Fast Expectation Propagation for Sparse Signal Reconstruction With a Fourier Dictionary

Fangqing Xiao, Dirk Slock
Communication Systems Department
Eurecom, France
Email: {fangqing.xiao, dirk.slock}@eurecom.fr

***Abstract*—Sparse signal reconstruction (SSR) involves tackling large underdetermined systems of linear equations while incorporating constraints or regularizers. Expectation propagation (EP) emerges as a robust method for SSR, converting these constraints into prior information. However, the cubic complexity of matrix inversion per EP cycle hinders its implementation in large systems without approximation. In various applications like direction of arrival estimation (DoA), radar imaging etc., the signal to be recovered exhibits sparsity in the Fourier dictionary. To address this, we present a fast EP algorithm based on the Gohberg-Semencul (G-S) formula and Levinson-Durbin (L-D) type algorithm, boasting only quadratic complexity. Notably, no approximation operations or random measurement matrices are required for matrix inversion compared to approximate message passing (AMP) and other message passing based algorithms. Furthermore, it is compatible with non-identically and independently distributed (n.i.i.d.) priors. Numerical simulations conclusively demonstrate the efficacy of fast EP.**

I. Introduction

Compressed sensing (CS) [1] is a signal processing technique aimed at efficiently acquiring and reconstructing a sparse signal within an underdetermined linear system. It can be succinctly formulated as:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{v}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{C}^{N\times 1}$ represents the sparse signal vector to be recovered, $\boldsymbol{y} \in \mathbb{C}^{M\times 1}$ denotes the measurement data vector, $\boldsymbol{v} \in \mathbb{C}^{M\times 1}$ stands for the additive white complex Gaussian noise vector, and $\boldsymbol{H} \in \mathbb{C}^{M\times N}$ denotes the overcomplete dictionary matrix, where $M \ll N$.

One of the most well-known approaches is to frame this problem as a Bayesian inference problem, with a focus on determining the posterior probability. In this scenario, the Bernoulli-Gaussian model [2] which is understood as the minimization of the L_0 norm of the vector $\boldsymbol{x}$ is commonly employed to represent the prior. However, in practice, the posterior distribution often becomes intractable with this prior, necessitating the use of variational inference methods.

For an independent and identically distributed (i.i.d.) Gaussian/complex Gaussian matrix $\boldsymbol{H}$, the approximate message passing (AMP) [3], as proven in [4], was shown to be an asymptotically Bayesian-optimal algorithm for SSR when the belief-propagation threshold is smaller than the compression rate [5]. However, it has been recognized that the original AMP fails to converge for non-i.i.d. $\boldsymbol{H}$ [6]. Another approach is to leverage the expectation propagation (EP) method, which does not require any specific properties of the measurement matrix [7], [8]. Moreover, with unitarily invariant measurement matrices, EP is asymptotically Bayes-optimal [9] as AMP. However, EP entails higher computational complexity, primarily due to a high-dimension matrix inversion per EP cycle under large systems. Therefore, reducing the computational complexity of EP becomes crucial for enhancing its efficiency.

In fields such as array signal processing [10], Direction of Arrival (DoA) estimation [11], radar imaging [12], etc., the measurement matrix $\boldsymbol{H}$ often assumes the form of a Fourier matrix, which is unitarily invariant. Given this characteristic, we develop a fast-EP algorithm. The key feature of our fast-EP algorithm lies in its ability to express the matrix inverted per EP cycle as a Hermitian-Toeplitz matrix. The inversion of a Hermitian-Toeplitz matrix can be accelerated using the Gohberg-Semencul (G-S) formula [13] and Levinson-Durbin (L-D) type algorithm [14]. Moreover, owing to the Fourier matrix's properties, the fast Fourier transform (FFT) can be employed to enhance the computation efficiency of our algorithm [15]. It is noteworthy that the fast implementation based on the G-S formula has found success in various fields [16]. However, its application to EP for solving the sparse signal recovery problem is novel. Compared to standard EP, this approach can reduce complexity from cubic to quadratic complexity. While other EP-like algorithms, such as vector approximate message passing (VAMP) [17], also exhibit quadratic complexity, they are limited to the i.i.d. case and often require approximation. In contrast, our fast EP can handle the non-i.i.d. case. Numerical simulations validate the effectiveness of fast EP, confirming its potential in practical applications.

Notations: We denote vectors as $\boldsymbol{x}$ and matrices as $\boldsymbol{X}$.

For a complex Gaussian random vector $\boldsymbol{x}$ with mean $\boldsymbol{m}$ and covariance $\boldsymbol{\Sigma}$, its probability density function is represented as $\mathcal{CN}(\boldsymbol{x};\boldsymbol{m},\boldsymbol{\Sigma})$. The symbols $\boldsymbol{I}_M$ and $\boldsymbol{0}_M$ signify the $M\times M$ identity matrix and zero vector of size M, respectively. The notation $(\cdot)^T$ indicates the transpose of a matrix, while $\mathbb{C}$ denotes the complex field. The function $diag(\boldsymbol{C})$ returns a vector with its elements being the diagonal elements of the square matrix $\boldsymbol{C}$. Additionally, $(\cdot)^*$, $(\cdot)^T$ and $(\cdot)^H$ represent the conjugate operator, the transpose operator and the conjugate transpose operator, respectively.

