, \rho = \|\mathbf{r}_0\|, \mathbf{p}_1 = \mathbf{r}_0/\rho$,
$\mathbf{w} = \rho \mathbf{e}_1^{(j_m)}$, $\mathbf{T}$ and $\mathbf{U}$ are $(j_m + 1) \times j_m$ zero matrix ;

2 **for** $j = 1$ **to** j_m **do**

3 $\quad \widetilde{\mathbf{p}}_{j+1} = \mathbf{A}\mathbf{p}_j$;

4 $\quad$ **for** $i = 1$ **to** j **do**

5 $\quad\quad t_{i,j} = (\mathbf{A}\mathbf{p}_j, \mathbf{p}_i)$, $\widetilde{\mathbf{p}}_{j+1} \leftarrow \widetilde{\mathbf{p}}_{j+1} - t_{i,j}\mathbf{p}_i$.

6 $\quad$ **end**

7 $\quad t_{j+1,j} = \|\widetilde{\mathbf{p}}_{j+1}\|, \mathbf{p}_{j+1} = \widetilde{\mathbf{p}}_{j+1}/t_{j+1,j}$;

8 $\quad \mathbf{T}(1 : j + 1, j) \leftarrow [t_{1,j}, t_{2,j}, \ldots, t_{j+1,j}]^{\mathrm{T}}$;

9 $\quad$ Compute $\mathbf{\Psi}_i^{(j_m)}$, for $i = 1, 2, \ldots, j$;

10 $\quad \mathbf{U}(:, j) \leftarrow \mathbf{\Psi}_j^{(j_m)} \ldots \mathbf{\Psi}_2^{(j_m)} \mathbf{\Psi}_1^{(j_m)} \mathbf{T}(:, j)$,
$\quad \mathbf{w} \leftarrow \mathbf{\Psi}_j^{(j_m)} \mathbf{w}$;

11 $\quad$ **if** $|\mathbf{w}(j + 1)| < \epsilon_g$ **then**

12 $\quad\quad$ **Solve:** the LS problem
$\quad\quad \min_{\mathbf{k}_j} \|\mathbf{w}(1 : j + 1) - \mathbf{U}(1 : j + 1, 1 : j)\mathbf{k}_j\|$,
$\quad\quad \mathbf{f}_j = \mathbf{f}_0 + \mathbf{P}_j \mathbf{k}_j$;

13 $\quad\quad$ **Stop**

14 $\quad$ **end**

15 **end**

16 **Solve:** the LS problem $\min_{\mathbf{k}_{j_m}} \|\mathbf{w} - \mathbf{U}\mathbf{k}_{j_m}\|$,
$\mathbf{f}_{j_m} = \mathbf{f}_0 + \mathbf{P}_{j_m} \mathbf{k}_{j_m}$;

17 **Restart:** $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{f}_{j_m}, \rho = \|\mathbf{r}_0\|, \mathbf{p}_1 = \mathbf{r}_0/\rho$,
$\mathbf{w} = \rho \mathbf{e}_1^{(j_m)}$, $\mathbf{T}$ and $\mathbf{U}$ are $(j_m + 1) \times j_m$ zero matrix;

18 **Go to Line 3**.

---

and the diagonal elements have much larger magnitude than other elements. By sparsifying $\mathbf{A}$ according to the guidelines, $\mathbf{A}$ becomes much sparser, and the sparsity level of $\mathbf{A}^{-1}$ can be greatly enhanced. Thus, we can use the FSPAI algorithm to yield a sparse approximation of $\mathbf{A}^{-1}$, which is inherently parallelizable and can further save the computational time.

We observe that $\mathbf{A}$ is Hermitian positive definite (HPD), since $\mathbf{A} = \mathbf{H}_{\mathrm{DD}} \mathbf{V} \mathbf{H}_{\mathrm{DD}}^{\mathrm{H}} + N_0 \mathbf{I}$ is the sum of a Gram matrix and an identity matrix [25]. For an HPD matrix $\mathbf{A} = \mathbf{L}_A^{\mathrm{H}} \mathbf{L}_A$ with Cholesky factor $\mathbf{L}_A$, the inverse $\mathbf{A}^{-1}$ can be obtained by deriving a sparse approximate Cholesky factor $\mathbf{L}$ such that $\mathbf{L}\mathbf{L}^{\mathrm{H}} \approx \mathbf{A}^{-1}$. Here, $\mathbf{L}$ is a lower triangular matrix and can be obtained by minimizing the Frobenius norm $\|\mathbf{L}_A \mathbf{L} - \mathbf{I}\|_F$ with respect to a prescribed sparsity pattern of $\mathbf{L}$, denoted by $\mathbf{\Upsilon}_{\mathbf{L}} = \mathcal{L}$ [26]. This is equivalent to minimizing the Kaporin condition number of $\mathbf{L}^{\mathrm{H}} \mathbf{A} \mathbf{L}$, which is defined as [27]

$$K = \frac{\mathrm{tr}(\mathbf{L}^{\mathrm{H}} \mathbf{A} \mathbf{L})}{MN \cdot \det(\mathbf{L}^{\mathrm{H}} \mathbf{A} \mathbf{L})^{\frac{1}{MN}}}, \quad (33)$$

where $\mathrm{tr}(\cdot)$ and $\det(\cdot)$ denote the trace and determinant of a matrix respectively, and K measures the quality of the sparse approximation where $K \geq 1$. Evidently, a more precise $\mathbf{L}$ makes $\mathbf{L}^{\mathrm{H}} \mathbf{A} \mathbf{L}$ closer to $\mathbf{I}$, and thus K is closer to one. By minimizing K, FSPAI can capture the sparsity pattern of $\mathbf{L}$ dynamically [28]. Since the consequent $\mathbf{L}$ is typically sparse, there is a significant reduction in complexity. As opposed to the conventional FSPAI used in symmetric positive definite

linear system, FSPAI applied in our system should be modified to match the HPD case.

Initially, a prescribed sparsity pattern $\mathbf{\Upsilon_L}$ is selected, e.g., $\mathbf{\Upsilon_L} = \mathbf{I}$ with only diagonal elements nonzero. Let $\mathbf{a}_k$ and $\mathbf{l}_k$ be the k-th column of $\mathbf{A}$ and $\mathbf{L}$ respectively. Let $\mathcal{C}_k$ denote the set of nonzero element indices of $\mathbf{l}_k$ and $\widetilde{\mathcal{C}}_k = \mathcal{C}_k \backslash \{k\}$. Then we have (34) as shown at the bottom of the next page, where $l_{kk} = \mathbf{L}(k,k) \in \mathbb{R}^+$ and $a_{kk} = \mathbf{A}(k,k) \in \mathbb{R}^+$. For each column, l_{kk} and $\mathbf{l}_k(\widetilde{\mathcal{C}}_k)$ are obtained to minimize K. To do so, we let $\partial K / \partial l_{kk} = 0$ and $\partial K / \partial \mathbf{l}_k(\widetilde{\mathcal{C}}_k) = 0$, and then

$$\mathbf{q}_k = \mathbf{A}(\widetilde{\mathcal{C}}_k, \widetilde{\mathcal{C}}_k)^{-1} \mathbf{a}_k(\widetilde{\mathcal{C}}_k), \tag{35a}$$

$$l_{kk} = \left(a_{kk} - \mathbf{a}_k(\widetilde{\mathcal{C}}_k)^{\mathrm{H}} \mathbf{q}_k\right)^{-\frac{1}{2}}, \tag{35b}$$

$$\mathbf{l}_k(\widetilde{\mathcal{C}}_k) = -l_{kk} \mathbf{q}_k. \tag{35c}$$

Since l_{kk} and $\mathbf{l}_k(\widetilde{\mathcal{C}}_k)$ are independently computed for each column, $\mathbf{L}$ can be obtained column by column in parallel. Due to the parallelism, we can only consider $\mathbf{l}_k$ in the following.

