Analysis for sensing resource reduction via state evolution

The Thesis Submitted to the Electrical and Electronic Engineering Department

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Yang Lu

Imperial College London

United Kingdom

2nd January 2018

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

I hereby declare that to the best of my knowledge, this thesis is my own work and all else sources of contents have been acknowledged and appropriately referenced.

– Yang Lu

Abstract

This thesis focuses on the approximate message passing (AMP) based algorithms for solving compressed sensing problems and provides corresponding modifications and state evolution analyses based on the following situations.

We consider the correlated distributed compressed sensing (C-DCS) model, in which multiple measurement instances are included. This model allows correlation between measurement matrices and signals across different measurement instances. We modified the AMP algorithm for the C-DCS model such that it can handle correlated matrices and correlated signals. Correctness justification is provided for our proposed algorithm for two special cases: distributed compressed sensing (DCS) and multiple measurement vectors (MMV) models. Simulations show that the empirical results almost perfectly match the theoretical predictions achieved by state evolution.

We consider a practical signal transmission/receiving application with fixed energy budget and assume that the thermal noise is the dominant noise source. Under such conditions, we observe that the overall signal-to-noise ratio (SNR) per measurement decreases quadratically with the increase of the number of measurements. By applying the AMP algorithm and state evolution analysis, we are able to provide an optimal number of measurements to minimize the mean squared error of the estimate which is different from the common wisdom

where more measurements often mean a better performance. Numerical results justify the correctness of our analysis.

The performance of AMP may severely deteriorate when the measurement matrix is not a standard Gaussian random matrix. We propose an improved AMP (IAMP) algorithm that works better for non i.i.d. Gaussian random matrices when the correlations between elements of the measurement matrix deviate from those of the standard Gaussian. The derivation is based on a modification of the message passing mechanism that removes the conditional independence assumption. Examples are provided to demonstrate the performance improvement of IAMP where both a particularly designed matrix and a matrix from real applications are used.

Acknowledgement

First and foremost, I would like to express my sincere gratitude and appreciation to my PhD supervisor, Dr. Wei Dai. I have been so lucky to have a supervisor who was always by my side and willing to give help. During the past four years, he provided me with his professional guidance, selfless support, earnest encouragement and funded me to participate in various international conferences.

I would like to thank Dr. Cong Ling who is a senior lecturer in the Department of Electrical and Electronic Engineering at Imperial College London for his valuable and constructive comments and suggestions during my early stage assessment and late stage review.

I must express appreciation to Dr. Deniz Gunduz from Imperial College London and Dr. Ramji Venkataramanan from the University of Cambridge for finding time from their busy schedules to go through my thesis and serve as my internal and external examiners.

I would like to thank my friends, especially Dr. Guangyu Zhou and Dr. Xiaochen Zhao, who were the former students of Dr. Wei Dai, for their kind help and advice in both life and academia.

Finally, I would like to acknowledge my parents, who shared all the ups and downs of my entire PhD study career, for their continued support and

encouragement in life. Without their trust, support and love, I would not have finished this thesis.

Contents

List of Publications

List of publications arising directly from this thesis

1. Yang Lu and Wei Dai. "Improved AMP (IAMP) for Non-ideal Measurement Matrices", IEEE International Conference on European Signal Processing (EUSIPCO), 2015.

2. Yang Lu and Wei Dai. "Independent versus Repeated Measurements: A Performance Quantification via State Evolution". IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016.

3. Yang Lu and Wei Dai. "Extended AMP Algorithm for Correlated Distributed Compressed Sensing Model". IEEE International Conference on Digital Signal Processing (DSP), 2016

4. Yang Lu, Wei Dai and Yonina C. Eldar. "Optimal Number of Measurements for Compressed Sensing with Quadratically Decreasing SNR". Submitted to IEEE International Conference on European Signal Processing (EUSIPCO), 2017.

Other publications

1. Wenbo Ding, Yang Lu, Fang Yang, Wei Dai and Jian Song. "Sparse Channel State Information Acquisition for Power Line Communications", IEEE International Conference on Communications (ICC), 2015.

2. Ji Wu, Yang Lu and Wei Dai. "Off-grid Compressed Sensing for WiFi-based Passive Radar", IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2016.

3. Wenbo Ding, Yang Lu, Fang Yang, Wei Dai, Pan Li, Sicong Liu and Jian Song. "Spectrally Efficient CSI Acquisition for Power Line Communications: A Bayesian Compressive Sensing Perspective", IEEE Journal On Selected Areas in Communications, 2016.

List of Figures

List of Algorithms

Nomenclature

ADMM	Alternating Direction Method of Multipliers
AMP	Approximate Message Passing
AWGN	Additive White Gaussian Noise
BG	Bernoulli-Gaussian
BP	Basis Pursuit
BPDN	Basis Pursuit Denoising
C-DCS	Correlated-Distributed Compressed Sensing
CoSaMP	Compressive Sensing Matching Pursuit
CS	Compressed Sensing (Compressive Sensing)
CT	Computerized Tomography
DC-OMP	Distributed and Collaborative Orthogonal Matching Pursuit
DCS	Distributed Compressed Sensing
DCT	Discrete Cosines Transform
DFT	Discrete Fourier Transform

DiSP	Distributed Subspace Pursuit
DST	Discrete Sines Transform
DWT	Discrete Wavelet Transform
ECG	Electrocardiogram
EM	Expectation-Maximization
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Technique
FISTA	Fast Iterative Shrinkage-Thresholding Algorithm
GAMP	Generalized Approximate Message Passing
IAMP	Improved Approximate Message Passing
IST	Iterative Soft-thresholding
JPEG	Joint Photographic Experts Group
LASSO	Least Absolute Shrinkage and Selection Operator
LF	Least-Favoourite
LP	Linear Programming
M-FOCUSS	MMV Focal Under-determined System Solution
MMI	Multiple Measurement Instances
MMSE	Minimum Mean Squared Error
MMV	Multiple Measurement Vectors

MP	Message Passing
MP3	MPEG-1 or MPEG-2 Audio Layer III
MPEG	Moving Picture Experts Group
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
MUSIC	Multiple Signal Classification
OMP	Orthogonal Matching Pursuit
QP	Quadratic Programming
RADAR	Compressive Radio Detecting and Ranging
RIP	Restricted Isometry Property
SA-MUSIC	Subspace Augmented Multiple Signal Classifier
SE	State Evolution
SNR	Signal-to-Noise Ratio
SP	Subspace Pursuit
StOMP	Stagewise Orthogonal Matching Pursuit
SVD	Singular Value Decomposition
WMI	Woodbury Matrix Identity
WSNs	Wireless Sensor Networks

Chapter 1

Introduction

1.1 Motivation of Compressed Sensing

In the traditional digital signal processing field, the Nyquist sampling theorem plays as the fundamental role which says in order to capture all the information of a finite bandwidth continuous-time signal, the minimum sampling rate must at least twice ($\times 2$) the highest frequency component of the continuous-time signal. Otherwise, aliasing will occur when converting the digital sequence back to the continuous-time domain. Similarly, in linear algebra, the fundamental rule tells us that at least n independent measurements is required in order to ensure the reconstruction of an n-dimensional signal. Otherwise, the solution is not unique. These principles underlie most devices of current technology, such as analogue to digital conversion and medical imaging processing [2]. Compressed sensing (CS) is a novel theory which was introduced in [3, 4], providing a new data acquisition approach which breaks the limitations of above principles under the assumption that the original signal has a sparse representation in some transform domain.

Traditional compression techniques such as JPEG and MP3, firstly, require a fully sampled sequence, then approximate the signal by only storing a small set of the largest basis coefficients (e.g. in Fourier domain or Wavelet domain) while setting other basis coefficients to zero. Ignoring small basis coefficients will lose some information, but the compressed signal is still a good approximation of the original signal with a significantly reduced file size. The drawback is that the fully sampled sequence should obey the Nyquist sampling theorem in order to acquire full information which is sometimes a costly and difficult measurement procedure. But at the same time, a lot of the (negligible) information will eventually be thrown away in the latter compression process, this seems to be a waste of resources [2]. CS, alternatively, directly embeds the compression process in the sampling stage by using a small number of measurements to acquire the maximum amount of information from the signal [5].