II. Brief Review of System Model

In this paper, we consider the Bernoulli-Gaussian prior of n.i.i.d. $\boldsymbol{x}$, and its probability density function (pdf) can be expressed as:

$$p(\boldsymbol{x};\rho,\boldsymbol{\xi})=\prod_{i=1}^{N}p(\boldsymbol{x}_i)=\prod_{i=1}^{N}(1-\rho)\delta(x_i)+\rho\,\mathcal{CN}(x_i;0,\xi_i^{-1}), \tag{2}$$

$\delta(\cdot)$ represents the Dirac delta function [18], and $\rho\in[0,1]$ signifies the sparse coefficient, utilized to model any prior knowledge regarding the sparsity of the signal. Here, ξ_i represents the precision (inverse variance) of x_i, where $x_i\neq 0$. Typically, ρ and $\boldsymbol{\xi}$ are deterministic yet unknown.

The observation noise $\boldsymbol{v}$ is characterized as a zero-mean complex white Gaussian vector with an unknown precision λ. Its pdf is defined as follows:

$$p(\boldsymbol{v};\lambda)=\prod_{i=1}^{M}p(v_i)=\mathcal{CN}(\boldsymbol{v};\boldsymbol{0}_M,\lambda^{-1}\boldsymbol{I}_M). \tag{3}$$

Based on the above assumptions, the marginal density of $\boldsymbol{x}$ can be derived as follows:

$$p(\boldsymbol{x}|\boldsymbol{y};\rho,\boldsymbol{\xi},\lambda)=\frac{1}{Z_p}p(\boldsymbol{y}|\boldsymbol{x};\lambda)p(\boldsymbol{x};\rho,\boldsymbol{\xi}), \tag{4}$$

where the partition function Z_p is determined through integration:

$$Z_p=\int p(\boldsymbol{y}|\boldsymbol{x};\lambda)p(\boldsymbol{x};\rho,\boldsymbol{\xi})\mathrm{d}\boldsymbol{x}. \tag{5}$$

Using the minimum mean square error (MMSE) estimator, we aim to find a recovered sparse signal $\hat{\boldsymbol{x}}$ whose components are the first-order moment of function (4). However, even in the absence of unknown parameters, the MMSE estimator becomes impractical due to the exponentially growing computational complexity. Additionally, because of the non-convex nature of the Bernoulli-Gaussian prior, there is no technique capable of directly solving the maximum a posteriori (MAP) optimization problem. Therefore, it becomes crucial to approximate the posterior distribution with another tractable distribution. To accomplish this objective, we introduce the expectation propagation (EP) approximation schema, which relies on an adaptive complex Gaussian approximation.

III. Expectation Propagation

The purpose of the EP algorithm is to find a complex Gaussian distribution $\mathcal{CN}(\boldsymbol{x};\boldsymbol{m},\boldsymbol{C_m})$ approximating the posterior $p(\boldsymbol{x}|\boldsymbol{y};\rho,\boldsymbol{\xi},\lambda)$ of (4) as:

$$p(\boldsymbol{x}|\boldsymbol{y};\rho,\boldsymbol{\xi},\lambda)\approx q(\boldsymbol{x})=\mathcal{CN}(\boldsymbol{x};\boldsymbol{m},\boldsymbol{C_m}), \tag{6}$$

where ρ, ξ and λ are supposed to be known. Although they are typically unknown in real scenarios, the main focus of this paper is not to study how to estimate them jointly. However, they can be estimated using the Expectation Maximization (EM) technique, with further details available in [19].

To obtain $\boldsymbol{m}$ and $\boldsymbol{C_m}$ in the linear mixing data model, we factorize the approximate distribution $q(\boldsymbol{x})$ as follows:

$$q(\boldsymbol{x})=\prod_{i=1}^{N}q(x_i;m_i,[\boldsymbol{C_m}]_{ii})\propto p(\boldsymbol{y}|\boldsymbol{x};\lambda)\prod_{i=1}^{N}f_i(x_i), \tag{7}$$

where $f_i(x_i)$ is supposed to be complex Gaussian $\mathcal{CN}(x_i;p_i,[\boldsymbol{C_p}]_{ii})$ and $\prod_{i=1}^{N}f_i(x_i)=\mathcal{CN}(\boldsymbol{x};\boldsymbol{p},\boldsymbol{C_p})$. Here, $\boldsymbol{C_p}$ is always a diagonal matrix as each x_i is independent. The EP is given as:

1) Initialize the factors: $f_i(x_i)$
2) Compute the posterior for $\boldsymbol{x}$ from the product of $f_i(x_i)$:

$$q(\boldsymbol{x})=\frac{p(\boldsymbol{y}|\boldsymbol{x})\prod_{i=1}^{N}f_i(x_i)}{\int p(\boldsymbol{y}|\boldsymbol{x})\prod_{i=1}^{N}f_i(x_i)d\boldsymbol{x}}. \tag{8}$$