FSPAI iteratively improves the approximation by carefully choosing new index r and adding it to $\mathcal{C}_k$. This process of augmenting $\mathcal{C}_k$ is called an inner iteration of FSPAI. The improved approximation is derived as $\mathbf{l}_k' = \mathbf{l}_k + l_{rk} \mathbf{e}_r$, where $\mathbf{e}_r$ denotes the r-th column of $\mathbf{I}$ and $r > k$. To minimize the new Kaporin condition number K_r' for index r, we need to solve [21]

$$\min_{l_{rk}} K_r'$$

$$= \min_{l_{rk}} \frac{\mathrm{tr}\left((\mathbf{L} + l_{rk}\mathbf{e}_r\mathbf{e}_k^{\mathrm{H}})^{\mathrm{H}} \mathbf{A}(\mathbf{L} + l_{rk}\mathbf{e}_r\mathbf{e}_k^{\mathrm{H}})\right)}{MN \cdot \det\left((\mathbf{L}^{\mathrm{H}} + l_{rk}^*\mathbf{e}_k\mathbf{e}_r^{\mathrm{H}})\mathbf{A}(\mathbf{L} + l_{rk}\mathbf{e}_r\mathbf{e}_k^{\mathrm{H}})\right)^{\frac{1}{MN}}}$$

$$= \min_{l_{rk}} \frac{\mathrm{tr}(\mathbf{L}^{\mathrm{H}}\mathbf{A}\mathbf{L}) + 2\mathrm{Re}\{l_{rk}^*\mathbf{a}_r^{\mathrm{H}}\mathbf{l}_k\} + a_{rr}|l_{rk}|^2}{MN \cdot \det(\mathbf{A})^{\frac{1}{MN}} \left(\prod_{k=1}^{MN} l_{kk}^{\frac{2}{MN}}\right)}, \tag{36}$$

which can be derived by letting the derivative $\partial K_r' / \partial l_{rk} = 0$. Then, we have

$$l_{rk} = -\frac{\mathbf{a}_r^{\mathrm{H}}\mathbf{l}_k}{a_{rr}} = -\frac{\mathbf{a}_r(\mathcal{C}_k)^{\mathrm{H}}\mathbf{l}_k(\mathcal{C}_k)}{a_{rr}},$$

where the sparsity of $\mathbf{l}_k$ is taken into account, and thus trivial computation is entailed to derive l_{rk}. With l_{rk}, the minimum of K_r' for index r is

$$\min_{l_{rk}} K_r' = \left(1 - \frac{\eta_{rk}}{MN} K\right), \tag{37}$$

where $\eta_{rk} = |\mathbf{a}_r^{\mathrm{H}}\mathbf{l}_k|^2 / a_{rr}$ measures the reduction in Kaporin condition number by adding the new index r. Due to the sparsity of $\mathbf{A}$, only a small fraction of indices r satisfy $\mathbf{a}_r(\mathcal{C}_k)^{\mathrm{H}}\mathbf{l}_k(\mathcal{C}_k) \neq 0$, such that the approximation can be improved as shown in (37). Resorting to the graph of $\mathbf{A}$ in Section IV, we can readily find these indices that are defined as $\widehat{\mathcal{C}}_k = \{r : r > k \text{ and } \mathbf{a}_r(\mathcal{C}_k)^{\mathrm{H}}\mathbf{l}_k(\mathcal{C}_k) \neq 0\} \backslash \mathcal{C}_k$. Thereafter, we can add one or more indices from $\widehat{\mathcal{C}}_k$ with the largest η_{rk} to $\mathcal{C}_k$, and obtain augmented indices sets $\mathcal{C}_k'$ and $\widetilde{\mathcal{C}}_k' = \mathcal{C}_k' \backslash \{k\}$. Then a more precise solution $\mathbf{l}_k'$ is derived according to (35). Note that the complexity of FSPAI is dominated by computing the inverse in (35a). Hence, the cardinality of $\widetilde{\mathcal{C}}_k$ should not exceed a certain value ζ. The algorithm keeps iterating until

---

Algorithm 2 FSPAI

1 Initialization: $\mathbf{\Upsilon_L} = \mathbf{I}$, $\epsilon_f \geq 0$, $\zeta \geq 0$;
2 for $k = 1$ **to** MN **do**
3 $\mathcal{C}_k \leftarrow \{k\}$, $\widetilde{\mathcal{C}}_k \leftarrow \emptyset$, $l_{kk} \leftarrow \sqrt{a_{kk}}$;
4   **while** $|\widetilde{\mathcal{C}}_k| < \zeta$ **do**
5 $\widehat{\mathcal{C}}_k = \{r : r > k \text{ and } \mathbf{a}_r(\mathcal{C}_k)^{\mathrm{H}}\mathbf{l}_r(\mathcal{C}_k) \neq 0\} \backslash \mathcal{C}_k$;
6 $\eta_{rk} \leftarrow |\mathbf{a}_r(\mathcal{C}_k)^{\mathrm{H}}\mathbf{l}_r(\mathcal{C}_k)|^2 / a_{rr}$, for each $r \in \widehat{\mathcal{C}}_k$;
7 $\eta_{\max} = \max_{r \in \widehat{\mathcal{C}}_k} \eta_{rk}$;
8   **if** $\eta_{\max} < \epsilon_f$ **then**
9   Stop.
10   **end**
11 $\mathcal{C}_k \leftarrow \mathcal{C}_k \cup \{r : \eta_{rk} = \eta_{\max}\}$, $\widetilde{\mathcal{C}}_k \leftarrow \mathcal{C}_k \backslash \{k\}$;
12 $\mathbf{q}_k \leftarrow \mathbf{A}(\widetilde{\mathcal{C}}_k, \widetilde{\mathcal{C}}_k)^{-1} \mathbf{a}_k(\widetilde{\mathcal{C}}_k)$;
13 $l_{kk} \leftarrow (a_{kk} - \mathbf{a}_k(\widetilde{\mathcal{C}}_k)^{\mathrm{H}} \mathbf{q}_k)^{-\frac{1}{2}}$;
14 $\mathbf{l}_k(\widetilde{\mathcal{C}}_k) \leftarrow -l_{kk}\mathbf{q}_k$.
15   **end**
16 end

$|\widetilde{\mathcal{C}}_k| > \zeta$ or the maximum of η_{rk} is smaller than a well-chosen tolerance ϵ_f. The values of ζ and ϵ_f will be optimized from considerable simulations in Section VII to reach a balance between the BER performance and complexity. Especially, ζ should be carefully chosen, since it largely determines the complexity order of our equalizer.

The entire FSPAI is summarized in Algorithm 2, where the initial sparsity pattern of $\mathbf{L}$ is set to be diagonal, and for each column, only one single index with the largest ζ_{rk} is added to $\mathcal{C}_k$ during each index update. Other initial sparsity patterns and index update strategies can be readily applied with trivial modifications.

VI. THEORETICAL ANALYSIS

A. Convergence Analysis

In what follows, full GMRES (without restarting) used in the MMSE estimator is proven to globally converge, and the upper bound of its convergence rate is derived. Then we prove that for restarted GMRES, j_m can be chosen arbitrarily without losing the global convergence.

