Turbo Compressed Sensing with Partial DFT Sensing Matrix

Junjie Ma, Xiaojun Yuan, *Member, IEEE*, and Li Ping, *Fellow, IEEE*

***Abstract*—In this letter, we propose a turbo compressed sensing algorithm with partial discrete Fourier transform (DFT) sensing matrices. Interestingly, the state evolution of the proposed algorithm is shown to be consistent with that derived using the replica method. Numerical results demonstrate that the proposed algorithm outperforms the well-known approximate message passing (AMP) algorithm when a partial DFT sensing matrix is involved.**

***Index Terms*—Approximate message passing (AMP), compressed sensing, partial DFT matrix, replica method, state evolution.**

I. Introduction

PARTIAL discrete Fourier transform (DFT) sensing matrices have found many applications [1] and an efficient signal recovery algorithm is highly desirable for related compressed sensing problems. *Approximate message passing* (AMP) [2]–[4] is an iterative algorithm for this purpose. The state evolution of AMP with independent and identically distributed (i.i.d.) Gaussian sensing matrices is shown to be consistent with that derived using the replica method [4]. This implies that AMP can potentially provide near-optimal performance when i.i.d. Gaussian sensing matrices are involved. However, the situation is different for partial DFT sensing matrices whose entries are not independently drawn. Recent results in [5], [6] pointed out that, using the replica method, the optimal reconstruction performance of a system based on a partial DFT matrix is different from that based on an i.i.d. Gaussian matrix.

In this letter, we propose a turbo-type iterative algorithm [7] for the problem. The proposed algorithm involves two local processors. One processor handles the information related to a partial DFT sensing matrix using the linear minimum mean-square error (LMMSE) principle. The other processor handles the sparsity information. Our main contribution is a novel way to compute extrinsic messages related to the sparsity information. The state evolution of the proposed algorithm coincides with that predicted by the replica method [5], [6]. This indicates the potentially excellent performance of the proposed algorithm, as confirmed by Monte Carlo simulations.

II. Problem Description

Consider the following linear system

$$\boldsymbol{y} = \boldsymbol{F}_{\text{partial}}\boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{C}^{N\times 1}$ is a sparse signal to be estimated, $\boldsymbol{y} \in \mathbb{C}^{M\times 1}$ the received signal, and $\boldsymbol{n} \sim \mathcal{CN}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ the Gaussian noise. $\boldsymbol{F}_{\text{partial}}$ consists of M randomly selected and reordered rows of the unitary DFT matrix $\boldsymbol{F} \in \mathbb{C}^{N\times N}$, where the (m,n)th entry of $\boldsymbol{F}$ is given by $\frac{1}{\sqrt{N}}e^{-2\pi j(m-1)(n-1)/N}$ with $j = \sqrt{-1}$. The entries of the sparse signal $\boldsymbol{x}$ is assumed to be i.i.d., with the jth entry of $\boldsymbol{x}$ following the Bernoulli-Gaussian distribution [4]:

$$x_j \sim \begin{cases} 0 & \text{probability} = 1-\lambda, \\ \mathcal{CN}(0,\lambda^{-1}) & \text{probability} = \lambda. \end{cases} \tag{2}$$

In (2), the variance of each x_j is normalized, i.e., $\mathrm{E}[|x_j|^2] = 1$.

The partial DFT matrix in (1) can be rewritten as

$$\boldsymbol{F}_{\text{partial}} = \boldsymbol{SF}, \tag{3}$$

where $\boldsymbol{S}$ is a *selection* matrix consisting of M randomly selected and reordered rows of the $N \times N$ identity matrix. Define

$$\boldsymbol{z} = \boldsymbol{Fx}. \tag{4}$$

Together with (3), we rewrite the system model in (1) as

$$\boldsymbol{y} = \boldsymbol{Sz} + \boldsymbol{n}. \tag{5}$$

A. Standard Turbo Algorithm

The proposed algorithm is based on the turbo principle in iterative decoding [7]. Before introducing the proposed solution, we will first discuss a standard algorithm and explain its potential problem.

For the problem in (1), the block diagram of a standard turbo detector is illustrated in Fig. 1(a). It consists of two modules: module A is an LMMSE estimator and module B a sparsity combiner. The LMMSE estimator produces a coarse extrinsic estimate of $\boldsymbol{x}$ based on the observation $\boldsymbol{y}$. The sparsity combiner refines the estimate using the sparse distribution in (2). Here, the extrinsic output [7] of a module is fed to the other module as the *a priori* input. The two modules are executed iteratively until convergence. At the end of the iteration, the final estimate of $\boldsymbol{x}$ is based on the *a posteriori* output of the sparsity combiner. We next discuss the detailed operations of the algorithm in Fig. 1(a).