1.2 Mathematical Model of Compressed Sensing

In CS, the problem is usually mathematically represented by the following linear system:

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\alpha} + \boldsymbol{w}, \tag{1.2.1}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ represents the observation, $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$ $(m < n)$ denotes the sensing matrix, $\boldsymbol{\alpha} \in \mathbb{R}^n$ is the unknown signal and $\boldsymbol{w} \in \mathbb{R}^m$ is the additive noise. In addition, the signal $\boldsymbol{\alpha}$ is assumed to have a sparse representation in

certain basis $\boldsymbol{\Psi}$, mathematically say

$$\boldsymbol{\alpha} = \boldsymbol{\Psi}\boldsymbol{x},$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the S-sparse representation of $\boldsymbol{\alpha}$ in the basis of $\boldsymbol{\Psi} \in \mathbb{R}^{n\times n}$. We call a signal S-sparse when at most S ($\ll n$) of its coefficients are non-zero, i.e.

$$\|\boldsymbol{x}\|_0 \leq S,$$

where $\|\cdot\|_0$ denotes the ℓ_0-pseudo norm which counts the number of non-zero elements of $\boldsymbol{x}$. The sparse representation assumption is reasonable because a lot of natural signals and synthetic signals inherently have sparse representations [6, 2], such as electrocardiogram (ECG) signals [7, 8], audio/music signals [9] as well as image/video signals [10]. The corresponding basis might be discrete Fourier transform (DFT) matrices, discrete cosines transform (DCT) matrices, discrete sines transform (DST) matrices, Haar transform matrices or discrete wavelet transform (DWT) matrices, to name a few.

Let $\boldsymbol{A} \coloneqq \boldsymbol{\Phi}\boldsymbol{\Psi}$, one can rewrite (1.2.1) as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}. \tag{1.2.2}$$

In the rest of the thesis, the problem formula and analysis mainly focus on (1.2.2). See Fig 1.2.1 for the intuitive representation.

The initial attempt to solve (1.2.2) for the noise free case is via ℓ_0-minimization:

$$\min_x \|\boldsymbol{x}\|_0 \text{ s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y},$$

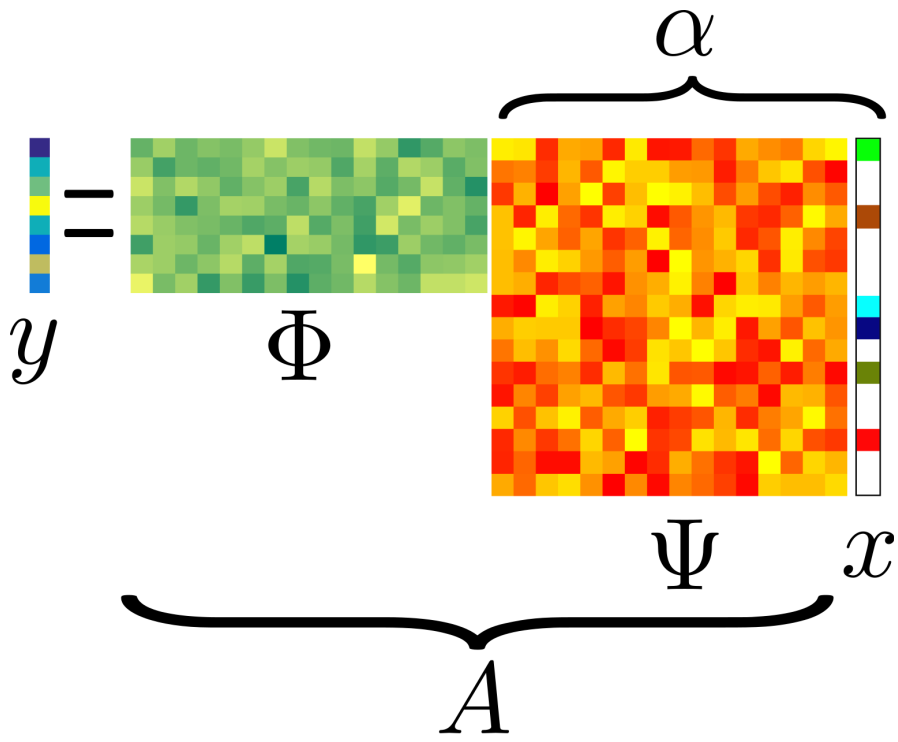


Figure 1.2.1: Sparse representation (without noise). $\boldsymbol{y}$ represents the observation, $\boldsymbol{\Phi}$ is the sensing matrix, $\boldsymbol{x}$ denotes the sparse representation in the basis $\boldsymbol{\Psi}$ and the sparsity $S = 6$.

which, unfortunately, is an NP-hard problem thus there is no computationally tractable algorithm currently exist to efficiently solve it. Alternative approaches should be tried either by relaxing the problem or seeking an approximate answer.

1.3 Algorithms for Sparse Recovery

A number of reconstruction algorithms have been proposed during the recent decades especially after [3, 4, 11] had been published around 2005. Most of these algorithms can be categorized into three major classes as follows. (A more detailed classification can be found in [5].)

1. Convex Relaxation

   Algorithms in this class treat the reconstruction task as a convex optimization problem through linear programming (LP) [12] or quadratic programming (QP) by replacing the ℓ_0-norm with the ℓ_1-norm. The representative algorithms include basis pursuit (BP), BP denoising (BPDN) [13] and least absolute shrinkage and selection operator (LASSO) [14] and their corresponding mathematical formulas are

$$\begin{aligned}
\text{BP}: &\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{ s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}, \\
\text{BPDN}: &\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{ s.t. } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \leq \epsilon_e, \\
\text{LASSO}: &\min_{\boldsymbol{x}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 \text{ s.t. } \|\boldsymbol{x}\|_1 \leq S,
\end{aligned}$$

   where ϵ_e is a small positive value and S represents the number of non-zero elements of $\boldsymbol{x}$. BP is suitable for noise free case while BPDN and LASSO are suitable for the noisy case. BPDN and LASSO are equivalent

problems (for certain corresponding values of S and λ) both of which can be solved via

$$\text{BPDN/LASSO} : \min_{\boldsymbol{x}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 , \tag{1.3.1}$$

where λ is a penalty parameter which controls the trade-off between reconstruction fidelity and the sparsity level.

Other algorithms include least angle regression (LARS) [15] and Dantzig selector [16], to name a few. The estimation performance achieved through convex optimization usually is quite good (e.g. few measurements required and low mean squared error (MSE)) at the cost of a relatively high computational complexity compared with greedy algorithms.

2. Greedy Algorithms

As its name, greedy algorithms solve the reconstruction problem iteration by iteration. At each iteration, a local optimum is achieved by minimizing a least squared error problem related with observation $\boldsymbol{y}$ and the information of selected columns of $\boldsymbol{A}$ will be updated for the next iteration. The process will continue until meeting some stopping criteria, such as the maximum number of iterations, or the MSE of the current estimation has already been smaller than a required value. Most greedy algorithms are easy to implement and have low computational complexity for each iteration, thus usually provide high speed reconstruction. The representative algorithms include orthogonal matching pursuit (OMP) [17], regularized OMP [18], stagewise OMP (StOMP) [19], subspace pur-

suit (SP) [20], compressive sampling matching pursuit (CoSaMP) [21] and gradient pursuits [22], etc. Unfortunately, in most situations, the greedy algorithms will not provide a globally optimal solution because the greedy choices are made based on a local criterion.

3. Iterative Thresholding Algorithms

In some articles, the iterative thresholding algorithms are considered as a sub-category of greedy algorithms. Here we list these algorithms separately from above because these iterative thresholding algorithms usually do not need to update the information of selected columns of $\boldsymbol{A}$. Instead, they estimate the signal through a noise corrupted version by soft or hard thresholding functions [23, 24]. In addition, the accelerated iterative thresholding algorithms such as fast iterative shrinkage-thresholding algorithm (FISTA) [25] and Nesterov [26] are eventually fast convex optimization solvers for the LASSO problem [27].