For full GMRES, considering the definition of Krylov subspace, the residual norm at the j-th inner iteration for $j \geq 1$ can be written as

$$\|\mathbf{r}_j\| = \min_{\Delta\mathbf{f}_j \in \mathcal{K}_j} \|\mathbf{r}_0 - \mathbf{A}\Delta\mathbf{f}_j\| = \min_{p_j \in \mathcal{P}_j} \|p_j(\mathbf{A})\mathbf{r}_0\|, \tag{38}$$

where $\mathcal{P}_j$ denotes the set of polynomials satisfying $p_j(\mathbf{A}) = \sum_{i=0}^{j} \alpha_i \mathbf{A}^i$ and $\alpha_0 = 1$ [29]. Since $\mathbf{A}$ is Hermitian, $\mathbf{A}$ can be diagonalized as $\mathbf{A} = \mathbf{U}_A \mathbf{\Lambda} \mathbf{U}_A^{\mathrm{H}}$ with unitary matrix $\mathbf{U}_A$ and diagonal matrix $\mathbf{\Lambda}$. As proven in [22], for a diagonalizable $\mathbf{A}$, the relative residual norm satisfies

$$\frac{\|\mathbf{r}_j\|}{\|\mathbf{r}_0\|} \leq \min_{p_j \in \mathcal{P}_j} \max_{\lambda_A \in \mathrm{eig}(\mathbf{A})} |p_j(\lambda_A)|, \tag{39}$$

where $\mathrm{eig}(\mathbf{A})$ denotes the spectrum of $\mathbf{A}$ and λ_A represents the eigenvalues of $\mathbf{A}$. Since $\mathbf{A}$ is HPD, we have $\lambda_A \in \mathbb{R}^+$

and (39) can be bounded as [30]

$$\frac{\|\mathbf{r}_j\|}{\|\mathbf{r}_0\|} \le \min_{p_j \in \mathcal{P}_j} \max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} |p_j(\lambda)| = [T_j(\varrho)]^{-1}, \quad (40)$$

where $[\lambda_{\min}, \lambda_{\max}]$ is an interval in $\mathbb{R}$ with $\lambda_{\min}$ and $\lambda_{\max}$ denoting the minimal and maximal eigenvalues of $\mathbf{A}$ respectively, and $\varrho = (\lambda_{\max} + \lambda_{\min})/(\lambda_{\max} - \lambda_{\min})$ satisfying $\varrho \ge 1$. Here, T_j is the Chebyshev polynomial of degree j of the first kind as

$$T_j(\varrho) = \frac{1}{2}\left[\left(\varrho + \sqrt{\varrho^2 - 1}\right)^j + \left(\varrho - \sqrt{\varrho^2 - 1}\right)^j\right]. \quad (41)$$

If $\lambda_{\max} = \lambda_{\min} \equiv \lambda$, i.e., $\varrho = 1$, then $\mathbf{A} = \lambda \mathbf{U}_A \mathbf{I} \mathbf{U}_A^{\mathrm{H}} = \lambda \mathbf{I}$. This happens if and only if $\mathbf{H}_{\mathrm{DD}}$ is diagonal, i.e., $P = 1$ with $\tau = 0$ and $\nu = 0$. This indicates that there is only one time-invariant path, which conflicts with the assumption of the doubly selective channel, and thus $\varrho > 1$.

Theorem 1: Full GMRES applied to MMSE estimation globally converges with a convergence rate at least $[T_j(\varrho)]^{-1}$.

Proof: Applying Jessen's inequality to the concave function x^n yields the inequality $(x_1^n + x_2^n)/2 \ge [(x_1 + x_2)/2]^n$. Then the Chebyshev polynomial in (41) satisfies

$$[T_j(\varrho)]^{-1} \le \left(\left(\varrho + \sqrt{\varrho^2 - 1} + \varrho - \sqrt{\varrho^2 - 1}\right)/2\right)^{-j}$$
$$= \varrho^{-j} < 1, \quad (42)$$

for any $j \ge 1$. Thus, according to (40), the relative residual norm decreases monotonically with j, and thus the GMRES algorithm converges at least as rapidly as $[T_j(\varrho)]^{-1}$. $\square$

From Theorem 1, we can infer the relation between E_b/N_0 and the convergence rate as follows.

Corollary 1: The convergence rate of full GMRES decreases with E_b/N_0.

Proof: For $\mathbf{A} = \mathbf{H}_{\mathrm{DD}}\mathbf{V}\mathbf{H}_{\mathrm{DD}}^{\mathrm{H}} + N_0\mathbf{I}$, we have $\lambda_A = \overline{\lambda} + N_0$, where $\overline{\lambda}$ denotes the eigenvalues of $\mathbf{H}_{\mathrm{DD}}\mathbf{V}\mathbf{H}_{\mathrm{DD}}^{\mathrm{H}}$. Then ϱ can be rewritten as

$$\varrho = \frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} = \frac{\overline{\lambda}_{\max} + \overline{\lambda}_{\min} + 2N_0}{\overline{\lambda}_{\max} - \overline{\lambda}_{\min}}. \quad (43)$$

Thus, for a specific channel realization $\mathbf{H}_{\mathrm{DD}}$, ϱ is totally governed by N_0. Besides, it can be easily proven that $T_j(\varrho)$ monotonically increases with ϱ for $\varrho > 1$. Since the energy of transmitted symbols is normalized, the increase of E_b/N_0 is equivalent to the decrease of N_0. Accordingly, ϱ and $T_j(\varrho)$ decrease. Hence, the convergence rate of full GMRES decreases with E_b/N_0. $\square$

Next, we discuss the convergence of restarted GMRES, which is slower than that of full GMRES [31]. This is due to the fact that once the algorithm is restarted, the current

approximation subspace is discarded, and the residual generated later will not be orthogonal to the previously obtained subspace. Nevertheless, the convergence of restarted GMRES used in the MMSE estimator can still be guaranteed as stated in Theorem 2.

Theorem 2: Restarted GMRES applied to MMSE estimation converges for any $j_m \ge 1$.

Proof: The proof is given in the Appendix. $\square$

B. Complexity Analysis

In this subsection, we focus on the complexity of GMRES and FSPAI while neglecting other operations that take trivial computational time due to the inherent sparsity, such as $\mathbf{LL}^{\mathrm{H}}$ and ξ_n. We can infer from (26) that the number of nonzero elements in each column of $\mathbf{A}$ is at most $\psi = 2\binom{P}{2} + 1$. This is because for $p = p'$, $(\delta_k, \delta_l) = (0,0)$ which corresponds to the diagonal elements; for $p \ne p'$, there are at most $\binom{P}{2}$ combinations of (p, p'), where each combination has two distinct pairs (δ_k, δ_l) and $(-\delta_k, -\delta_l)$ corresponding to the off-diagonal elements. Moreover, after excluding the diagonal element, we observe that the degree of node satisfies $\varphi_A \le \psi - 1 = P(P - 1)$. Since the maximum degree of $\mathbf{A}$ is $P(P - 1)$, which has an order $\mathcal{O}(P^2)$, we prefer to use P as a reference to determine some other parameters, including j_m, ζ, and ϵ_D.

According to [22], full GMRES has a complexity order $\mathcal{O}((J + |\mathcal{D}|)JMN)$, where $|\mathcal{D}|$ is the number of nonzero elements in each column of $\mathbf{A}$ and J is the number of iterations to reach the tolerance ϵ_g. Typically, J is less than $|\mathcal{D}|$, and thus the complexity order is roughly $\mathcal{O}(|\mathcal{D}|JMN)$. For restarted GMRES, the complexity order reduces to $\mathcal{O}(n_{cyc}(j_m + |\mathcal{D}|)j_m MN)$ in total, where n_{cyc} denotes the required number of restart cycles. Since generally $j_m \ll |\mathcal{D}|$, restarted GMRES has a complexity order $\mathcal{O}(n_{cyc}|\mathcal{D}|j_m MN)$.