3) Until all $f_i(x_i)$ converge:
   a) Choose a $f_i(x_i)$ to refine
   b) Remove $f_i(x_i)$ from the posterior and integral out $\boldsymbol{x}$ except x_i to get an extrinsic:

$$b(x_i)=\int\frac{q(\boldsymbol{x})}{f_i(x_i)}d\boldsymbol{x}_{\backslash i}. \tag{9}$$

   c) Combine with real prior $p(x_i|\xi_i)$ and minimize Kullback–Leibler (KL) divergence to get an approximate marginal posterior $q(x_i)$:

$$q(x_i)=\arg\min_{q(x_i)}D_{KL}\left[b(x_i)p(x_i|\xi_i)||q(x_i)\right]. \tag{10}$$

   d) Update $f_i(x_i)\propto q(x_i)/b(x_i)$
4) Generate an approximated posterior $q(\boldsymbol{x})$:

$$q(\boldsymbol{x})=\frac{p(\boldsymbol{y}|\boldsymbol{x})\prod_{i=1}^{N}f_i(x_i)}{\int p(\boldsymbol{y}|\boldsymbol{x})\prod_{i=1}^{N}f_i(x_i)d\boldsymbol{x}}. \tag{11}$$

This algorithm consistently converges to a fixed point with complex Gaussian approximation factors. However, if initialized too far away from the fixed point, it may diverge. In (8), two operations are required as follows:

$$C_{\boldsymbol{m}} = (\lambda \boldsymbol{H}^H \boldsymbol{H} + \boldsymbol{C}_{\boldsymbol{p}}^{-1})^{-1}; \tag{12}$$

$$\boldsymbol{m} = \boldsymbol{C}_{\boldsymbol{m}}(\lambda \boldsymbol{H}^H \boldsymbol{y} + \boldsymbol{C}_{\boldsymbol{p}}^{-1}\boldsymbol{p}), \tag{13}$$

where $\boldsymbol{p}$ and $\boldsymbol{C}_{\boldsymbol{p}}$ represent the mean and covariance of the approximated prior of $\boldsymbol{x}$ during EP, respectively.

According to the Woodbury matrix identity [20], the matrix $\boldsymbol{C}_{\boldsymbol{m}}$ in (12) can be written as:

$$\boldsymbol{C}_{\boldsymbol{m}} = \boldsymbol{C}_{\boldsymbol{p}} - \lambda \boldsymbol{C}_{\boldsymbol{p}} \boldsymbol{H}^H \boldsymbol{\Sigma}^{-1} \boldsymbol{H} \boldsymbol{C}_{\boldsymbol{p}}, , \tag{14}$$

where

$$\boldsymbol{\Sigma} = \boldsymbol{I}_M + \lambda \boldsymbol{H} \boldsymbol{C}_{\boldsymbol{p}}^{-1} \boldsymbol{H}^H. \tag{15}$$

Also, the vector $\boldsymbol{m}$ can be written as following by substituting (14) and (15) into (13):

$$\begin{aligned} \boldsymbol{m} &= \lambda \boldsymbol{C}_{\boldsymbol{p}} \boldsymbol{H}^H \boldsymbol{\Sigma}^{-1} \boldsymbol{y} + (\boldsymbol{I}_N - \lambda \boldsymbol{C}_{\boldsymbol{p}} \boldsymbol{H}^H \boldsymbol{\Sigma}^{-1} \boldsymbol{H}) \boldsymbol{p} \\ &= \boldsymbol{p} + \lambda \boldsymbol{C}_{\boldsymbol{p}} \boldsymbol{H}^H \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{H} \boldsymbol{p}). \end{aligned} \tag{16}$$

In EP, the process known as "learning" involves finding the optimal mean of the posterior distribution of $\boldsymbol{x}$. It entails updating and iterating $\boldsymbol{m}$ and $\boldsymbol{C}_{\boldsymbol{m}}$, using them to derive the new prior distribution of $\boldsymbol{x}$. Once the iterative process converges, the recovered sparse signal $\hat{\boldsymbol{x}}$ is determined by the mean vector of the posterior distribution:

$$\hat{\boldsymbol{x}} = \boldsymbol{m}, \tag{17a}$$

$$\hat{\boldsymbol{C}}_{\boldsymbol{x}} = diag(\boldsymbol{C}_{\boldsymbol{m}}). \tag{17b}$$

From the above iteration procedure, we observe that the key to utilizing EP to solve the sparse signal reconstruction problem lies in computing $\boldsymbol{C}_{\boldsymbol{m}}$ and $\boldsymbol{m}$ during each iteration. This process, from (14) to (16), involves the inversion of an $M \times M$ matrix $\boldsymbol{\Sigma}$, where M is the dimension of the observed data. It is well-known that the general inverse method requires $\mathcal{O}(M^3)$ operations, resulting in a significant computational burden even for moderate data sizes.

Apart from using approximations under large system assumptions like AMP and VAMP, there is no general method available to accelerate the computation of the inverse of $\boldsymbol{\Sigma}$ for an arbitrary dictionary matrix $\boldsymbol{\Sigma}$. However, we find that the Fourier dictionary is commonly applied in many practical applications. Fortunately, when the dictionary matrix $\boldsymbol{H}$ is a Fourier matrix, $\boldsymbol{\Sigma}$ exhibits properties that enable significant acceleration of computations involving $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^{-1}$. In the following section, we present the fast implementations of these methods.