Recently, a new proposed algorithm which is called approximate message passing (AMP) [1, 28] has triggered a lot of attention in the CS field. It has a very similar structure as iterative soft-thresholding (IST) algorithm but with an additional term to the residual part at each iteration:

$$\begin{aligned}
\text{IST}: \boldsymbol{x}^{t+1} &= \eta\left(\boldsymbol{A}^T\boldsymbol{r}^t + \boldsymbol{x}^t\right), \\
\boldsymbol{r}^{t+1} &= \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{t+1}, \\
\text{AMP}: \boldsymbol{x}^{t+1} &= \eta\left(\boldsymbol{A}^T\boldsymbol{r}^t + \boldsymbol{x}^t\right), \\
\boldsymbol{r}^{t+1} &= \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{t+1} + \frac{1}{\delta}\left\langle \eta'\left(\boldsymbol{A}^T\boldsymbol{r}^t + \boldsymbol{x}^t\right)\right\rangle \boldsymbol{r}^t,
\end{aligned}$$

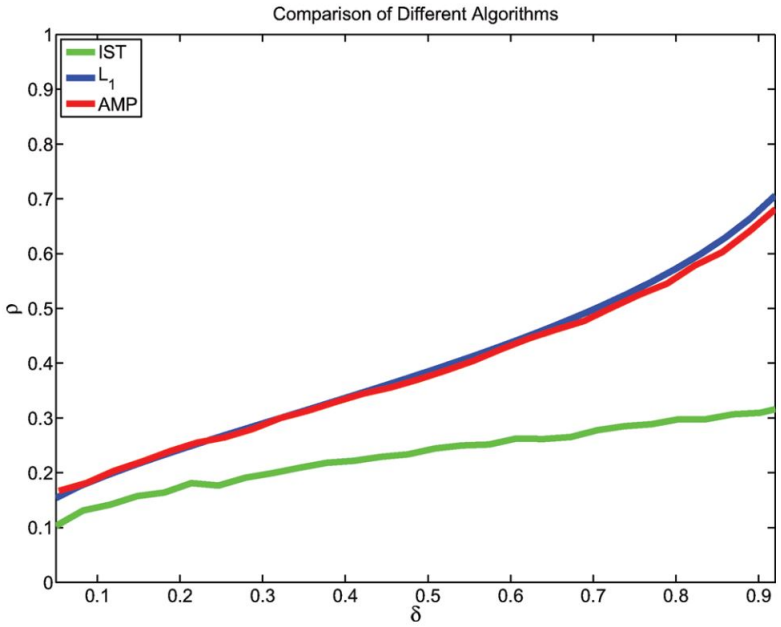


Figure 1.3.1: Phase transition curves of IST (with non-optimally tuned threshold value), ℓ_1 minimization and AMP algorithms [1]

where $\eta(\cdot)$ is an element-wise operator (e.g. a soft-thresholding function), $\delta = \frac{m}{n} > 0$ is the undersampling ratio (m is the number of measurements and n is the dimension of $\boldsymbol{x}^t$), and $\boldsymbol{r}^{t+1}$ represents the residual and the additional term

$$\frac{1}{\delta}\left\langle \eta'\left(\boldsymbol{A}^T\boldsymbol{r}^t + \boldsymbol{x}^t\right)\right\rangle \boldsymbol{r}^t, \tag{1.3.2}$$

is usually referred to as the Onsager term.

Here we give a short description of the effect of the Onsager term. We start with the IST algorithm. The input of the $\eta(\tilde{\boldsymbol{x}}^t)$ function can be decoupled into the superposition of the ground truth signal $\boldsymbol{x}$ and the equivalent noise $\boldsymbol{w}_e^t$ (see Chapter 2 for the derivation):

$$\begin{aligned} \tilde{\boldsymbol{x}}^t &= \boldsymbol{x} + \boldsymbol{w}_e^t, \\ \boldsymbol{w}_e^t &= \left(\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I}\right)\left(\boldsymbol{x} - \boldsymbol{x}^t\right) + \boldsymbol{A}^T\boldsymbol{w}. \end{aligned} \tag{1.3.3}$$

By assuming the terms of $\left(\boldsymbol{A}^T\boldsymbol{A}-\boldsymbol{I}\right)$, $(\boldsymbol{x}-\boldsymbol{x}^t)$ and $\boldsymbol{A}^T\boldsymbol{w}$ are mutually independent and the elements of each of them are i.i.d. (or approximately i.i.d.), the statistics of $\boldsymbol{w}_e^t$ can be easily and well estimated. The problem of IST is that after the first iteration, the terms of $\left(\boldsymbol{A}^T\boldsymbol{A}-\boldsymbol{I}\right)$ and $(\boldsymbol{x}-\boldsymbol{x}^t)$ will be dependent (the estimation of $\boldsymbol{x}^t$ depends on $\boldsymbol{A}$) and this correlation cannot be ignored [29]. Thus, the statistics of $\boldsymbol{w}_e^t$ obtained based on (1.3.3) and the aforementioned independence assumption are no longer precise. In this situation, the threshold value of the soft-thresholding function $\eta(\cdot)$ of IST is not optimally tuned, which affects the performance. On the other hand, the Onsager term of AMP plays the critical role which can asymptotically cancel these correlations, keeping that mutual independence assumption valid across iterations. It is proved in [1, 28] that for i.i.d Gaussian measurement matrices by adding this Onsager term, the performance of AMP dramatically outperformed IST (with non-optimally tuned threshold value) via substantially improving the sparsity-undersampling trade-off (the phase transition curve) see Fig 1.3.1 for illustration. For the optimally tuned IST algorithm, which provably solves the LASSO problem as discussed in [27], should have the same sparsity-undersampling trade-off of AMP but with a significantly larger number of iterations to achieve convergence. (Notice: the sparsity-undersampling trade-off or the phase transition curve of an algorithm describes the relationship between the undersampling ratio δ and the normalized sparsity level $\rho \coloneqq \frac{S}{m}$. It provides the information that for a given sparsity level, the minimum number of measurements required in order to successfully reconstruct the signal with high probability. The theoretical results for AMP are for the linear sparsity, constant undersampling ratio regime (i.e. $S = O(n)$), whereas the original compressed sensing papers assume a sub-linear sparsity regime

(i.e. $S = o(n)$). A detailed explanation of the phase transition curve will be given in Chapter 2.)

The advantage of AMP algorithm is that the sparsity-undersampling trade-off (the phase transition curve) achieved for the i.i.d Gaussian matrices matching the theoretical curve of LP-based reconstruction, which is the best one currently known [1]. At the same time, AMP is an iterative algorithm which has low computational complexity per iteration and fast convergence speed with i.i.d. Gaussian matrices. Moreover, the performance of AMP algorithm can be predicted via a simple scalar iteration which is called state evolution (SE) rather than restricted isometry property (RIP) (listed below). AMP is essentially suitable for large-scale applications which traditional LP-based algorithms may have difficulty to handle.

One drawback of AMP algorithm is that it is designed for a specific class of measurement matrices (i.i.d. Gaussian/sub-Gaussian). The performance of AMP may severely deteriorate if the measurement matrix is significantly different from the standard Gaussian random matrix, whereas convex optimization based algorithms such as (optimally tuned) IST/FISTA work with general matrices.

A bunch of AMP related algorithms have been proposed after the first AMP paper [1] had been published since 2009. Typical ones of the modified AMP algorithms include generalized AMP (GAMP) [30], swept AMP (SwAMP) [31] expectation-maximization Gaussian-mixture AMP (EM-GM-AMP) [32], complex AMP (CAMP) [33] and vector AMP (VAMP) [34], etc. The performance of AMP has been rigorously analysed in [29] in the limit for large dimensions and in [35] for large but finite dimensions. The applications directly related with AMP analysis include sparse superposition codes [36] and spatial coupling

[37, 38].

1.4 Restricted Isometry Property

The reconstruction performances of CS algorithms largely depend on the structure of the measurement matrix $\boldsymbol{A}$. The most popular tool used for judging whether the current matrix $\boldsymbol{A}$ is a good choice for CS problems is called restricted isometry property (RIP), which was first introduced in [3] to analyse the stability and recoverability of CS [39].