In FSPAI, the computational complexity is dominated by computing the inverse $\mathbf{A}(\widetilde{\mathcal{C}}_k, \widetilde{\mathcal{C}}_k)^{-1}$ of (35a). Since FSPAI aims to derive an approximation of the Cholesky factor of $\mathbf{A}^{-1}$, we focus on the sparsity level of $\mathbf{L}$. According to Algorithm 2, if a column of $\mathbf{L}$ has a degree $\varphi_L = D$, the complexity order of this column is $\mathcal{O}(D^4)$. However, the practical degree D is quite problem-dependent, because the random matrix $\mathbf{A}$ may lead to strikingly different sparsity level of $\mathbf{L}$. Nevertheless, with the aid of $F_{\varphi_L}(D)$, the average complexity is derived as $\mathcal{O}(\sum_{D=1}^{\zeta} D^4 P(\varphi_L = D)MN/n_p) = \mathcal{O}(\mathbb{E}\{\varphi_L^4\}MN/n_p)$, where n_p is the number of processors available for parallel computing and ζ is the maximum node degrees of $\mathbf{L}$ as given in Algorithm 2. Furthermore, the worst case of FSPAI has a complexity order $\mathcal{O}(\zeta^4 MN/n_p)$, where

$$K = \frac{\mathrm{tr}(\mathbf{L}^{\mathrm{H}}\mathbf{A}\mathbf{L})}{MN \cdot \det(\mathbf{L}^{\mathrm{H}}\mathbf{A}\mathbf{L})^{\frac{1}{MN}}} = \frac{\sum_{k=1}^{MN} \mathbf{l}_k^{\mathrm{H}}\mathbf{A}\mathbf{l}_k}{MN \cdot [\det(\mathbf{L}^{\mathrm{H}})\det(\mathbf{A})\det(\mathbf{L})]^{\frac{1}{MN}}}$$
$$= \frac{\sum_{k=1}^{MN}\left[l_{kk}^2 a_{kk} + 2l_{kk}\mathrm{Re}\{\mathbf{l}_k(\widetilde{\mathcal{C}}_k)^{\mathrm{H}}\mathbf{a}_k(\widetilde{\mathcal{C}}_k)\} + \mathbf{l}_k(\widetilde{\mathcal{C}}_k)^{\mathrm{H}}\mathbf{A}(\widetilde{\mathcal{C}}_k, \widetilde{\mathcal{C}}_k)\mathbf{l}_k(\widetilde{\mathcal{C}}_k)\right]}{MN \cdot \det(\mathbf{A})^{\frac{1}{MN}}\left(\prod_{k=1}^{MN} l_{kk}^{\frac{2}{MN}}\right)}, \quad (34)$$
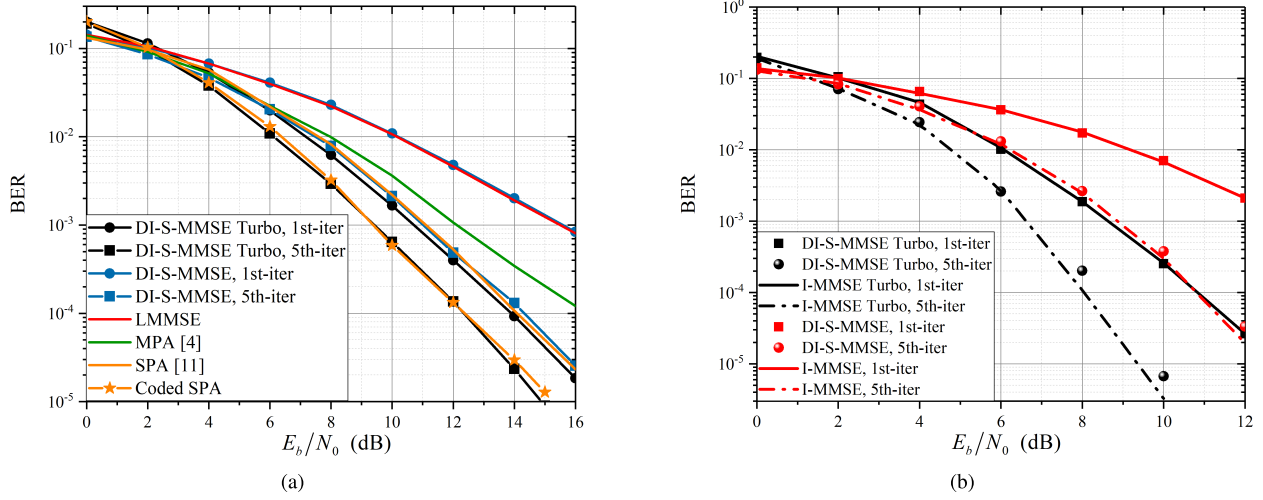
Fig. 5. The BER performance of our proposed DI-S-MMSE (turbo) equalizers, where $\epsilon_g = \epsilon_f = \epsilon_A = 10^{-3}$, $\epsilon_D = P/4$ and $\zeta = P$. (a) With $P = 4$, we present the BER performance of the LMMSE equalizer, MPA detector and (coded) SPA detector for comparison. (b) With $P = 8$, we simulate the BER performance of the I-MMSE (turbo) receivers as a lower bound, where $\mathbf{A}^{-1}$ is directly computed.

each node has degree ζ. Thus, even in the worst case, the complexity increases linearly with MN. As will be observed in the simulations, $\zeta = P$ is sufficient for our equalizer, and φ_L typically concentrates around zero, especially at high E_b/N_0. This indicates that $\mathbf{L}$ is generally very sparse, and the average complexity is far less than that of the worst case.

VII. SIMULATION RESULTS

In this section, the BER performance of OTFS with different detectors are simulated and discussed. Then, we analyze the convergence rate of full and restarted GMRES respectively. Thereafter, to evaluate the computational complexity of FSPAI, we investigate the sparsity level of $\mathbf{L}$. Lastly, we apply our proposed equalizer to the fractional Doppler case, and present the corresponding BER performance and complexity. We set $M = 64$, $N = 32$, and the length of CP is $L = 16$. An $R_c = 0.5$ convolutional code with generator (5,7) in octal is used. It is assumed that each path gain follows Rayleigh distribution with uniform power delay profile, i.e., $\mathbb{E}\{|h_p|^2\} = 1/P$. The delay and Doppler indices are uniformly chosen such that $0 \leq l \leq l_{\max}$ and $-k_{\max} \leq k \leq k_{\max}$, where $l_{\max} = 10$ and $k_{\max} = 6$. Compared with the DI-S-MMSE turbo equalizer, the DI-S-MMSE equalizer is used for uncoded OTFS systems, where the MMSE estimator treats its own extrinsic information generated previously as the *a priori* information in the next outer iteration.

In Fig. 5 (a), the BER performance of OTFS system with our proposed DI-S-MMSE (turbo) receivers is presented. For comparison, the OTFS systems with LMMSE, MPA [4] and (coded) SPA [32] detectors are also simulated. Here, the SPA is a near-optimal symbol-wise MAP algorithm [11], and considering that the complexity of SPA detector increases exponentially with P, we choose $P = 4$. At the first outer iteration, the DI-S-MMSE equalizer resorts to GMRES and achieves the same BER of the conventional LMMSE equalizer. After 5 outer iterations, the BER of the DI-S-MMSE equalizer has a roughly 4 dB gain compared with that of the LMMSE

detector at around BER $= 10^{-3}$, and roughly 1.3 dB gain compared with that of the MPA detector. More importantly, it approaches the performance of the SPA detector. Furthermore, with the help of the decoder, the DI-S-MMSE turbo equalizer achieves around 2 dB code gain compared with the DI-S-MMSE equalizer at BER $= 10^{-4}$. Similarly, the BER performance of the DI-S-MMSE turbo equalizer approaches that of coded SPA detector.