Manuscript received May 18, 2014; accepted August 15, 2014. Date of publication August 26, 2014; date of current version September 08, 2014. The work of J. Ma and L. Ping were supported by the Research Grant Council, Hong Kong, under Project CityU 118013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Alex Dimakis

J. Ma and L. Ping are with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, SAR, China (e-mail: junjiema2-c@my.cityu.edu.hk; eeliping@cityu.edu.hk).

X. Yuan is with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China (e-mail: yuanxj@shanghaitech.edu.cn).

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

Digital Object Identifier 10.1109/LSP.2014.2351822

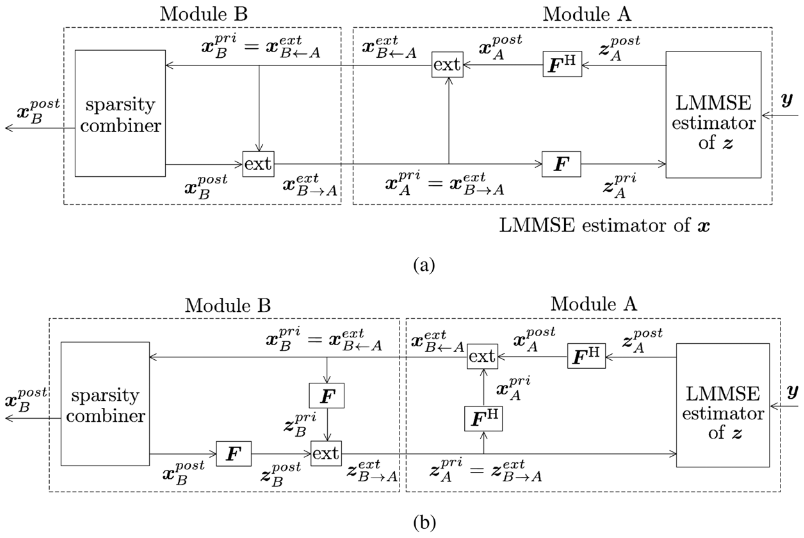

Fig. 1. Block diagrams of a standard turbo algorithm and the proposed turbo algorithm. “ext” represents extrinsic information computation. (a) A standard turbo detector (b) The proposed turbo detector.

Module A:

Assumption 1: The entries of $\boldsymbol{x}$ are i.i.d. with *a priori* mean $\boldsymbol{x}_A^{pri}$ and variance v_A^{pri}.

The *a priori* information about $\boldsymbol{x}$ is obtained from the feedback of the sparsity combiner, which will be discussed later. With Assumption 1, the *a priori* mean of $\boldsymbol{z} = \boldsymbol{F}\boldsymbol{x}$ is given by

$$\boldsymbol{z}_A^{pri} = \boldsymbol{F}\boldsymbol{x}_A^{pri} \tag{6}$$

and the variance is v_A^{pri}. From (5), the LMMSE estimator and the mean-square error (MSE) matrix of $\boldsymbol{z}$ are respectively given by [8]

$$\boldsymbol{z}_A^{post} = \boldsymbol{z}_A^{pri} + \frac{v_A^{pri}}{v_A^{pri} + \sigma^2}\boldsymbol{S}^{\mathrm{H}}\left(\boldsymbol{y} - \boldsymbol{S}\boldsymbol{z}_A^{pri}\right), \tag{7a}$$

$$\boldsymbol{V}_A^{post} = v_A^{pri} \cdot \boldsymbol{I}_N - \frac{\left(v_A^{pri}\right)^2}{v_A^{pri} + \sigma^2}\boldsymbol{S}^{\mathrm{H}}\boldsymbol{S}. \tag{7b}$$

From $\boldsymbol{x} = \boldsymbol{F}^{\mathrm{H}}\boldsymbol{z}$, the LMMSE estimator of $\boldsymbol{x}$ is

$$\boldsymbol{x}_A^{post} = \boldsymbol{F}^{\mathrm{H}}\boldsymbol{z}_A^{post}. \tag{8}$$