Definition 1.4.1 (Restricted Isometry Property [3])**.** The matrix $\boldsymbol{A}$ has the restricted isometry property (RIP) of order S if there exists a constant $0 < \delta_S < 1$ such that

$$(1-\delta_S)\left\|\boldsymbol{x}\right\|_2^2 \leq \left\|\boldsymbol{A}\boldsymbol{x}\right\|_x^2 \leq (1+\delta_S)\left\|\boldsymbol{x}\right\|_2^2,$$
$$\text{or } (1-\delta_S) \leq \frac{\left\|\boldsymbol{A}\boldsymbol{x}\right\|_x^2}{\left\|\boldsymbol{x}\right\|_2^2} \leq (1+\delta_S)$$

for all S-sparse vectors $\boldsymbol{x}$.

In linear algebra, the RIP characterizes matrices which are nearly orthonormal, at least when operating on sparse vectors and a smaller δ_S usually represents a better RIP for the current sparsity value S. The problem is that for a given large matrix $\boldsymbol{A}$, the RIP condition is usually difficult to check because the computation of these constants (δ_S) is strongly NP-hard [40] and is hard to approximate as well [41]. (It has been shown that for many random matrices such as random Gaussian, Bernoulli and partial Fourier matrices, the RIP condition is satisfied with high probability with number of measurements nearly linear in the sparsity level [42].)

1.5 Applications of Compressed Sensing

Due to the small amount of measurements required of CS and sparse representations of signals, it has lots of potential applications in various kinds of fields include undersampling [1, 43], imaging and localization [44, 45, 46], sparse learning [47] and denoising [48], to name a few. Some specific examples are as follows:

1. Medical imaging such as ECG [49], magnetic resonance imaging (MRI) [50, 51] and computerized tomography (CT) [52]. MRI scan requires the patents to lie on a flat bed and keep as still as possible during the scan which usually lasts 15 to 90 minutes [53]. The children and patients suffer from attention deficit hyperactivity disorder (ADHD) may have difficulty to meet the requirement of long time stillness. CT scan contains x-ray radiation, reducing the scan time will reduce the radiation dosage absorbed by the body.

2. Wireless sensor networks (WSNs) [54, 55]. Reducing the number of measurements means reducing the energy consumption of wireless sensors for both data acquisition and transmission. As most of the wireless sensors are battery powered, it will further reduce the labour cost for battery replacement.

3. Compressive radio detecting and ranging (RADAR) [56, 44]. The RADAR systems designed based on CS framework can avoid the using of pulse compression matched filter at the receiver thus simplify the hardware design [44, 5]. Analysis and simulation results in [44, 57] demonstrate that CS techniques can help to improve the resolution of the classical

RADAR systems of which the resolution is restricted by transmitted signal's bandwidth and time-frequency uncertainty principles.

4. Machine learning such as multi-class classification [58] and feature selection [59]. This kind of application requires the appropriate design of matrix $\boldsymbol{A}$ in (1.2.2) based on the given training data set, and utilize the inherent sparse property of the coefficient vector $\boldsymbol{x}$. Taking the classification problem as an example, one can separate the columns of $\boldsymbol{A}$ into different classes by adding labels. When a new data $\boldsymbol{y}$ (e.g. an image) comes in, by applying a proper CS algorithm, an estimate of $\boldsymbol{x}$ will be achieved. By finding the largest magnitude of elements inside the estimated signal, one is able to label the new data.

1.6 Main Contributions

This thesis focuses on AMP based algorithms and the main contributions are as follows:

1. We extend the AMP algorithm to the correlated distributed compressed sensing (C-DCS) cases, i.e. $\boldsymbol{y}_k = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{w}_k \ \forall k \in \{1, 2, ..., K\}$. This model can universally tackle the distributed compressed sensing (DCS), multiple measurement vectors (MMV) and the situations between these two special cases. We consider the measurement matrices $\boldsymbol{A}_k$'s and unknown signals $\boldsymbol{x}_k$'s have the same dimensions and allow correlations exist in both $\boldsymbol{A}_k$'s and $\boldsymbol{x}_k$'s across different measurement instances. By grouping the correlated elements from different measurement instances together to form the block signal and block matrix, we are able to explic-

itly outline the matrices/signals correlation effects and apply SE technique to predict the reconstruction performance. Correctness justifications for the DCS and MMV cases are given. For the general cases, due to the complexity and difficulty of applying the Gaussian conditional lemma, the rigorous analysis is not provided but simulation results show the probability of the correctness. Details are provided in Chapter 3.

2. We consider a practical signal transmission/receiving application with fixed energy budget such as radar/sonar. This kind of system can be modelled by linear equations as (1.2.2) with the assumption that the total energy can be allocated to signals is fixed and thermal noise is the dominant noise source. Under this circumstances, the signal energy per measurement decreases linearly and the noise energy per measurement increases approximately linearly with the increase of the number of measurements. Thus the signal-to-noise ratio (SNR) decreases quadratically with the number of measurements. By applying the state evolution technique for AMP algorithm, we are able to find an optimal number of measurements required to achieve the minimum mean squared error (MMSE) metric. We consider three typical signal models: Gaussian, Bernoulli-Gaussian (BG) and least-favourite (LF) distributions (with a soft-thresholding estimator) in both real and complex domains. Our analysis shows that for these signal models, the optimal under-sampling ratio (measurement number divided by signal dimension) is always upper bounded by 2. Details are provided in Chapter 4.

3. We propose an improved AMP (IAMP) algorithm that can work better for non i.i.d. Gaussian random matrices. The proposed algorithm is

equivalent to AMP for standard Gaussian random matrices but provides better recovery when the correlations between elements of the measurement matrix deviate from the standard Gaussian random matrices. The proposed algorithm is based on a new message passing mechanism with all messages are computed at the variable nodes. Details are provided in Chapter 5.

1.7 Organization of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, we review the background of AMP algorithm and recall the heuristic derivation of AMP from a standard message passing algorithm. In Chapter 3, we discuss the extended AMP algorithm for the C-DCS model. In Chapter 4, we apply SE techniques for AMP algorithm to analyse the optimal number of measurements required for a practical signal transmission/receiving application with fixed energy budget. In Chapter 5, we discuss the IAMP algorithm. In the last chapter, we give conclusions about the thesis and possible future research work.

Chapter 2

Approximate Message Passing

In this chapter, we briefly describe the background of the special algorithm which is called AMP. Compared with other CS algorithms such as OMP [17], SP [20] and CoSaMP [21], etc. whose performance guarantees are often based on RIP condition, the performance of AMP algorithm is based on SE.

In order to make the chapter self-sufficient, we first review the required information.

Consider the linear system

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}, \tag{2.0.1}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ denotes the vector of observations, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is a Gaussian random matrix with elements i.i.d drawn from distribution $\mathcal{N}\left(0, \frac{1}{m}\right)$, $\boldsymbol{x} \in \mathbb{R}^n$ stands for the sparse unknown signal vector and $\boldsymbol{w} \in \mathbb{R}^m$ is the additive white Gaussian noise (AWGN) with distribution $\mathcal{N}(0, \sigma_w^2)$. In CS, $m < n$ is usually assumed such that (2.0.1) is an underdetermined linear system which cannot be directly solved by the normal equation to achieve a unique solution. In

addition, $\boldsymbol{x}$ is assumed to be S-sparse which means that $\boldsymbol{x}$ has at most S non-zero components. The aim of CS is to find the sparest solution that satisfies (2.0.1).