IV. Fast Computation with Fourier Matrix

In the model described in (1), the overcomplete dictionary matrix $\boldsymbol{H}$ is an overcomplete Fourier matrix. The $(n+1)$th column $\boldsymbol{h}_M(\omega_n)$ of $\boldsymbol{H}$ is defined as follow

$$\boldsymbol{h}_M(\omega_n) = [1, e^{-j\omega_n}, \cdots, e^{-j(M-1)\omega_n}]^T, \tag{18}$$

where $\omega_n = 2\pi n/N$, $n = 0, \cdots, N-1$. In this case, the matrix $\boldsymbol{\Sigma}$ defined in (15) can be denoted as:

$$\boldsymbol{\Sigma} = \boldsymbol{I}_M + \lambda \boldsymbol{\Psi}, \tag{19}$$

where $\boldsymbol{\Psi} = \boldsymbol{H} \boldsymbol{C}_{\boldsymbol{p}}^{-1} \boldsymbol{H}^H$ is a Hermitian-Toeplitz matrix which has the following structure:

$$\boldsymbol{\Psi} = \begin{bmatrix} a_0 & a_1^* & \cdots & a_{M-1}^* \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1^* \\ a_{M-1} & \cdots & a_1 & a_0 \end{bmatrix}. \tag{20}$$

$\boldsymbol{C}_{\boldsymbol{p}}$ is a diagonal matrix which is given as:

$$\boldsymbol{C}_{\boldsymbol{p}} = \begin{bmatrix} \gamma_0^{-1} & & & \\ & \gamma_1^{-1} & & \\ & & \ddots & \\ & & & \gamma_{N-1}^{-1} \end{bmatrix}. \tag{21}$$

Therefore, using the definition of $\boldsymbol{\Psi}$, given $\boldsymbol{C}_{\boldsymbol{p}}$, we have

$$a_m = \sum_{n=0}^{N-1} \gamma_n e^{-j2\pi mn/N}, m \in [0, \cdots, M-1]. \tag{22}$$

Referring to the definition of the Fourier transform, a_m in (22) can be computed by applying an N-point FFT to the vector composed of all γ_n in sequence and extracting the first M values. According to (20), the matrix $\boldsymbol{\Psi}$ can be constructed, and $\boldsymbol{\Sigma}$ can be computed using (19) with the given $\boldsymbol{\Psi}$. Clearly, the computation of $\boldsymbol{\Psi}$ requires $\mathcal{O}(N \log_2 N)$ floating-point operations.

As $\boldsymbol{\Psi}$ is a Hermitian-Toeplitz matrix, $\boldsymbol{\Sigma}$ also possesses the Hermitian-Toeplitz structure and evidently shares the same characteristics as $\boldsymbol{\Psi}$. Compared to an arbitrary matrix, computations involving a Toeplitz matrix and its inverse can be significantly simplified due to the structural properties of the Toeplitz matrix. The inverse of a Toeplitz matrix can be decomposed using the Gohberg-Semencul formula and has a low displacement rank, leading to reduced computational complexity. Given that $\boldsymbol{\Sigma}$ is a Hermitian-Toeplitz matrix, it can be represented as:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_0 & \boldsymbol{\sigma}_{M-1}^H \\ \boldsymbol{\sigma}_{M-1} & \boldsymbol{\Sigma}_{M-1} \end{bmatrix} \tag{23}$$

$$= \begin{bmatrix} \boldsymbol{\Sigma}_{M-1} & \tilde{\boldsymbol{\sigma}}_{M-1}^* \\ \tilde{\boldsymbol{\sigma}}_{M-1}^T & \sigma_0 \end{bmatrix}, \tag{24}$$

where $\mathbf{\Sigma}_{M-1}$ denotes the $(M-1)\times(M-1)$ submatrix of $\mathbf{\Sigma}\in\mathbb{C}^{M\times M}$, and $\mathbf{\Sigma}_{M-1}$ is also a Hermitian-Toeplitz matrix. $\boldsymbol{\sigma}_{M-1}=[\sigma_1,\sigma_2,\cdots,\sigma_{M-1}]^T$ and $\tilde{\boldsymbol{\sigma}}_{M-1}=[\sigma_{M-1},\cdots,\sigma_2,\sigma_1]^T$. Then, $\mathbf{\Sigma}^{-1}$ can be computed by applying the matrix inversion formula to the right-hand sides (RHS) of (23) and (24) as follows:

$$\mathbf{\Sigma}^{-1}=\begin{bmatrix}0 & \mathbf{0}_{M-1}^T\\ \mathbf{0}_{M-1} & \mathbf{\Sigma}_{M-1}^{-1}\end{bmatrix}+\frac{1}{\epsilon_{M-1}}\begin{bmatrix}1\\ \boldsymbol{a}_{N-1}\end{bmatrix}\begin{bmatrix}1 & \boldsymbol{a}_{N-1}^H\end{bmatrix} \tag{25}$$

$$=\begin{bmatrix}\mathbf{\Sigma}_{M-1}^{-1} & \mathbf{0}_{M-1}\\ \mathbf{0}_{M-1}^T & 0\end{bmatrix}+\frac{1}{\varepsilon_{M-1}}\begin{bmatrix}\boldsymbol{b}_{N-1}^*\\ 1\end{bmatrix}\begin{bmatrix}\boldsymbol{b}_{N-1}^T & 1\end{bmatrix}, \tag{26}$$