The associated MSE matrix is

$$\boldsymbol{F}^{\mathrm{H}}\boldsymbol{V}_A^{post}\boldsymbol{F}. \tag{9}$$

It can be verified that the diagonals of $\boldsymbol{F}^{\mathrm{H}}\boldsymbol{V}_A^{post}\boldsymbol{F}$, which are the *a posteriori* MSEs, are identical and given by

$$v_A^{post} = v_A^{pri} - \frac{M}{N} \cdot \frac{\left(v_A^{pri}\right)^2}{v_A^{pri} + \sigma^2}. \tag{10}$$

Using the concise formulas in [9], [10], the extrinsic LMMSE estimate and the MSE of $\boldsymbol{x}$ can be computed by

$$\boldsymbol{x}_{B\leftarrow A}^{ext} = v_{B\leftarrow A}^{ext}\left(\frac{\boldsymbol{x}_A^{post}}{v_A^{post}} - \frac{\boldsymbol{x}_A^{pri}}{v_A^{pri}}\right), \tag{11a}$$

$$v_{B\leftarrow A}^{ext} = \left(\frac{1}{v_A^{post}} - \frac{1}{v_A^{pri}}\right)^{-1}. \tag{11b}$$

Module B: The LMMSE estimator effectively makes a Gaussian assumption on $\boldsymbol{x}$ and ignores the sparsity information of $\boldsymbol{x}$. The function of the sparsity combiner is to refine the LMMSE estimate of $\boldsymbol{x}$ by combining the sparsity information in (2).

Assumption 2: $\boldsymbol{x}_B^{pri}$ is modeled as an additive white Gaussian noise (AWGN) observation of $\boldsymbol{x}$, i.e.,

$$\boldsymbol{x}_B^{pri} = \boldsymbol{x} + \boldsymbol{w}, \tag{12}$$

where $\boldsymbol{w} \sim \mathcal{CN}(\boldsymbol{0}, v_B^{pri}\boldsymbol{I})$ and is independent of $\boldsymbol{x}$. Here, $\boldsymbol{x}_B^{pri}$ and v_B^{pri} are updated by the extrinsic output of module A, i.e.

$$\boldsymbol{x}_B^{pri} = \boldsymbol{x}_{B\leftarrow A}^{ext} \text{ and } v_B^{pri} = v_{B\leftarrow A}^{ext}. \tag{13}$$

Based on Assumption 2, the minimum mean-square error (MMSE) estimator of $\boldsymbol{x}$ conditioned on $\boldsymbol{x}_B^{pri}$ is a component-wise operation and given by

$$x_{j,B}^{post} = \mathrm{E}\left[x_j|\boldsymbol{x}_B^{pri}\right] = \mathrm{E}\left[x_j|x_{j,B}^{pri}\right], \forall j, \tag{14}$$

where $x_{j,B}^{post}$ and $x_{j,B}^{pri}$ denote the jth entry of $\boldsymbol{x}_B^{post}$ and $\boldsymbol{x}_B^{pri}$ respectively. $\mathrm{E}[\cdot]$ is with respect to the joint distribution of $\boldsymbol{x}$ and $\boldsymbol{x}_B^{pri}$ characterized by (12). The detailed operations of the above MMSE estimation can be found in, e.g., [6]. The conditional variance corresponding to (14) is given by

$$v_{j,B}^{post} = \mathrm{var}\left[x_j|\boldsymbol{x}_B^{pri}\right] = \mathrm{var}\left[x_j|x_{j,B}^{pri}\right], \forall j, \tag{15}$$

where $\mathrm{var}[a|b] \equiv \mathrm{E}[|a - \mathrm{E}[a|b]|^2|b]$.

We next compute the extrinsic estimate of each x_j by excluding the contribution of $x_{j,B}^{pri}$. Under Assumption 2, the MMSE estimation in (14) is a component-wise operation. Excluding the contribution of x_j, the extrinsic estimate of x_j becomes

$$x_{j,B\to A}^{ext} = \mathrm{E}[x_j|\boldsymbol{x}_{\sim j,B}^{pri}] = \mathrm{E}\left[x_j\right] = 0, \forall j, \tag{16}$$

where $\boldsymbol{x}_{\sim j,B}^{pri}$ is obtained from $\boldsymbol{x}_B^{pri}$ by excluding the jth entry $x_{j,B}^{pri}$. The extrinsic estimate of module B will be treated as *a priori* mean for module A in the next iteration.