Fig. 5 (b) shows the BER performance of our proposed DI-S-MMSE (turbo) receivers with $P = 8$. To evaluate the performance loss, we present the BER performance of iterative MMSE (I-MMSE) and I-MMSE turbo receivers, where $\mathbf{A}^{-1}$ is directly computed without resorting to GMRES and FSPAI. Thus, the I-MMSE (turbo) receivers serve as lower bounds of our proposed receivers. It is found that there is no BER performance loss at the first outer iteration. On the other hand, at the subsequent outer iterations, the use of FSPAI brings a marginal BER performance degradation. For example, there is around 0.2 dB E_b/N_0 loss after 5 outer iterations. Thus, $\zeta = P$ is sufficient for the DI-S-MMSE (turbo) equalizer.

Fig. 6 shows the convergence performance of full and restarted GMRES. In Fig. 6 (a), we investigate the average relative residual norms of full GMRES applied to two sparse linear systems and their upper bounds with the number of inner iterations, where GMRES1 and GMRES2 refer to linear systems $\mathbf{A}\mathbf{f}_1 = \mathbf{y}$ and $\mathbf{A}\mathbf{f}_2 = \mathbf{h}_n$ respectively. It is observed that full GMRES can converge steadily at any E_b/N_0 in both sparse linear systems. In addition, the average relative residual norms of the upper bound have a gap compared with those of both linear systems, but they have the same slope. This clearly substantiates Theorem 1, where the upper bound provides a good approximation of the convergence rate. Furthermore, GMRES converges faster at a smaller E_b/N_0. For example, for GMRES1 and a given tolerance 10^{-3}, around 15 inner iterations are required at $E_b/N_0 = 6$ dB, while around 22 inner iterations are needed at $E_b/N_0 = 10$ dB. This verifies the observation in Corollary 1.
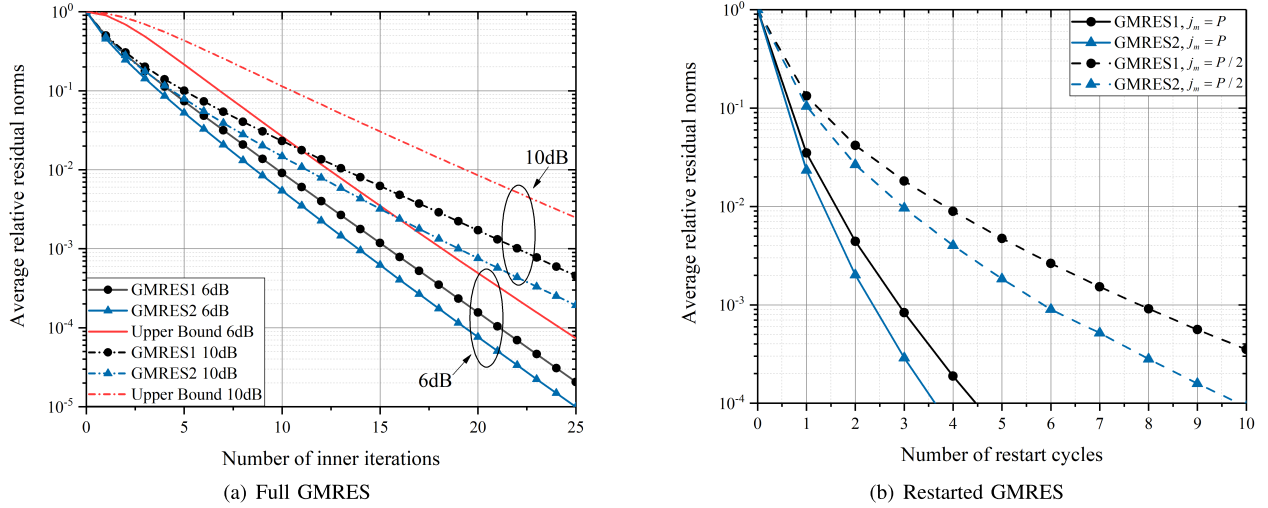
Fig. 6. Average relative residual norms of full and restarted GMRES applied to two linear systems, where $P = 8$. (a) Full GMRES cases and the corresponding upper bounds with the number of inner iterations at $E_b/N_0 = 6$ dB and 10 dB. (b) Restarted GMRES cases with the number of restart cycles, where $E_b/N_0 = 10$ dB.

Fig. 6 (b) shows the average relative residual norms of restarted GMRES applied to two sparse linear systems with the number of restart cycles at $E_b/N_0 = 10$ dB, where the number of inner iterations of a restart cycle is set as $j_m = P = 8$ and $j_m = P/2 = 4$ respectively. We observe that the convergence of the current restart cycle tends to be slower than that of the previous one. For example, for GMRES1 with $j_m = P/2$, the average relative residual norms are reduced by around 90% (form 1 to around 0.1) at the first restart cycle, but around 70% (from 0.1 to 0.03) at the second restart cycle. Hence, the reduction of the residual norms of the current restart cycle is worse than that of the previous restart cycle, which is consistent with the convergence analysis. Furthermore, a smaller j_m results in a slower convergence, and a larger n_{cyc} is required to reach the drop tolerance ϵ_g. For both linear systems with $j_m = P$, $n_{cyc} = 3$ is sufficient to reduce the residual norms to 10^{-3}, while $n_{cyc} = 8$ is sufficient for $j_m = P/2$. Nevertheless, the latter greatly reduces the complexity of each restart cycle compared with the former, since the maximum dimension of Krylov subspace is restricted. More importantly, regardless of the choice of j_m, restarted GMRES can always converge, which demonstrates Theorem 2.

Then, Fig. 7 shows the sparsity level of $\mathbf{L}$, which largely determines the complexity of FSPAI as analyzed in Section VI-B. From $F_{\varphi_L}(D)$, we can readily observe the distribution of φ_L, and derive the average complexity of FSPAI. Fig. 7 (a) illustrates the CDF $F_{\varphi_L}(D)$ of the second outer iteration at different E_b/N_0 with $P = 8$. After considerable simulations, we have optimized the parameters to achieve a low complexity order and maintain a good BER performance. The drop tolerances of GMRES and FSPAI are set to be $\epsilon_g = \epsilon_f = 10^{-3}$. The parameters of sparsification guidelines are chosen to be $\epsilon_A = 10^{-3}$ and $\epsilon_D = P/4$ respectively. The maximum node degree of FSPAI $\zeta = P = 8$ is sufficient to avoid noticeable performance loss. For the DI-S-MMSE turbo receiver at $E_b/N_0 = 6$ dB, $F_{\varphi_L}(0) \approx 0.48$ shows

that around 48% nodes have degree equal to 0; at $E_b/N_0 = 10$ dB, $F_{\varphi_L}(0) \approx 1$ indicates that most φ_L is zero and $\mathbf{L}$ is nearly a diagonal matrix. This is because with the increase of E_b/N_0, the extrinsic information generated by the decoder is more reliable. Thus, the new variances are closer to zero, and together with the sparsification guidelines applied to $\mathbf{A}$, a sparser approximation $\mathbf{L}$ is made possible. Furthermore, at the same E_b/N_0, $\mathbf{L}$ of the DI-S-MMSE turbo receiver tends to be much sparser than that of the DI-S-MMSE receiver. This is demonstrated in the figure that $F_{\varphi_L}(D)$ of the DI-S-MMSE turbo receiver is strictly larger than that of the DI-S-MMSE receiver, so that $\mathbf{L}$ is sparser and requires less computation. Hence, the simulation results verify that FSPAI can indeed reduce the complexity substantially, and the reduction is more manifest with the help of the decoder.