where

$$\boldsymbol{a}_{N-1}=-\mathbf{\Sigma}_{M-1}^{-1}\boldsymbol{\sigma}_{M-1}, \tag{27}$$

$$\epsilon_{M-1}=\sigma_0-\boldsymbol{\sigma}_{M-1}^H\mathbf{\Sigma}_{M-1}^{-1}\boldsymbol{\sigma}_{M-1}, \tag{28}$$

$$\boldsymbol{b}_{N-1}^*=-\mathbf{\Sigma}_{M-1}^{-1}\tilde{\boldsymbol{\sigma}}_{M-1}^*, \tag{29}$$

$$\varepsilon_{M-1}=\sigma_0-\tilde{\boldsymbol{\sigma}}_{M-1}^T\mathbf{\Sigma}_{M-1}^{-1}\tilde{\boldsymbol{\sigma}}_{M-1}^*. \tag{30}$$

For the sake of convenience, we define a matrix $\boldsymbol{T}_M\in\mathbb{C}^{M\times M}$ in which the sub-diagonal elements are one and the other elements are 0:

$$\boldsymbol{T}_M=\begin{bmatrix}0 & \cdots & 0 & 1\\ 0 & \cdots & 1 & 0\\ \vdots & \iddots & \vdots & \vdots\\ 1 & \cdots & 0 & 0\end{bmatrix}, \tag{31}$$

therefore, $\tilde{\boldsymbol{\sigma}}_{M-1}$ in (24) can be obtained as:

$$\tilde{\boldsymbol{\sigma}}_{M-1}=\boldsymbol{T}_{M-1}\boldsymbol{\sigma}_{M-1}. \tag{32}$$

According to the basic property of a Hermitian-Toeplitz matrix, we can get:

$$\mathbf{\Sigma}_{M-1}^T=\mathbf{\Sigma}_{M-1}^*=\boldsymbol{T}_{M-1}\mathbf{\Sigma}_{M-1}\boldsymbol{T}_{M-1}, \tag{33}$$

then (27) and (28) can be rewritten as:

$$\begin{aligned}\boldsymbol{a}_{N-1}&=-\mathbf{\Sigma}_{M-1}^{-1}\boldsymbol{\sigma}_{M-1}=(-\mathbf{\Sigma}_{M-1}^{-*}\boldsymbol{\sigma}_{M-1}^*)^*\\ &=-(\mathbf{\Sigma}_{M-1}^{-T}\boldsymbol{\sigma}_{M-1}^*)^*\\ &=-(\boldsymbol{T}_{M-1}\mathbf{\Sigma}_{M-1}^{-1}\boldsymbol{T}_{M-1}\boldsymbol{\sigma}_{M-1}^*)^*\\ &=-(\boldsymbol{T}_{M-1}\mathbf{\Sigma}_{M-1}^{-1}\tilde{\boldsymbol{\sigma}}_{M-1}^*)^*\\ &=\boldsymbol{T}_{M-1}(-\mathbf{\Sigma}_{M-1}^{-*}\tilde{\boldsymbol{\sigma}}_{M-1})=\tilde{\boldsymbol{b}}_{M-1};\end{aligned} \tag{34}$$

$$\begin{aligned}\epsilon_{M-1}&=\sigma_0-\boldsymbol{\sigma}_{M-1}^H\mathbf{\Sigma}_{M-1}^{-1}\boldsymbol{\sigma}_{M-1}\\ &=\sigma_0-\boldsymbol{\sigma}_{M-1}^H\boldsymbol{T}_{M-1}\mathbf{\Sigma}_{M-1}^{-T}\boldsymbol{T}_{M-1}\boldsymbol{\sigma}_{M-1}\\ &=\sigma_0-\tilde{\boldsymbol{\sigma}}_{M-1}^H\mathbf{\Sigma}_{M-1}^{-T}\tilde{\boldsymbol{\sigma}}_{M-1}\\ &=(\sigma_0-\tilde{\boldsymbol{\sigma}}_{M-1}^T\mathbf{\Sigma}_{M-1}^{-1}\tilde{\boldsymbol{\sigma}}_{M-1}^*)^T=\varepsilon_{M-1}.\end{aligned} \tag{35}$$

Substituting the results of (34) and (35) into (26) yields:

$$\mathbf{\Sigma}^{-1}=\begin{bmatrix}\mathbf{\Sigma}_{M-1}^{-1} & \mathbf{0}\\ \mathbf{0}^T & 0\end{bmatrix}+\frac{1}{\varepsilon_{M-1}}\begin{bmatrix}\tilde{\boldsymbol{a}}_{N-1}^*\\ 1\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{a}}_{N-1}^T & 1\end{bmatrix}. \tag{36}$$

We define another matrix $\boldsymbol{S}_M\in\mathbb{C}^{M\times M}$, which is a lower-triangular matrix where the -1th main diagonal elements are 1, and the other elements are 0, as shown below:

$$\boldsymbol{S}_M=\begin{bmatrix}0 & 0 & \cdots & 0\\ 1 & 0 & \cdots & 0\\ \vdots & \ddots & \ddots & \vdots\\ 0 & \cdots & 1 & 0\end{bmatrix}. \tag{37}$$