The following observations are useful:

- The LMMSE operation ensures that module A in Fig. 1(a) is optimal (in the LMMSE sense) if the sparsity information is ignored and no iteration is involved. Note that AMP cannot make such a claim due to the distributive nature of message passing.

- However, from (16), the extrinsic estimate of module B is zero and so iterative processing does not provide any further improvement.

In what follows, we will develop an alterative processor in Fig. 1(b) that maintains the advantage but avoids the disadvantage.

B. Proposed Turbo Compressed Sensing Algorithm

The proposed algorithm is illustrated in Fig. 1(b). Module A computes extrinsic information of $\boldsymbol{x}$ and Module B computes the extrinsic information of $\boldsymbol{z}$. This is different from the standard approach in Fig. 1(a) where both modules compute extrinsic information of the same variable $\boldsymbol{x}$.

Module A: Module A includes the LMMSE estimator of $\boldsymbol{z}$ and two IDFTs. The operations of Module A are roughly the same as that in Fig. 1(a), except that the input is $\boldsymbol{z}_A^{pri}$ in Fig. 1(b).

Module B: As discussed in Section II-A, the sparsity combiner produces no extrinsic estimate of $\boldsymbol{x}$. In the proposed algorithm, module B now computes the extrinsic estimate of $\boldsymbol{z}$ instead of $\boldsymbol{x}$.

Assumption 3: The *a posteriori* distributions of $\boldsymbol{z}$ conditioned on $\boldsymbol{z}_B^{pri}$ are Gaussian, i.e.[1]

$$\Pr\left(z_j|\boldsymbol{z}_B^{pri}\right) = \mathcal{CN}\left(z_{j,B}^{post}, v_B^{post}\right)^{1}, \ \forall j \tag{17}$$

where $z_{j,B}^{post}$ is the jth entry of the following *a posteriori* mean vector

$$\boldsymbol{z}_B^{post} = \boldsymbol{F}\boldsymbol{x}_B^{post}, \tag{18}$$

and v_B^{post} is the *a posteriori* variance given by

$$v_B^{post} = \frac{1}{N}\sum_{j=1}^{N} v_{j,B}^{post}, \tag{19}$$

where $v_{j,B}^{post}$ is the variance of $x_{j,B}^{post}$ in (15).

Intuitively, when N is large, Assumption 3 can be justified by the mixing effect of the DFT and the central limit theorem. Eqn. (18) is due to $\boldsymbol{z} = \boldsymbol{F}\boldsymbol{x}$. As the entries of $\boldsymbol{x}$ are *a priori* independent (from Assumption 2) and the sparsity combiner is a component-wise operation, the entries of $\boldsymbol{x}$ are also *a posteriori* independent, and so (19) follows.

From Assumption 2, the *a priori* estimate $\boldsymbol{z}_B^{pri} = \boldsymbol{F}\boldsymbol{x}_B^{pri}$ is an AWGN observation of $\boldsymbol{z}$, i.e.

$$\boldsymbol{z}_B^{pri} = \boldsymbol{F}\boldsymbol{x}_B^{pri} = \boldsymbol{z} + \boldsymbol{F}\boldsymbol{w}. \tag{20}$$

As $\boldsymbol{w}$ is i.i.d. Gaussian with mean zero and variance v_B^{pri}, $\boldsymbol{F}\boldsymbol{w}$ has the same distribution and we have

$$\Pr\left(z_{j,B}^{pri}|z_j\right) = \mathcal{CN}\left(z_{j,B}^{pri}, v_B^{pri}\right), \ \forall j. \tag{21}$$

From (20), $z_{j,B}^{pri}$ and $\boldsymbol{z}_{\sim j,B}^{pri}$ are conditionally independent given z_j. It can then be verified that

$$\Pr\left(z_j|\boldsymbol{z}_B^{pri}\right) \propto \Pr\left(z_j|\boldsymbol{z}_{\sim j,B}^{pri}\right)\cdot\Pr\left(z_{j,B}^{pri}|z_j\right), \ \forall j \tag{22}$$

where $\propto$ denotes equality up to a constant scaling factor independent of z_j. Based on (17), (21) and (22), the extrinsic distribution $\Pr(z_j|\boldsymbol{z}_{\sim j,B}^{pri})$ is Gaussian [9], [11] and given by

$$\Pr\left(z_j|\boldsymbol{z}_{\sim j,B}^{pri}\right) = \mathcal{CN}\left(z_{j,B\to A}^{ext}, v_{B\to A}^{ext}\right), \ \forall j \tag{23}$$

where

$$z_{j,B\to A}^{ext} = v_{B\to A}^{ext}\left(\frac{z_{j,B}^{post}}{v_B^{post}} - \frac{z_{j,B}^{pri}}{v_B^{pri}}\right), \tag{24a}$$

$$v_{B\to A}^{ext} = \left(\frac{1}{v_B^{post}} - \frac{1}{v_B^{pri}}\right)^{-1}. \tag{24b}$$