2.1 Overview of AMP

AMP is an iterative algorithm which can be simply described by the following two equations with an initial guess $\boldsymbol{x}^0 = \mathbf{0}$, at each iteration t, we estimate $\boldsymbol{x}^{t+1}$ via

$$\boldsymbol{x}^{t+1} = \eta\left(\boldsymbol{A}^T\boldsymbol{r}^t + \boldsymbol{x}^t\right), \tag{2.1.1}$$

$$\boldsymbol{r}^{t+1} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{t+1} + \frac{1}{\delta}\left\langle \eta'\left(\boldsymbol{A}^T\boldsymbol{r}^t + \boldsymbol{x}^t\right)\right\rangle \boldsymbol{r}^t, \tag{2.1.2}$$

where $\eta(\cdot)$ is the component-wise denoiser, $\eta'(\cdot)$ denotes the derivative of the function $\eta(\cdot)$, $\boldsymbol{A}^T$ is the transpose of matrix $\boldsymbol{A}$, $\delta \coloneqq \frac{m}{n}$ and $\langle \boldsymbol{v}\rangle \coloneqq \frac{1}{n}\sum_{i=1}^{n} v_i$ computes the average of $\boldsymbol{v} \in \mathbb{R}^n$. The last term of (2.1.2) is usually referred to as the Onsager term.

For the signal $\boldsymbol{x}$, the elements are assumed to be i.i.d. drawn from an unknown distribution p_x (only the sparsity $\epsilon = \frac{S}{n}$ is given). Denote the class of these kinds of signals as $\mathcal{F}_\epsilon$, we have $p_x \in \mathcal{F}_\epsilon$. In [1, 60] the soft-thresholding function is selected as the estimator

$$x_i^{t+1} = \eta\left(\tilde{x}_i^t\right) = \begin{cases} \tilde{x}_i^t - \theta^t & \text{if } \tilde{x}_i^t > \theta^t \\ \tilde{x}_i^t + \theta^t & \text{if } \tilde{x}_i^t < -\theta^t \\ 0 & \text{otherwise} \end{cases} \tag{2.1.3}$$

where $\tilde{\boldsymbol{x}}^t \coloneqq \boldsymbol{A}^T\boldsymbol{r}^t + \boldsymbol{x}^t$ and the non-negative value θ^t is the corresponding threshold (will be talked about later). The corresponding derivative is

$$\eta'\left(\tilde{x}_i^t\right) = \begin{cases} 1 & \text{if } \left|\tilde{x}_i^t\right| > \left|\theta^t\right|, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1.4}$$

Worst case analysis considers the following minimax problem

$$\inf_{\hat{x}} \sup_{p_x \in \mathcal{F}_\epsilon} \mathrm{E}\left[\left|x - \hat{x}\right|^2\right],$$

where $\hat{x}$ is the estimate of x. When the estimator (2.1.3) is applied, the following least-favourite (LF) distribution is chosen [60],

$$p_{LF}(x) = \frac{\epsilon}{2}\delta^f_{x=-\infty} + (1-\epsilon)\,\delta^f_{x=0} + \frac{\epsilon}{2}\delta^f_{x=+\infty}, \tag{2.1.5}$$

where δ^f is the Dirac delta function. The superscript ^{f} is used to distinguish the delta function δ^f from the constant δ.

The soft-thresholding function with the LF distribution is a universal denoiser for different kinds of sparse signals, providing sub-optimal solutions for distributions other than LF but with the same sparsity level. For other given distributions, the $\eta(\cdot)$ function can be optimally designed as the MMSE estimator which should outperform the soft-thresholding function.

2.2 Threshold Value: θ^t

In [29, 60], a heuristic argument was presented to derive the specific form of $\tilde{\boldsymbol{x}}^t$ and θ^t. The idea is to decouple the input of the $\eta(\cdot)$ function into the

superposition of the original signal $\boldsymbol{x}$ and white Gaussian noise. Consider three modifications at each iteration t: replace 1) the matrix $\boldsymbol{A}$ with a new independent copy $\boldsymbol{A}(t)$, 2) the observation vector $\boldsymbol{y}$ with $\boldsymbol{y}^t = \boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{w}$ and 3) the Onsager term in (2.1.2) with $\mathbf{0}$. The input of the $\eta(\cdot)$ function can be written as

$$\begin{aligned}
\tilde{\boldsymbol{x}}^t &= \boldsymbol{A}(t)^T \boldsymbol{r}^t + \boldsymbol{x}^t, \\
&= \boldsymbol{A}(t)^T \left(\boldsymbol{y}^t - \boldsymbol{A}(t)\boldsymbol{x}^t\right) + \boldsymbol{x}^t, \\
&= \boldsymbol{A}(t)^T \left(\boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{w} - \boldsymbol{A}(t)\boldsymbol{x}^t\right) + \boldsymbol{x}^t + \boldsymbol{x} - \boldsymbol{x}, \\
&= \boldsymbol{x} + \boldsymbol{w}_e^t
\end{aligned} \tag{2.2.1}$$

which is the ground truth signal $\boldsymbol{x}$ plus an equivalent noise

$$\boldsymbol{w}_e^t \coloneqq \left(\boldsymbol{A}(t)^T \boldsymbol{A}(t) - \boldsymbol{I}\right)\left(\boldsymbol{x} - \boldsymbol{x}^t\right) + \boldsymbol{A}(t)^T \boldsymbol{w}. \tag{2.2.2}$$

By the central limit theorem and the assumption that $\boldsymbol{A}(t)$'s are independent across t, $\left(\boldsymbol{A}(t)^T \boldsymbol{A}(t) - \boldsymbol{I}\right)$ and $(\boldsymbol{x} - \boldsymbol{x}^t)$ are always independent, thus the equivalent noise $\boldsymbol{w}_e^t$ is always approximately Gaussian and contains i.i.d. components. Then the statistics (variance) of the equivalent noise $\boldsymbol{w}_e^t$ can be computed based on (2.2.2) which gives

$$\left(\sigma_e^t\right)^2 = \frac{1}{\delta}\mathrm{Err}_t + \sigma_w^2, \tag{2.2.3}$$

where

$$\mathrm{Err}_t = \lim_{n\to\infty} \frac{1}{n} \left\|\boldsymbol{x} - \boldsymbol{x}^t\right\|_2^2 \tag{2.2.4}$$

represents the MSE of the previous estimation ($\boldsymbol{x}^t$ is achieved at the $(t-1)$-th iteration).

When the soft-thresholding function (2.1.3) is chosen as the estimator with the corresponding LF distribution (2.1.5), Err_t is computed via

$$\mathrm{Err}_t = M\left(\epsilon, \alpha^\dagger\right)\left(\sigma_e^{t-1}\right)^2, \tag{2.2.5}$$

where $\left(\sigma_e^{t-1}\right)^2$ is the variance of the equivalent noise at the previous iteration (with initial value $\left(\sigma_e^0\right)^2 = \frac{\|y\|_2^2}{m}$) and

$$\alpha^\dagger = \arg\min_{\alpha\in\mathbb{R}_+} M(\epsilon, \alpha), \tag{2.2.6}$$

$$M(\epsilon, \alpha) \coloneqq \epsilon\left(1+\alpha^2\right) + (1-\epsilon)\left[2\left(1+\alpha^2\right)\Phi(-\alpha) - 2\alpha\phi(\alpha)\right], \tag{2.2.7}$$

where $\phi(x)$ is the standard Gaussian density and $\Phi(x) = \int_{-\infty}^{x}\phi(t)\,dt$ is the corresponding cumulative distribution function. The optimal threshold for current iteration t is achieved by

$$\theta^t = \alpha^\dagger\sigma_e^t. \tag{2.2.8}$$

AMP algorithm does not have independent copy of $\boldsymbol{A}(t)$ and new observation $\boldsymbol{y}^t$ at each iteration, but with the efforts of the Onsager term in (2.1.2), the Gaussianity of $\boldsymbol{w}_e^t$ still holds. The Onsager term asymptotically cancels the correlation between iterations.

2.3 State Evolution and Phase Transition

SE describes the asymptotic performance of the AMP algorithm in the asymptotic region, in which m, $n \to \infty$ and keep $\delta \to \frac{m}{n}$ as a constant. The performance of the system can be described by a sequence $\{\tau_t^2\}_{t\geq 0}$ with initial condition $\tau_0^2 = \sigma_w^2 + \mathrm{E}\left[X^2\right]/\delta$ (X with a density function p_x), for all $t > 0$, we have

$$\tau_{t+1}^2 = F\left(\tau_t^2, \theta^t\right), \tag{2.3.1}$$

$$F\left(\tau^2, \theta\right) \coloneqq \frac{1}{\delta}\mathrm{E}\left[\left(\eta\left(X + \tau Z; \theta\right) - X\right)^2\right] + \sigma_w^2, \tag{2.3.2}$$

where $Z \sim \mathcal{N}(0,1)$ is independent of X. More details about SE can be found in [60] and the correctness of SE has been rigorously proved in [29].