Then, we define the displacement representation [13] $\nabla\mathbf{\Sigma}_M^{-1}$ of $\mathbf{\Sigma}_M^{-1}$ as:

$$\begin{aligned}&\nabla\mathbf{\Sigma}_M^{-1}=\mathbf{\Sigma}_M^{-1}-\boldsymbol{S}_M\mathbf{\Sigma}_M^{-1}\boldsymbol{S}_M^T\\ &=\mathbf{\Sigma}_M^{-1}-\boldsymbol{S}_M\left(\begin{bmatrix}\mathbf{\Sigma}_{M-1}^{-1} & \mathbf{0}\\ \mathbf{0}^T & 0\end{bmatrix}+\frac{1}{\epsilon_{M-1}}\begin{bmatrix}\tilde{\boldsymbol{a}}_{N-1}^*\\ 1\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{a}}_{N-1}^T & 1\end{bmatrix}\right)\boldsymbol{S}_M^T\\ &=\frac{1}{\epsilon_{M-1}}\begin{bmatrix}1\\ \boldsymbol{a}_{M-1}\end{bmatrix}\begin{bmatrix}1 & \boldsymbol{a}_{M-1}^H\end{bmatrix}-\frac{1}{\epsilon_{M-1}}\begin{bmatrix}0\\ \tilde{\boldsymbol{a}}_{M-1}^*\end{bmatrix}\begin{bmatrix}0 & \tilde{\boldsymbol{a}}_{M-1}^T\end{bmatrix}.\end{aligned} \tag{38}$$

Let

$$\boldsymbol{\alpha}_M=\frac{1}{\sqrt{\epsilon_{M-1}}}\begin{bmatrix}1\\ \boldsymbol{a}_{M-1}\end{bmatrix}, \tag{39}$$

$$\boldsymbol{\beta}_M=\frac{1}{\sqrt{\epsilon_{M-1}}}\begin{bmatrix}0\\ \tilde{\boldsymbol{a}}_{M-1}^*\end{bmatrix}, \tag{40}$$

then $\boldsymbol{\alpha}_M$ in (40) can be represented as:

$$\nabla\mathbf{\Sigma}_M^{-1}=\boldsymbol{\alpha}_M\boldsymbol{\alpha}_M^H-\boldsymbol{\beta}_M\boldsymbol{\beta}_M^H. \tag{41}$$

By using the property that $(\boldsymbol{S}_M)^M\mathbf{\Sigma}_M^{-1}(\boldsymbol{S}_M^T)^M=0$ and (41), $\mathbf{\Sigma}_M^{-1}$ can be computed as:

$$\mathbf{\Sigma}_M^{-1}=\sum_{m=0}^{M-1}(\boldsymbol{S}_M)^m(\boldsymbol{\alpha}_M\boldsymbol{\alpha}_M^H-\boldsymbol{\beta}_M\boldsymbol{\beta}_M^H)(\boldsymbol{S}_M^T)^m \tag{42}$$

$$=\frac{1}{\epsilon_{M-1}}\sum_{m=0}^{M-1}(\boldsymbol{S}_M)^m(\begin{bmatrix}1\\ \boldsymbol{a}_{M-1}\end{bmatrix}\begin{bmatrix}1 & \boldsymbol{a}_{M-1}\end{bmatrix}-\begin{bmatrix}0\\ \tilde{\boldsymbol{a}}_{M-1}^*\end{bmatrix}\begin{bmatrix}0 & \tilde{\boldsymbol{a}}_{M-1}^*\end{bmatrix})(\boldsymbol{S}_M^T)^m \tag{43}$$

$$=\sum_{m=0}^{M-1}(\boldsymbol{S}_M)^m\nabla\mathbf{\Sigma}_M^{-1}(\boldsymbol{S}_M^T)^m. \tag{44}$$

Then defining two $M\times M$ lower triangular matrices $\mathbf{\Delta}_{\boldsymbol{\alpha}_M}$ and $\mathbf{\Delta}_{\boldsymbol{\beta}_M}$ as:

$$\mathbf{\Delta}_{\boldsymbol{\alpha}_M}=[\boldsymbol{\alpha}_M,\boldsymbol{S}_M\boldsymbol{\alpha}_M,\cdots,(\boldsymbol{S}_M)^{M-1}\boldsymbol{\alpha}_M]; \tag{45}$$

$$\mathbf{\Delta}_{\boldsymbol{\beta}_M}=[\boldsymbol{\beta}_M,\boldsymbol{S}_M\boldsymbol{\beta}_M,\cdots,(\boldsymbol{S}_M)^{M-1}\boldsymbol{\beta}_M]. \tag{46}$$

By using (45) and (46), (42) can be rewritten as:

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Delta}_{\boldsymbol{\alpha}_M}\boldsymbol{\Delta}_{\boldsymbol{\alpha}_M}^H - \boldsymbol{\Delta}_{\boldsymbol{\beta}_M}\boldsymbol{\Delta}_{\boldsymbol{\beta}_M}^H. \tag{47}$$