The extrinsic mean/variance in (24) will be treated as *a priori* mean/variance for module A in the next iteration. Note that in the standard turbo detector in Section II-A, module B produces no extrinsic output, as shown in (16). This is the main difference between the proposed algorithm and the standard detector.

Overall Algorithm: In the first iteration, $\boldsymbol{z}_A^{pri} = \boldsymbol{0}$ and $v_A^{pri} = 1$. The operations of module A and module B are executed iteratively until convergence.

The DFT/IDFT operations in Fig. 1(b) can be efficiently implemented using the fast Fourier transform (FFT). Also, the order of the "ext" operations in Fig. 1(b) (see (11) and (24)) and DFT/IDFT can be changed, and then one pair of DFT/IDFT can be saved. This is straightforward and we omit the details.

III. State Evolution

Following [2]–[4], we analyze the large-system performance of the proposed scheme by using state evolution.

A. State Evolution

We characterize the performance of the iterative algorithm by a recursion of two states, v_A^{pri} and v_B^{pri}. In the following, for notational brevity, we define

$$\eta \equiv \frac{1}{v_B^{pri}} \text{ and } v \equiv v_A^{pri}. \tag{25}$$

We define the following MMSE of the sparse signal estimation given an AWGN observation (with SNR η)

$$mmse(\eta) \equiv \mathrm{E}\left[|x - \mathrm{E}[x|x+\xi]|^2\right], \tag{26}$$

where x is a sparse signal modeled as (2) and $\xi \sim \mathcal{CN}(0, \eta^{-1})$. From (15) and based on Assumption 2,

$$v_B^{post} = \frac{1}{N}\sum_{j=1}^{N}\mathrm{var}\left[x_j|x_{j,B}^{post}\right] \to mmse(\eta). \tag{27}$$

Under Assumptions 1-3, we have the following proposition.

Proposition 1: The state evolution of the proposed turbo compressed sensing algorithm is characterized by

$$\eta_{t+1} = \frac{1}{\frac{N}{M}\cdot(v_t + \sigma^2) - v_t}, \tag{28a}$$

$$\frac{1}{v_{t+1}} = \frac{1}{mmse(\eta_{t+1})} - \eta_{t+1}, \tag{28b}$$

where the subscript t and $t+1$ indicate the iteration indices. The state evolution in (28) is derived by combining (10), (11b),

[1] With slight abuse of notation, here (and also in (21)) $\mathcal{CN}(m, v)$ denotes a Gaussian *function* of z_j with mean m and variance v.

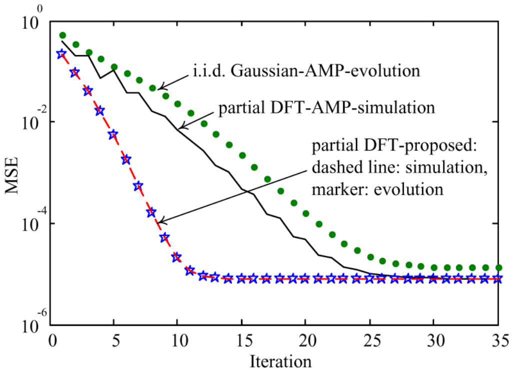

Fig. 2. Comparisons of the proposed algorithm and AMP. $N = 8192$, $M = 5734 (\approx 0.7N)$, $\lambda = 0.4$, and $\mathrm{SNR} = 50$ dB.

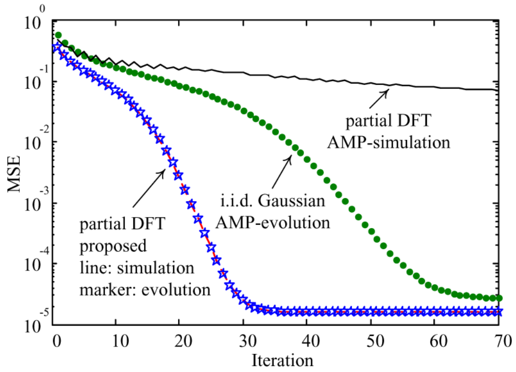

Fig. 3. Comparisons of the proposed algorithm and AMP. $N = 32768$, $M = 18022 (\approx 0.55N)$, $\lambda = 0.4$, and $\mathrm{SNR} = 50$ dB.