Fig. 7 (b) shows the sparsity level of $\mathbf{L}$ with the number of outer iterations at $E_b/N_0 = 6$ dB and 10 dB, where $P = 8$, and D is chosen to be 0 and $P/2$ respectively. Similarly, $F_{\varphi_L}(D)$ of the DI-S-MMSE turbo receiver is strictly larger than that of the DI-S-MMSE receiver at the same E_b/N_0, which indicates that our turbo receiver provides a smaller φ_L and thus a lower complexity. Besides, $F_{\varphi_L}(D)$ increases with E_b/N_0. More importantly, $F_{\varphi_L}(D)$ increases with the number of outer iterations, and the overall complexity of FSPAI is mainly dominated by the second outer iteration. This verifies that the *a priori* information is improved with the outer iteration, and the resultant new variances v_n are closer to zero, leading to a sparser $\mathbf{L}$. For example, at the third outer iteration and $E_b/N_0 = 10$ dB, both receivers have $F_{\varphi_L}(0) > 0.96$ and $F_{\varphi_L}(P/2) \approx 1$. Hence, $\mathbf{L}$ is nearly diagonal, which has 96% nodes with degree zero and almost all the nodes with degree less than $P/2$. From Fig. 5 and Fig. 7, it is verified that the turbo structure not only improves the BER performance, but also greatly reduces the complexity. Therefore, our DI-S-MMSE turbo equalizer can indeed achieve a great performance with a preferable complexity.
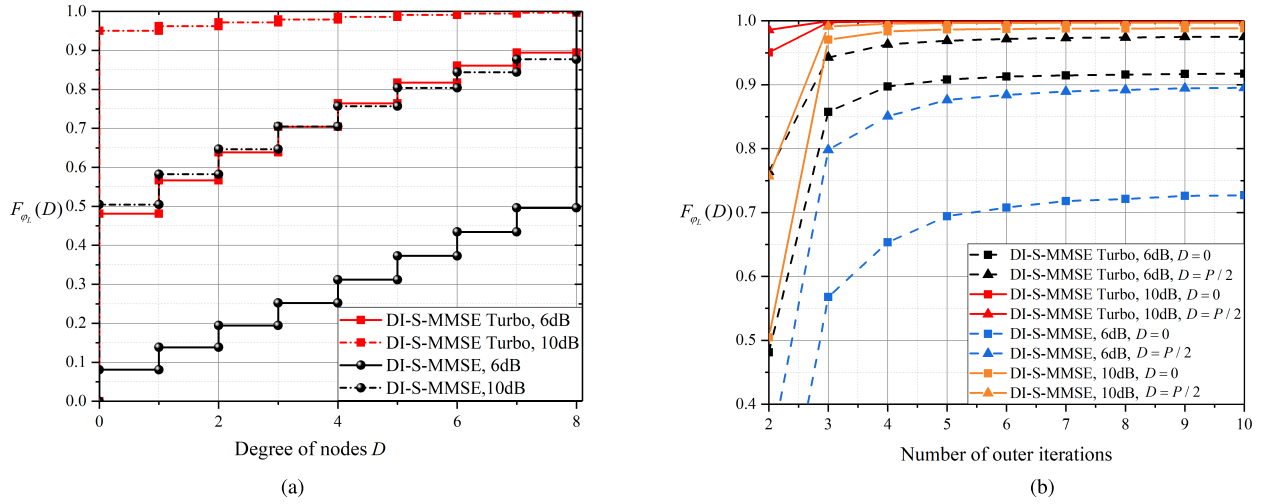
Fig. 7. The sparsity level of $\mathbf{L}$, measured by CDF $F_{\varphi_L}(D)$, where $E_b/N_0 = 6$ dB and 10 dB with $P = 8$. (a) The sparsity level of $\mathbf{L}$ at the second outer iteration. (b) The sparsity level of $\mathbf{L}$ versus the number of outer iterations, and D is set to be 0 and $P/2$ respectively.
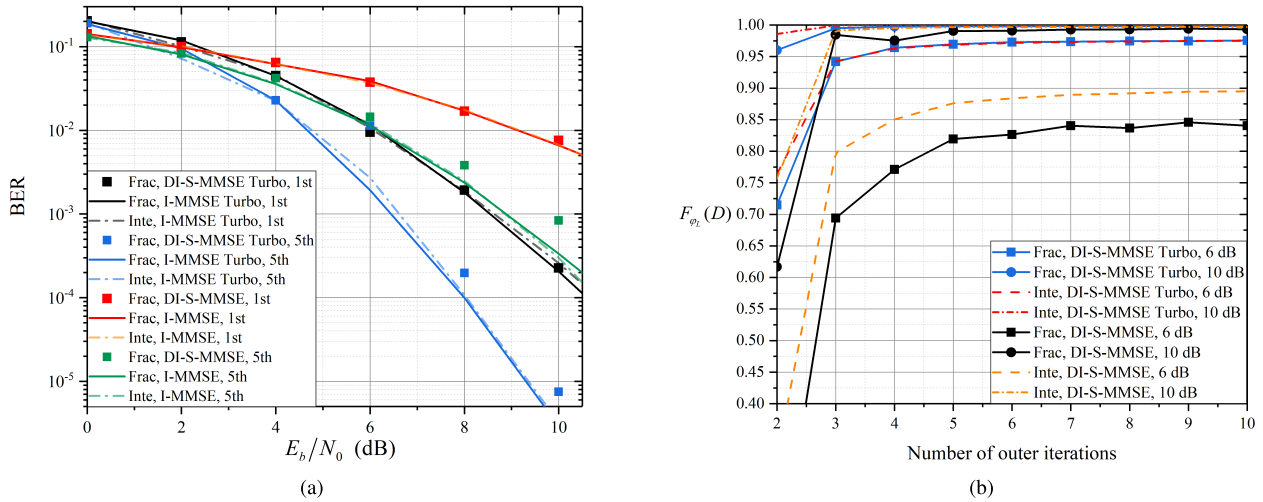

Fig. 8. The BER performance and the sparsity level of $\mathbf{L}$ with the fractional Doppler (denoted by "Frac"), where $P = 8$, $\epsilon_g = \epsilon_f = \epsilon_A = 10^{-3}$, $\epsilon_D = P/4$. For comparison, the BER and sparsity level with integer Doppler shifts (denoted by "Inte") are also presented. (a) The BER performance of our proposed DI-S-MMSE (turbo) equalizers. (b) The sparsity level of $\mathbf{L}$ versus the number of outer iterations at $E_b/N_0 = 6$ dB and 10 dB respectively, and D is set to be $P/2$.

In the following, we apply our equalizer to the fractional Doppler case, and analyze the corresponding BER and complexity respectively. Here, we use Jakes' formula to generate Doppler taps as $k_p + \kappa_p = k_{\max}\cos(\theta_p)$ for $p = 1, \ldots, P$, where $k_{\max} = 6$ and θ_p is uniformly distributed over $[-\pi, \pi]$. With the help of the decoder in the DI-S-MMSE turbo equalizer, $\zeta = P$ is sufficient to guarantee a good BER performance, while $\zeta = 3P$ is required for our uncoded equalizer. Fig. 8(a) shows the BER with $P = 8$, where the solid lines and square points denote the BER of the I-MMSE (turbo) and DI-S-MMSE (turbo) equalizers with the fractional Doppler respectively. Here, the former serves as the lower bound of the latter. We observe that there is no BER performance loss at the first outer iteration, which verifies the effectiveness of GMRES. However, even with $\zeta = 3P$ for our uncoded equalizer, around 0.8 dB E_b/N_0 loss is incurred at BER $= 10^{-3}$ due to the absence of the decoder. In contrast,

only $\zeta = P$ is sufficient for the DI-S-MMSE turbo equalizer and only causes around 0.3 dB performance loss at BER $= 10^{-4}$. This is because the decoder can output highly reliable extrinsic information, and thus the updated variances are close to zero. As a result, a highly sparse $\mathbf{A}$ can be derived without too much performance loss. More importantly, the dash lines represent the BER of the I-MMSE (turbo) equalizers without the fractional Doppler from Fig. 5(b). From the solid and dash lines, we observe that the I-MMSE (turbo) can achieve the same BER regardless of the fractional Doppler. This indicates that our equalizer can handle the fractional Doppler well.