The sequence $\{\tau_t^2\}_{t\geq 0}$ can be calculated using (2.2.3) and (2.2.5). As long as the sparsity level ϵ is given, the value of $\alpha^\dagger$ can be easily calculated via (2.2.6) and $M\left(\epsilon, \alpha^\dagger\right)$ becomes a constant. Assume AMP algorithm converges, when $t \to \infty$, the value of σ_e^t (or τ_t) coverages to a steady state point and the final MSE of the algorithm can be calculated. The SE technique is often used to predict the asymptotic performance of the system even without applying the AMP algorithm.

Let $\rho = \frac{S}{m}$, then $\epsilon = \rho\delta$. AMP algorithm has a phase transition curve shown as in Fig 2.3.1 which is achieved under the asymptotic assumption. The curve represents the trade-off between sparsity level and under-sampling ratio, separating the figure into two regions [1], in the noise free case:

- Region 1 (below the curve): in this region, $F\left(\tau^2, \theta\right) < \tau^2$ for all $\tau^2 \in (0, \mathrm{E}\left(X^2\right)]$. When $t \to \infty$, $\tau_t^2 \to 0$: the SE converges to zero.

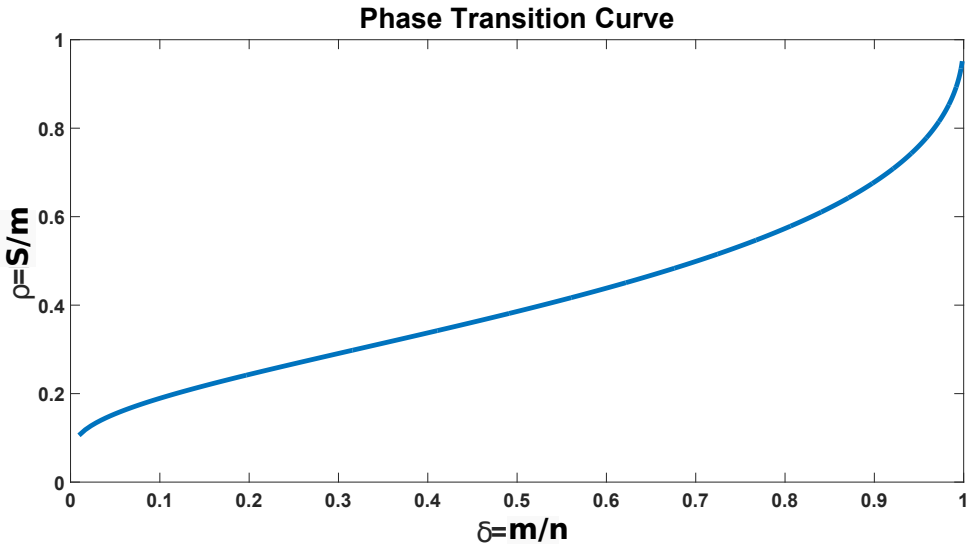


Figure 2.3.1: Phase transition curve of AMP. Below the curve is region 1 where we can successfully reconstruct the signal with high probability, above the curve is region 2 where we cannot reconstruct the signal with high probability.

- Region 2 (above the curve): in this region, the SE will not converge to zero.

In the noisy case (with noise variance σ_w^2), the same curve exists but τ_t^2 will converge to a non-zero value $M_c\sigma_w^2/M\left(\epsilon,\alpha^\dagger\right)$ and the estimation error will converge to $M_c\sigma_w^2$ where

$$M_c = \begin{cases} \frac{M\left(\epsilon,\alpha^\dagger\right)}{1-M\left(\epsilon,\alpha^\dagger\right)/\delta}, & \text{in Region 1} \\ \infty, & \text{in Region 2} \end{cases} \tag{2.3.3}$$

which is called the noise sensitivity. Interested readers can refer to [61] for more details.

2.4 A Heuristic Derivation from Message Passing

The authors of [29, 60] provide a heuristic derivation of AMP from the traditional message passing (MP) algorithm which contains the following two iterative equations,

$$r^t_{a\to i} = y_a - \sum_{j\in[n]\setminus i} A_{aj} x^t_{j\to a}, \tag{2.4.1}$$

$$x^{t+1}_{i\to a} = \eta\left(\sum_{b\in[m]\setminus a} A_{bi} r^t_{b\to i}\right), \tag{2.4.2}$$

where subscript $a \to i$ in (2.4.1) represents the message from the factor node (contains information of observations) $a \in [m]$ to the variable node (contains information of signals) $i \in [n]$, and $i \to a$ in (2.4.2) represents the message from the variable node i to the factor node a, $[n] \coloneqq \{1, 2, ..., n\}$ and $[n]\setminus i$ denotes the set of $[n]$ but without element i. A direct calculation of MP based on (2.4.1) and (2.4.2) will not be practical especially for large dimension systems as it requires to update mn messages per iteration. The computational complexity of MP is extremely high and approximation must be applied.

The right-hand side of (2.4.1) contains a summation over $\Theta(n)$ messages. For any fixed a, $r^t_{a\to i}$ only depends on i as it excluded from the summation $\sum_{j\in[n]\setminus i} A_{aj} x^t_{j\to a}$. Similarly, for any given i, $x^{t+1}_{i\to a}$ only depends on a. Thus we can set

$$r^t_{a\to i} = r^t_a + \Delta r^t_{a\to i},\ x^t_{i\to a} = x^t_i + \Delta x^t_{i\to a}, \tag{2.4.3}$$

where $\Delta r^t_{a\to i}$ and $\Delta x^t_{i\to a}$ denote some small values $O\left(m^{-1/2}\right)$. Substituting

(2.4.3) into (2.4.1) and (2.4.2) provides

$$r_a^t + \Delta r_{a\to i}^t = y_a - \sum_{j\in[n]} A_{aj}\left(x_j^t + \Delta x_{j\to a}^t\right) + A_{ai}\left(x_i^t + \Delta x_{i\to a}^t\right),$$
$$x_i^{t+1} + \Delta x_{i\to a}^{t+1} = \eta\left(\sum_{b\in[m]} A_{bi}\left(r_b^t + \Delta r_{b\to i}^t\right) - A_{ai}\left(r_a^t + \Delta r_{a\to i}^t\right)\right).$$

Recall that $A_{ai} = O\left(m^{-1/2}\right)$ by definition, it will be safe to drop the terms $A_{ai}\Delta x_{i\to a}^t$ and $A_{ai}\Delta r_{a\to i}^t$ (Note: the terms inside $\sum$ cannot be ignored). We have

$$r_a^t + \Delta r_{a\to i}^t = y_a - \sum_{j\in[n]} A_{aj}\left(x_j^t + \Delta x_{j\to a}^t\right) + A_{ai}x_i^t, \tag{2.4.4}$$
$$x_i^{t+1} + \Delta x_{i\to a}^{t+1} = \eta\left(\sum_{b\in[m]} A_{bi}\left(r_b^t + \Delta r_{b\to i}^t\right) - A_{ai}r_a^t\right).$$

By applying first order Taylor's expansion to the second equation listed above, we achieve the following approximation

$$x_i^{t+1} + \Delta x_{i\to a}^{t+1} \approx \eta\left(\sum_{b\in[m]} A_{bi}\left(r_b^t + \Delta r_{b\to i}^t\right)\right) - \eta'\left(\sum_{b\in[m]} A_{bi}\left(r_b^t + \Delta r_{b\to i}^t\right)\right) A_{ai}r_a^t, \tag{2.4.5}$$

where $\eta(\cdot)$ only needs to be almost-everywhere differentiable according to [29, 60], thus this approximation is suitable for the soft-thresholding function which is only non-differentiable at $\eta(\pm\theta)$.