Then, the expression on the right-hand side of (47) is termed the G-S formula, and $\boldsymbol{\alpha}_M$ and $\boldsymbol{\beta}_M$ are the G-S-type factors of $\boldsymbol{\Sigma}_M^{-1}$. These factors can be computed using the Levinson-Durbin (L-D)-type algorithm with only $\mathcal{O}(M^2)$ floating-point operations (flops), as provided below:

1) Initializing ϵ_1 and a_1:
   a) $a_1 = \frac{-\sigma_1}{\sigma_0}$;
   b) $\epsilon_1 = \sigma_0 + a_1\sigma_1^*$.
2) For $m = 2, \cdots, M-1$:
   a) $\phi_{m-1} = \boldsymbol{a}_{m-1}^T \boldsymbol{S}_{m-1}\boldsymbol{\sigma}_{M-1}^* + \sigma_m$;
   b) $\boldsymbol{a}_m = \begin{bmatrix}\boldsymbol{a}_{m-1}\\ 0\end{bmatrix} - \frac{\phi_{m-1}}{\epsilon_{m-1}}\begin{bmatrix}\boldsymbol{S}_{m-1}\boldsymbol{a}_{m-1}^*\\ 1\end{bmatrix}$;
   c) $\epsilon_m = \epsilon_{m-1} - |\phi_{m-1}|^2/\epsilon_{m-1}$.
3) Calculating displacement matrix $\nabla\boldsymbol{\Sigma}_M^{-1}$ in (41).
4) Calculating $\boldsymbol{\Sigma}_M^{-1}$ in (46).

As stated above, the computations of $\boldsymbol{m}$ and $diag(\boldsymbol{C_m})$ are crucial during a EP iteration. For $[\boldsymbol{C_m}]_{nn}$, it is straightforward to obtain that

$$[\boldsymbol{C_m}]_{nn} = [\boldsymbol{C_p}]_{nn} - \lambda[\boldsymbol{C_p}]_{nn}^2[\boldsymbol{H}^H\boldsymbol{\Sigma}^{-1}\boldsymbol{H}]_{nn}. \tag{48}$$

Let

$$\boldsymbol{Q} = \boldsymbol{H}^H\boldsymbol{\Sigma}^{-1}\boldsymbol{H}, \tag{49}$$

We define a vector $\boldsymbol{q} = [q_0, q_1, \cdots, q_{N-1}]^T$, which comprises all the diagonal elements of $\boldsymbol{Q}$. The $(n+1)$th value of $\boldsymbol{q}_n$ can be computed as:

$$q_n = \sum_{k=-M+1}^{M+1} z_k \exp^{-j2\pi kn/N}, \tag{50}$$

where z_k is the sum of all the entries on the kth main diagonal of $\boldsymbol{\Sigma}^{-1}$. It should be noted that $z_{-k} = z_k^*$ since $\boldsymbol{\Sigma}^{-1}$ is a Hermitian matrix.

Let $\widetilde{\boldsymbol{z}} = [z_0, z_1, \cdots, z_{N-1}, \boldsymbol{0}_{N+1-2M}^T, z_{N-1}^*, \cdots, z_1^*]^T$, (50) can be calculated by:

$$q_n = \sum_{k=0}^{N-1} \widetilde{\boldsymbol{z}}_k \exp^{-j2\pi nk/N}, \tag{51}$$

which implies that $\boldsymbol{q}$ can be computed using FFT. Therefore, $diag(\boldsymbol{C_m})$ can be computed via (48) and the FFT of (51) with $\mathcal{O}(N\log_2 N)$ floating-point operations. The computation of $\boldsymbol{m}$ in (16) can be divided into several steps to avoid matrix multiplication with $\mathcal{O}(N^2)$ flops. Therefore, the overall computational complexity per EP iteration is $\mathcal{O}(N^2)$ as $N \gg M$.

V. Simulation

In this section, the performance of fast-EP is evaluated through numerical simulations. Additionally, fast-EP is compared with normal EP under various scenarios. To illustrate the reconstruction performance of the algorithm, we define the normalized root-mean-square error (NRMSE) of reconstruction as:

$$\text{NRMSE} = \frac{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}, \tag{52}$$

where $\boldsymbol{x}$ denotes the real signal and $\hat{\boldsymbol{x}}$ is the estimated signal. The size of the Fourier dictionary matrix $\boldsymbol{H}$ is set to be 64×256 with a frequency oversampling factor $N/M = 4$. The signal $\boldsymbol{x}$ is generated with respect to the prior (2) with $\rho = 0.95$ and $\boldsymbol{\xi} = [1, 2, \cdots, 256]^T$. During the simulation, ρ and $\boldsymbol{\xi}$ are assumed to be known. The signal-to-noise ratio (SNR) is set to be 10 dB. In Fig. 1, we illustrate the signal reconstruction results of the fast EP algorithm.

The NRMSE and computation time between fast EP and normal EP are estimated by averaging the results of 100 independent experiments. The reconstruction results of fast-EP are good, which verifies its efficiency in sparse signal estimation. The NRMSE and computational time of EP and fast-EP are calculated by averaging the results of 100 independent experiments. The estimated average computational time and the average NRMSE are listed in Table I. The computational time of fast-EP is almost 14 times shorter than that of normal EP. It is worth pointing out, however, that the actual efficiency could potentially be further improved, as the authors' limited programming skills may have prevented them from fully utilizing the capabilities of fast-EP algorithm.