In Fig. 8(b), we present the sparsity level $F_{\varphi_L}(D)$ of $\mathbf{L}$ with the fractional Doppler at $E_b/N_0 = 6$ dB and 10 dB. For comparison, we also present $F_{\varphi_L}(D)$ from Fig. 7(b) where the fractional Doppler does not exist. As shown in Fig. 8(b), $\mathbf{L}$ is much sparser in the DI-S-MMSE turbo equalizer (blue lines) than that in our uncoded equalizer (black lines). For

Low complexity LMMSE equalizer [10]	$\mathcal{O}(\frac{MN}{2}\log_2 M + MN(2l_{max}^2 + 2P^2 k_{max}))$
MPA detector [4]	$\mathcal{O}(n_{\text{iter}} MNP|S|))$
SPA detector [11]	$\mathcal{O}(n_{\text{iter}} MNP|\mathcal{S}|^P)$
Our proposed equalizer	$\mathcal{O}(MN(n_{cyc}|\mathcal{D}|j_m + \frac{n_{\text{iter}-1}}{n_p}\mathbb{E}\{\varphi_L^4\}))$

example, at the second outer iteration and $E_b/N_0 = 10$ dB, the former has around 96% nodes with degree not larger than $P/2$, while the latter has around 62% nodes with degree not larger than $P/2$. More importantly, compared with the integer Doppler case (red lines), the sparsity level of $\mathbf{L}$ in the DI-S-MMSE turbo equalizer is hardly influenced. This is because the updated variances derived from the decoder is so close to zero that the first term of $\mathbf{A} = \mathbf{H}_{\text{DD}}\mathbf{V}\mathbf{H}_{\text{DD}}^{\text{H}} + N_0\mathbf{I}$ is almost negligible. In this case, the diagonal elements of $\mathbf{A}$ have dominant magnitude, and thus $\mathbf{L}$ can still be highly sparse. This observation validates that the decoder can not only improve the BER performance, but also greatly reduce the complexity, even with the fractional Doppler.

The computational complexity of the aforementioned detectors is summarized in Table I, where n_{iter} is the number of iterations of the MPA and SPA detector, and also the number of outer iterations of our equalizer. In our equalizer, we use restarted GMRES and neglect the decoder for a fair comparison. It is observed that our equalizer has a complexity order linearly increasing with MN. Besides, since typically $\zeta = P$ is sufficient, even in the worst case, the overall complexity order of our equalizer increases with P^4 as discussed in Section VI-B. However, as observed in Fig. 7, the average complexity is far less than the worst case. In comparison with the LMMSE equalizer in [10] which has a log-linear order of complexity, our equalizer not only achieves a better BER performance, but also has an even lower complexity order. The MPA detector has a lower complexity than our equalizer but sacrifices the BER performance as shown in Fig. 5(a). For the SPA detector, the complexity order of the SPA detector is exponential to P. Hence, in richly scattered environment with large P, our equalizer has much lower complexity.

VIII. CONCLUSION

This paper developed a DI-S-MMSE turbo equalizer. To exploit the inherent sparsity of the channel matrix, we resorted to graph theory to analyze the sparse matrices and proposed two sparsification guidelines. At the initial outer iteration, we applied GMRES to avoid directly computing the inverse of the covariance matrix with little performance loss. In addition, GMRES can be readily extended to other LMMSE systems if the covariance matrix is sparse. Then, we modified FSPAI to derive an approximate inverse of the covariance matrix. Next, we proved the global convergence of full and restarted GMRES respectively, and demonstrated that the overall complexity order of our proposed equalizer increases linearly with the frame size. Simulation results demonstrated that our proposed equalizer can indeed deliver a great BER

performance with a tremendously reduced complexity and regardless of the fractional Doppler.

APPENDIX
PROOF OF THEOREM 2

After restarted GMRES performs j_m iterations, i.e., a restart cycle, the residual is given by

$$\|\mathbf{r}_{j_m}\| = \|\mathbf{r}_0 - \mathbf{A}\Delta\mathbf{f}'_{j_m}\| = \min_{\Delta\mathbf{f}_{j_m}\in\mathcal{K}_{j_m}} \|\mathbf{r}_0 - \mathbf{A}\Delta\mathbf{f}_{j_m}\|,$$
(44)

where $\Delta\mathbf{f}'_{j_m}$ is the solution of the LS problem. Considering $\Delta\mathbf{f}'_{j_m} \in \mathcal{K}_{j_m}$, $\mathbf{A}\Delta\mathbf{f}'_{j_m}$ belongs to a subspace spanned by $\{\mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{j_m}\mathbf{r}_0\}$. Then, from (44), the minimum residual is obtained when $\mathbf{r}_{j_m}$ is perpendicular to this subspace and $\mathbf{A}\Delta\mathbf{f}'_{j_m}$ is the projection of $\mathbf{r}_0$ onto this subspace. Hence, $\|\mathbf{r}_{j_m}\|^2 = \|\mathbf{r}_0\|^2 - \|\mathbf{A}\Delta\mathbf{f}'_{j_m}\|^2$, and $\mathbf{A}\Delta\mathbf{f}'_{j_m} = \mathbf{K}(\mathbf{K}^{\text{H}}\mathbf{K})^{-1}\mathbf{K}^{\text{H}}\mathbf{r}_0$ where $\mathbf{K} = [\mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{j_m}\mathbf{r}_0]$ [33]. Now, we have the relative residual

$$\|\mathbf{r}_{j_m}\|^2/\|\mathbf{r}_0\|^2 = 1 - \|\overline{\mathbf{K}}(\overline{\mathbf{K}}^{\text{H}}\overline{\mathbf{K}})^{-1}\overline{\mathbf{K}}^{\text{H}}\overline{\mathbf{r}}_0\|^2$$
$$= 1 - \overline{\mathbf{r}}_0^{\text{H}}\overline{\mathbf{K}}(\overline{\mathbf{K}}^{\text{H}}\overline{\mathbf{K}})^{-1}\overline{\mathbf{K}}^{\text{H}}\overline{\mathbf{r}}_0, \quad (45)$$

where $\overline{\mathbf{r}}_0 = \mathbf{r}_0/\|\mathbf{r}_0\|$ and $\overline{\mathbf{K}} = [\mathbf{A}\overline{\mathbf{r}}_0, \ldots, \mathbf{A}^{j_m}\overline{\mathbf{r}}_0]$. Then, the upper bound of (45) is given by

$$\frac{\|\mathbf{r}_{j_m}\|^2}{\|\mathbf{r}_0\|^2} \leq 1 - \overline{\mathbf{r}}_0^{\text{H}}\overline{\mathbf{K}}\overline{\mathbf{K}}^{\text{H}}\overline{\mathbf{r}}_0/\lambda_{\max}(\overline{\mathbf{K}}^{\text{H}}\overline{\mathbf{K}})$$

$$\overset{(a)}{\leq} 1 - \sum_{j=1}^{j_m} |\overline{\mathbf{r}}_0^{\text{H}}\mathbf{A}^j\overline{\mathbf{r}}_0|^2/\text{tr}(\overline{\mathbf{K}}^{\text{H}}\overline{\mathbf{K}}) = 1 - \varepsilon_{j_m},$$
(46)

where (a) follows from the fact that $\overline{\mathbf{K}}^{\text{H}}\overline{\mathbf{K}}$ is a Gram matrix and thus positive semidefinite. It is easy to know that $\text{tr}(\overline{\mathbf{K}}^{\text{H}}\overline{\mathbf{K}}) > 0$. On the other hand, since $\mathbf{A}^j$ is HPD, $\overline{\mathbf{r}}_0^{\text{H}}\mathbf{A}^j\overline{\mathbf{r}}_0 > 0$ for $j \geq 1$ and any $\overline{\mathbf{r}}_0$. Hence, for any $j_m \geq 1$, ε_{j_m} is always positive, and thus the relative residual is less than one. This proves the global convergence of restarted GMRES.