Now compare (2.4.4) and (2.4.5) with (2.4.3), a reasonable decomposition will be

$$r_a^t = y_a - \sum_{j\in[n]} A_{aj}\left(x_j^t + \Delta x_{j\to a}^t\right), \tag{2.4.6}$$

$$\Delta r^t_{a\to i} = A_{aj} x^t_i, \tag{2.4.7}$$

$$x^{t+1}_i = \eta\left(\sum_{b\in[m]} A_{bi}\left(r^t_b + \Delta r^t_{b\to i}\right)\right), \tag{2.4.8}$$

$$\Delta x^{t+1}_{i\to a} = -\eta'\left(\sum_{b\in[m]} A_{bi}\left(r^t_b + \Delta r^t_{b\to i}\right)\right) A_{ai} r^t_a. \tag{2.4.9}$$

Based on the definition $\sum_{a\in[m]} A^2_{ai} = 1$, the AMP algorithm can be achieved by substituting (2.4.9) into (2.4.6) and substituting (2.4.7) into (2.4.8).

Chapter 3

Correlated-Distributed Compressed Sensing

In this chapter, we study the correlated distributed compressed sensing (C-DCS) scenarios where the measurement matrices and the signals at different sensors can be correlated. It is assumed that the measurement matrices are Gaussian random matrices and the elements across signals at the the same positions share a same distribution. Our model is a generalization of the commonly used DCS model where the measurement matrices are independent and the standard MMV model where the measurement matrices are identical. Based on the famous AMP framework, an algorithm is developed to address correlated matrices and correlated signals. Correctness justification of the two special cases has been given. For the more general cases between DCS and MMV, we outline the complexity and difficulty to justify the correctness but simulation results validate the accuracy of SE .

3.1 Introduction

In many applications such as parallel magnetic resonance imaging [51], direction of arrival estimation [62], distributed sensor networks [63] and medical imaging [64], the system can be modelled as compressed sensing with multiple measurement instances (MMI), each involving a standard compressed sensing procedure. It is typical that the unknown signals or/and the measurement matrices from different measurement instances are not independent [65]. It has been widely accepted that the performance gain is possible by joint processing that explores the structure of multiple instances [65].

In this chapter, we model the C-DCS systems via $\boldsymbol{y}_k = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{w}_k$, $k \in [K]$ where K is the total number of measurement instances and $[K] \coloneqq \{1, 2, ..., K\}$. Let $x_{k,i}$ be the i-th element from the k-th signal $\boldsymbol{x}_k$. For the involved signals $\boldsymbol{x}_k$'s, we consider the elements in the same $\boldsymbol{x}_k$ (e.g. $x_{k,i}$ and $x_{k,j}$ for $i \neq j$, $i, j \in [n]$, $k \in [K]$) are independent from each other, whereas for elements across different $\boldsymbol{x}_k$'s at the same location (e.g. $x_{a,i}$ and $x_{b,i}$ for $a \neq b$, $a, b \in [k]$, $i \in [n]$) are dependent. There are different ways to model the measurement matrices $\boldsymbol{A}_k$'s. In decades long MMV model, it is typically assumed that $\boldsymbol{A}_k$'s are identical. In DCS setting [65, 43], it is often assumed that $\boldsymbol{A}_k$'s are independent. In a more recent work [66], a matrix innovation model is used where $\boldsymbol{A}_k$'s are modelled as a times series and the matrices at adjacent time instances, say $\boldsymbol{A}_{k-1}$ and $\boldsymbol{A}_k$, are correlated with a constant correlation factor. In this chapter, we consider the general model (C-DCS) that accommodates all the three cases above. In particular, we model each measurement matrix $\boldsymbol{A}_k$ as a Gaussian random matrix and correlations across measurement matrices by a multivariate Gaussian distribution.

Special cases of the C-DCS model has been studied in the literature. Most algorithmic solutions focus on exploring the common sparse support structure of the signals, resulting in distributed subspace pursuit (DiSP) [67], distributed and collaborative orthogonal matching pursuit (DC-OMP) [68, 69], subspace augmented multiple signal classifier (SA-MUSIC) [70], MMV focal underdetermined system solution (M-FOCUSS) [71], approximated message passing MMV (AMP-MMV) [66] and joint approximated message passing [43, 72], to name a few. Performance analysis mainly focuses on the DCS scenarios where the measurement matrices are independent. RIP based analysis has been used to analyse greedy algorithms [73, 67, 68, 69]. A more recent tool, SE for AMP, has been used to exactly quantify the performance in an asymptotic regime [66, 43].

The contributions of this chapter are

1. We consider both the measurement matrices and unknown signals in the C-DCS model are correlated. By grouping the correlated coefficients from different measurement instances together to form the block signal and block matrix (see Definition 3.2.1 and Definition 3.2.2), we are able to derive our algorithm which is called AMP-C-DCS based on the standard message passing algorithm and explicitly outline the matrices/signals correlation effects.

2. Although the derivation of our algorithm follows the same steps of the original AMP algorithm which is clearly understandable, the rigorous proof of the correctness is another story. Due to the correlation between measurement matrices $\boldsymbol{A}_k$'s, the extension of Gaussian conditional lemma, which is a key technique in the AMP proof provided in [29], is

not straightforward. We provide a detailed discussion about this problem and show the complexity and difficulty to justify the correctness in the general form of our algorithm. Simulation results validate the accuracy of SE.

3. We pay particular attention to the special cases: DCS and MMV. By certain rearrangement of the structure of the super components, which are treated as diagonal matrices in the DCS model and as row vectors in the MMV model, the general form of our algorithm degenerates to the two simplified models, such that the analysis in [29, 37] can be applied to prove the correctness of our algorithm for the two special cases. Our proposed algorithm can be considered as a generalized version of the algorithm proposed in [74, 37] which is only designed for the MMV problem. Although the correlated measurement matrices condition has been considered in [66] and performance improvements have been observed by reducing the matrix correlation level, the matrix correlation effect has not been inherently analysed in their algorithm.

The structure of this chapter is as follows. In section 2, we provides the detailed description of our proposed system model and recall the background of AMP algorithm. In section 3, we discuss the AMP-C-DCS algorithm. In section 4, we justify the correctness of our proposed algorithm. The case study of the Bernoulli-Gaussian (BG) and Gaussian signals and simulation results are listed in section 5.

3.2 Preliminaries

3.2.1 System Model for C-DCS

Consider an application system contains K sensors (measurement instances henceforth), each of which is represented by a linear system:

$$\boldsymbol{y}_k = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{w}_k, \forall k \in [K], \tag{3.2.1}$$

where $\boldsymbol{y}_k \in \mathbb{R}^m$ is the observation vector, $\boldsymbol{A}_k \in \mathbb{R}^{m \times n}$ denotes the measurement matrix, $\boldsymbol{x}_k \in \mathbb{R}^n$ stands for the unknown signal vector, $\boldsymbol{w}_k \in \mathbb{R}^m$ represents the additive measurement noise, and the set $[K] \coloneqq \{1, 2, ..., K\}$. The overall system can be written in a compact form with a diagonal structure as

$$\mathrm{D-model}: \ \boldsymbol{y}_D = \boldsymbol{A}_D \boldsymbol{x}_D + \boldsymbol{w}_D, \tag{3.2.2}$$

where $\boldsymbol{y}_D \coloneqq \mathrm{diag}\left([\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_K]\right) \in \mathbb{R}^{Km \times K}$ is a block diagonal matrix with diagonal elements (vectors or matrices) $\boldsymbol{y}_k$'s and same definition applies for $\boldsymbol{A}_D \in \mathbb{R}^{Km \times Kn}$, $\boldsymbol{x}_D \in \mathbb{R}^{Kn \times K}$ and $\boldsymbol{w}_D \in \mathbb{R}^{Km \times K}$. (Notice: in some cases such as Gaussian conditional lemma which we will talk about later, we treat observation $\boldsymbol{y}_D = \left[\boldsymbol{y}_1^T, \boldsymbol{y}_2^T, ..., \boldsymbol{y}_K^T\right]^T \in \mathbb{R}^{Km}$ as vector just for mathematical calculation, the same applies for unknown signals and noise. As long as the measurement matrix keeps the block diagonal structure, these models are mathematically equivalent)

For the signal, we allow correlation between the elements across different measurement instances. A statistical model is defined to allow more detailed description of signal components as follows.