TABLE I
Average Computation Time and Average NRMSE of EP and fast-EP

Algorithm	Average Computation Time	Average NRMSE
EP	2.7153s	0.8598
fast-EP	0.1931s	0.8598

VI. Conclusion

This paper introduces a fast EP algorithm for achieving sparse signal reconstruction when the measurement matrix is a Fourier matrix. Since the matrix to be inverted in each iteration within normal EP is a Hermitian-Toeplitz matrix, its inverses can be decomposed using G-S factorization, which can be solved via the Levinson-Durbin algorithm. Additionally, due to the Fourier matrix, FFT can also be applied to reduce computational complexity. The efficiency of SSR is significantly enhanced without compromising accuracy or resorting to approximation under large system assumptions, as

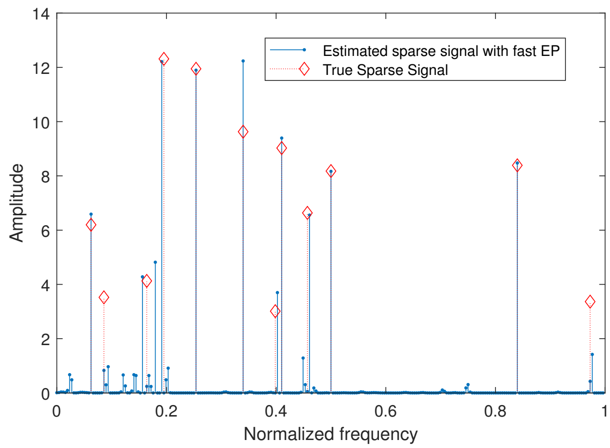

Fig. 1. SSR result of fast-EP

compared to AMP and VAMP. Moreover, it remains applicable w.r.t. n.i.i.d. prior distributions. In the future, we aim to explore the application of this fast-EP to practical scenarios in the real word.

Acknowledgment

EURECOM's research is partially supported by its industrial members: ORANGE, BMW, SAP, iABG, Norton LifeLock, and by the Franco-German projects 5G-OPERA and Cell-Free6G.

References

[1] D.L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] J.-Y. Tourneret A. O. Hero C. Bazot, N. Dobigeon, "A Bernoulli-Gaussian model for gene factor analysis," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5996–5999.

[3] A. Montanari D. L. Donoho, A. Maleki, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18,914–18,919, Nov. 2009.

[4] A. Montanari M. Bayati, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.

[5] T. Kawabata K. Takeuchi, T. Tanaka, "Performance improvement of iterative multiuser detection for large sparsely-spread CDMA systems by spatial coupling," *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1768–1794, Apr. 2015.

[6] A. Fletcher S. Rangan, P. Schniter, "On the convergence of approximate message passing with arbitrary matrices," in *Proc. 2014 IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, July 2014, pp. 236–240.

[7] T. P. Minka, "Expectation propagation for approximate Bayesian inference," *arXiv preprint*, 2013.

[8] M. Sánchez-Fernández F. Perez-Cruz J. Cespedes, P. M. Olmos, "Expectation propagation detection for high-order high-dimensional MIMO systems," *IEEE Trans. Commun.*, vol. 62, no. 8, pp. 2840–2849, Aug. 2014.

[9] K. Takeuchi, "Rigorous dynamics of expectation-propagation-based signal recovery from unitarily invariant measurements," in *IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 501–505.

[10] B. A. Satheesh, B. Deepa, S. Bhai, and S. A. Devi, "Compressive sensing for array signal processing," in *IEEE India Conf. (INDICON)*, 2012, pp. 555–560.

[11] F. Gini M. S. Greco S. Fortunati, R. Grasso, "Single snapshot DOA estimation using compressed sensing," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 2297–2301.

[12] P. Steeghs R. Baraniuk, "Compressive Radar Imaging," in *IEEE Radar Conf.*, 2007, pp. 128–133.

[13] J. Chun and T. Kailath, "A constructive proof of the Gohberg-Semencul formula," *Linear Algebra and its Applications*, vol. 121, pp. 475–489, 1989.

[14] P. Castiglioni, "Levinson-Durbin algorithm," *Encyclopedia of Biostatistics*, vol. 4, 2005.

[15] T. M. Peters and J. H. T. Bates, *The Discrete Fourier Transform and the Fast Fourier Transform*, pp. 175–194, Birkhäuser Boston, Boston, MA, 1998.

[16] T. L. Hansen, B. H. Fleury, and B. D. Rao, "Superfast line spectral estimation," *IEEE Trans. Signal Processing*, vol. 66, no. 10, pp. 2511–2526, 2018.

[17] A. K. Fletcher S. Rangan, P. Schniter, "Vector approximate message passing," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6664–6684, 2019.

[18] V. Balakrishnan, "All about the Dirac delta function ," *Resonance*, vol. 8, no. 8, pp. 48–58, 2003.

[19] J. Vila and P. Schniter, "Expectation-maximization Bernoulli-Gaussian approximate message passing," in *Conf. Record of the 45th Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2011, pp. 799–803.

[20] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," Tech. Rep. 7.15, Technical University of Denmark, 2008.