REFERENCES

[1] R. Hadani et al., "Orthogonal time frequency space modulation," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–6.

[2] R. Hadani and A. Monk, "OTFS: A new generation of modulation addressing the challenges of 5G," 2018, *arXiv:1802.02623*.

[3] S. K. Mohammed, "Derivation of OTFS modulation from first principles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 7619–7636, Aug. 2021.

[4] P. Raviteja, K. T. Phan, Y. Hong, and E. Viterbo, "Interference cancellation and iterative detection for orthogonal time frequency space modulation," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6501–6515, Oct. 2018.

[5] G. D. Surabhi, R. M. Augustine, and A. Chockalingam, "On the diversity of uncoded OTFS modulation in doubly-dispersive channels," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3049–3063, Jun. 2019.

[6] S. Li, J. Yuan, W. Yuan, Z. Wei, B. Bai, and D. W. K. Ng, "Performance analysis of coded OTFS systems over high-mobility channels," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 6033–6048, Sep. 2021.

[7] Z. Wei et al., "Orthogonal time-frequency space modulation: A promising next-generation waveform," *IEEE Wireless Commun. Mag.*, vol. 28, no. 4, pp. 136–144, Aug. 2021.

[8] P. Raviteja, K. T. Phan, and Y. Hong, "Embedded pilot-aided channel estimation for OTFS in delay–Doppler channels," *IEEE Trans. Veh. Tech.*, vol. 68, no. 5, pp. 4906–4917, May 2019.

[9] K. R. Murali and A. Chockalingam, "On OTFS modulation for high-Doppler fading channels," in *Proc. Inf. Theory Appl. Workshop (ITA)*, Feb. 2018, pp. 1–10.

[10] S. Tiwari, S. S. Das, and V. Rangamgari, "Low complexity LMMSE receiver for OTFS," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2205–2209, Dec. 2019.

[11] S. Li et al., "Hybrid MAP and PIC detection for OTFS modulation," *IEEE Trans. Veh. Technol.*, vol. 70, no. 7, pp. 7193–7198, Jul. 2021.

[12] W. Yuan, Z. Wei, J. Yuan, and D. W. K. Ng, "A simple variational Bayes detector for orthogonal time frequency space (OTFS) modulation," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7976–7980, Jul. 2020.

[13] Z. Yuan, F. Liu, W. Yuan, Q. Guo, Z. Wang, and J. Yuan, "Iterative detection for orthogonal time frequency space modulation with unitary approximate message passing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 714–725, Feb. 2022.

[14] S. Li, W. Yuan, Z. Wei, and J. Yuan, "Cross domain iterative detection for orthogonal time frequency space modulation," 2021, *arXiv:2101.03822*.

[15] P. Raviteja, Y. Hong, E. Viterbo, and E. Biglieri, "Practical pulse-shaping waveforms for reduced-cyclic-prefix OTFS," *IEEE Trans. Veh. Tech.*, vol. 68, no. 1, pp. 957–961, Jan. 2019.

[16] Z. Wei, W. Yuan, S. Li, J. Yuan, and D. W. K. Ng, "Transmitter and receiver window designs for orthogonal time-frequency space modulation," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2207–2223, Apr. 2021.

[17] H. V. Poor, *An Introduction to Signal Detection and Estimation*. Springer, 2013.

[18] M. Tuchler, A. C. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 673–683, Mar. 2002.

[19] G. H. Golub and C. F. Van Loan, *Matrix Computations* (Studies in the Mathematical Sciences). Baltimore, MD, USA: Johns Hopkins Univ., 1996.

[20] O. Østerby and Z. Zlatev, *Direct Methods for Sparse Matrices* (Lecture Notes in Computer Science), vol. 157. Berlin, Germany: Springer, 1983.

[21] M. Sedlacek, "Sparse approximate inverses for preconditioning, smoothing, and regularization," Ph.D. dissertation, Dept. Comput. Sci., Technische Universität München, München, Germany, 2012.

[22] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Statist. Comput.*, vol. 7, no. 3, pp. 856–869, 1986.

[23] W. E. Arnoldi, "The principle of minimized iterations in the solution of the matrix eigenvalue problem," *Quart. Appl. Math.*, vol. 9, no. 1, pp. 17–29, 1951.

[24] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Philadelphia, PA, USA: SIAM, 2003.

[25] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[26] L. Y. Kolotilina and A. Y. Yeremin, "Factorized sparse approximate inverse preconditionings I. Theory," *SIAM J. Matrix Anal. Appl.*, vol. 14, no. 1, pp. 45–58, Jan. 1993.

[27] I. E. Kaporin, "New convergence results and preconditioning strategies for the conjugate gradient method," *Numer. Linear Algebra Appl.*, vol. 1, no. 2, pp. 179–210, Mar. 1994.

[28] T. Huckle, "Factorized sparse approximate inverses for preconditioning," *J. Supercomput.*, vol. 25, no. 2, pp. 109–117, 2003.

[29] J. Liesen and P. Tichý, "Convergence analysis of Krylov subspace methods," *GAMM-Mitteilungen*, vol. 27, no. 2, pp. 153–173, Dec. 2004.

[30] Y. Saad, "Krylov subspace methods for solving large unsymmetric linear systems," *Math. Comput.*, vol. 37, no. 155, pp. 105–126, Jul. 1981.

[31] A. H. Baker, E. R. Jessup, and T. Manteuffel, "A technique for accelerating the convergence of restarted GMRES," *SIAM J. Matrix Anal. Appl.*, vol. 26, no. 4, pp. 962–984, 2005.

[32] 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.

[33] J. Zitko, "Generalization of convergence conditions for a restarted GMRES," *Numer. Linear Algebra with Appl.*, vol. 7, no. 3, pp. 117–131, Apr. 2000.

Haotian Li (Student Member, IEEE) received the B.E. degree in electronic and information engineering from the Harbin Institute of Technology, Harbin, China, in 2020, where he is currently pursuing the M.S. degree in electronic and information engineering. His research interests include signal processing and channel coding.

Qiyue Yu (Senior Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees from the Harbin Institute of Technology (HIT), China, in 2004, 2006, and 2010, respectively. From April 2007 to March 2008, she was at the Adachi Laboratory, Tohoku University, Japan, and was a Research Assistant of Tohoku University Global COE Program. In 2010, she was invited to the City University of Hong Kong to research on multi-user MIMO technology. She was invited to the University of Southern Queensland from January to March 2014. From September 2015 to September 2016, she was a Visiting Scholar at the University of California, Davis, and focused on coding theory. She is currently a Full Professor with the School of Electronics and Information Engineering, HIT. Her research interests include channel codes, multi-access techniques, equalization techniques, and MIMO for broadband wireless communications.