(15), (19), (24b), (25)–(27), together with some straightforward manipulations.

B. Fixed Point of State Evolution

Denote by $\eta_\star$ the convergence value of η. Combining (28a) and (28b) and eliminating v, $\eta_\star$ can be characterized by the following fixed point equation:

$$\frac{N}{M} \cdot \sigma^2 \cdot mmse(\eta_\star) \cdot \eta_\star^2 - \frac{N}{M} \cdot \left(mmse(\eta_\star) + \sigma^2 \right) \cdot \eta_\star + 1 = 0. \tag{29}$$

One solution of (29) is given by

$$\eta_\star = \frac{\left(mmse + \sigma^2 \right) - \sqrt{\left(mmse + \sigma^2 \right)^2 - 4\sigma^2 \cdot mmse \cdot \frac{M}{N}}}{2 \cdot \sigma^2 \cdot mmse}, \tag{30}$$

where $mmse$ represents $mmse(\eta_\star)$. Note that (29) has two solutions, but it can be shown that the other solution is not a valid convergence point.

It can be verified that (30) is consistent with that in [5, (17) and (37)] derived using the replica method. It can also be shown that (28) is equivalent to [6, (17)–(18)]. We omit the details here due to space limitation.

IV. Numerical Examples

In Fig. 2, we compare the MSE performance of the proposed algorithm with partial DFT matrices and AMP with i.i.d. Gaussian matrices. For a fair comparison, the variance of each entry in the i.i.d. Gaussian matrix is normalized to $1/N$. Here, the implementation of AMP is based on [4]. In simulation, MSE is obtained by averaging over 2000 realizations. We see that the proposed algorithm converges faster than AMP and also achieves lower convergence MSE. Moreover, the state evolution analysis agrees well with simulation. We can also directly apply AMP to the case with a partial DFT sensing matrix. From Fig. 2, we see that AMP with partial DFT matrices outperforms that with i.i.d. Gaussian matrices. This performance difference also indicates that the state evolution of AMP is not accurate when applied to partial DFT matrices. This is reasonable because the state evolution of AMP is developed for i.i.d. Gaussian matrices.

In Fig. 3, we reduce the measurement ratio M/N. (N is set to sufficiently large to ensure that the simulation performance agrees well with state evolution.) We see that AMP for partial DFT performs much worse than the proposed algorithm in this setup.

V. Conclusion and Discussions

The state evolution in Section III-A is developed based on three assumptions. Numerical results in Section IV demonstrate that the state evolution developed based on these assumptions is reasonably accurate. It is an interesting future research topic to establish more rigorous justifications for the state evolution. The analysis in [3] for AMP with i.i.d. Gaussian sensing matrices may shed light on this problem.

References

[1] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

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

[3] M. Bayati and A. Montanari, “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.

[4] S. Rangan, “Generalized approximate message passing for estimation with random linear mixing,” 2010 [Online]. Available: http://arxiv.org/abs/1010.5141, preprint

[5] A. Tulino, G. Caire, S. Verdu, and S. Shamai, “Support recovery with sparsely sampled free random matrices,” *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4243–4271, Jul. 2013.

[6] C. K. Wen and K. K. Wong, “Analysis of compressed sensing with spatially-coupled orthogonal matrices,” 2014 [Online]. Available: http://arxiv.org/abs/1402.3215, preprint

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

[8] S. M. Kay*, Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall PTR, 1993.

[9] Q. Guo and D. Huang, “A concise representation for the soft-in soft-out lmmse detector,” *IEEE Commun. Lett.*, vol. 15, no. 5, pp. 566–568, May 2011.

[10] X. Yuan, L. Ping, C. Xu, and A. Kavcic, “Achievable rates of mimo systems with linear precoding and iterative lmmse detection,” 2011 [Online]. Available: http://arxiv.org/ftp/arxiv/papers/1106/1106.0178.pdf, preprint

[11] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. Kschischang, “The factor graph approach to model-based signal processing,” *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, Jun. 2007.