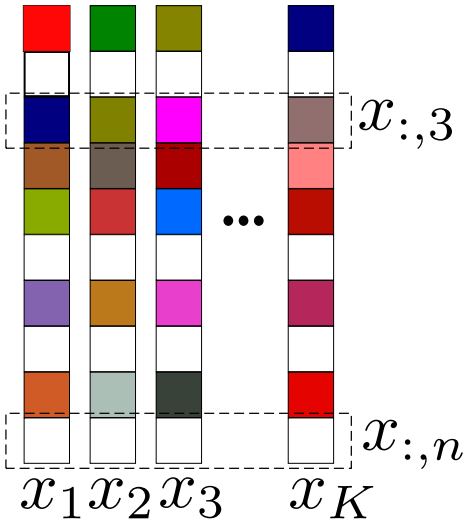


Figure 3.2.1: Group signals. $\boldsymbol{x}_k$'s, $\forall k \in [K]$ have the same signal dimension. $\boldsymbol{x}_{:,i}$'s, $\forall i \in [n]$ are the group signal elements, white colours represent zero elements. The elements in $\boldsymbol{x}_k$, $k \in [K]$ are independent from each other and $\boldsymbol{x}_{:,i}$'s, $i \in [n]$ are i.i.d drawn from a multivariate distribution (e.g. multivariate BG distribution and multivariate Gaussian distribution).

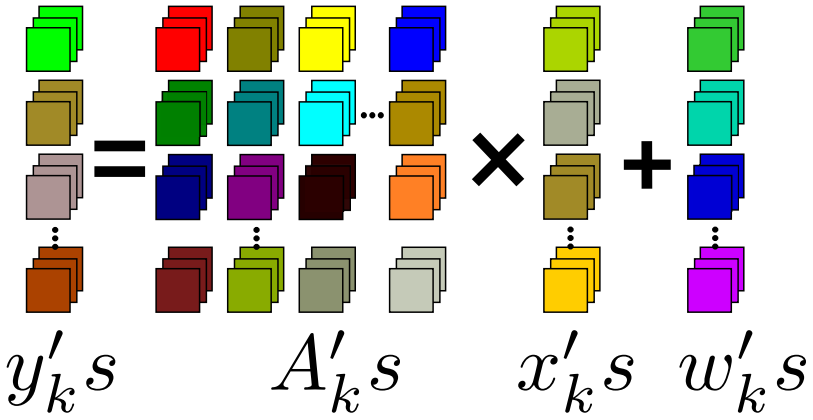
Definition 3.2.1 (Block Signal Model)**.** Consider K signals of the same dimension $\boldsymbol{x}_k \in \mathbb{R}^n$, $k \in [K]$. Group the i-th components of $\boldsymbol{x}_k$'s together to form the block component (super component) $\boldsymbol{x}_{:,i} = [x_{1,i}, ..., x_{K,i}]^T \in \mathbb{R}^K$, $i \in [n]$ (see Fig 3.2.1). Define the block signal as $\boldsymbol{x}_B \coloneqq \left[\boldsymbol{x}_{:,1}^T, ..., \boldsymbol{x}_{:,n}^T\right]^T \in \mathbb{R}^{Kn}$. The block components (super components) $\boldsymbol{x}_{:,i}$'s are i.i.d drawn from a multivariate distribution $p_{\boldsymbol{X}_{:,i}}(\boldsymbol{x})$ (e.g. multivariate BG distribution and multivariate Gaussian distribution).

Following the convention in compressed sensing, the measurement matrices $\boldsymbol{A}_k$'s are assumed to be Gaussian random matrices. Different from standard models in the literature, we allow the components of the measurement matrices to be correlated. This is motivated by the fact that in most compressed sensing applications, physical or design constraints do not allow independent measurement matrices at sensors. Specifically, we have

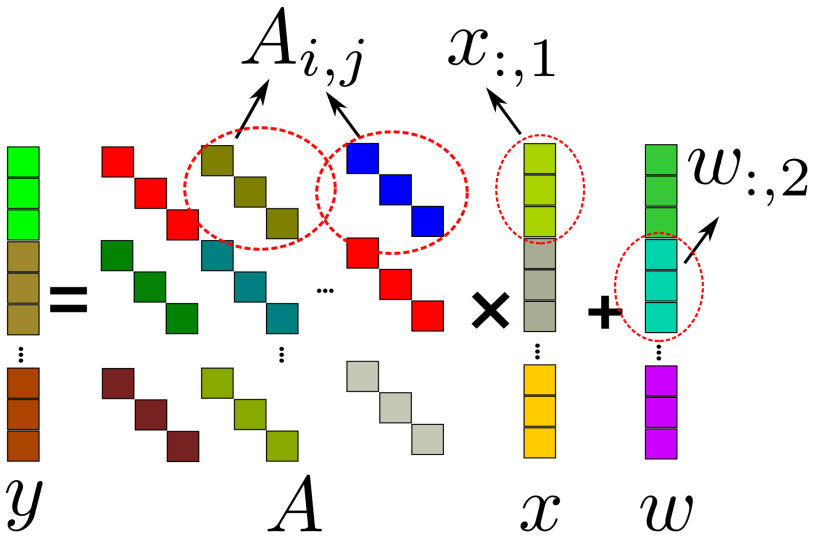
Definition 3.2.2 (Block Matrix Model)**.** Consider K matrices of the same dimension $\boldsymbol{A}_k \in \mathbb{R}^{m\times n}$, $k \in [K]$. Group the (i,j)-th components of $\boldsymbol{A}_k$'s together to form the block component (super component) $\boldsymbol{a}_{:,i,j} = [a_{1,i,j}, ..., a_{K,i,j}]^T \in \mathbb{R}^K$, $i \in [m]$, $j \in [n]$. Assume that the block components (super components) $\boldsymbol{a}_{:,i,j}$'s are i.i.d. multivariate Gaussian random variables based on the multivariate Gaussian distribution $\mathcal{N}\left(\boldsymbol{0}, \frac{1}{m}\boldsymbol{\Sigma}_A\right)$[1], where, for the normalization purpose, the diagonal elements of the covariance matrix $\boldsymbol{\Sigma}_A$ are one. Define the block matrix as $\boldsymbol{A}_B \in \mathbb{R}^{Km\times Kn}$ where the (i,j)-th block $\boldsymbol{A}_{B,i,j} \coloneqq \operatorname{diag}(\boldsymbol{a}_{:,i,j}) \in \mathbb{R}^{K\times K}$.

This proposed model generalizes several popular models in signal processing and compressed sensing. The traditional subspace-based methods, for example Multiple Signal Classification (MUSIC) and Estimation of Signal Parameters by Rotational Invariance Techniques (ESPRIT), typically assume the MMV model where the measurement matrices $\boldsymbol{A}_k$'s are identical [75]. This model can be interpreted as a special case of Definition 3.2.2 where all the entries in the covariance matrix $\boldsymbol{\Sigma}_A$ are one. A mathematical explanation is based on (3.2.1), which can be used to represent a source localization problem with multiple time samples. Assume that a same sensor takes multiple snapshots from the source during a short period, we will have repeated measurement matrices (i.e. $\boldsymbol{A}_k = \boldsymbol{A}_1$ for $k \in [K]$), and $\boldsymbol{x}_k$'s may be highly correlated. At the opposite extreme, the most common assumption of DCS is that the measurement matrices $\boldsymbol{A}_k$'s are independent. This corresponds to the case that the covariance matrix $\boldsymbol{\Sigma}_A$ in Definition 3.2.2 is the identity matrix. We can still use (3.2.1) and the source localization problem to explain this situation,

[1]The constant $\frac{1}{m}$ is introduced for normalization and convenience. The same normalization appeared in the literature [29].



(a) Multiple measurement instances



(b) B-model and super components

Figure 3.2.2: Equivalent model ($K = 3$). The elements at the same position from different measurement instances are represented with the same colour while the values of them are not necessary to be the same. $\boldsymbol{A}_{i,j}$, $\boldsymbol{x}_{:,i}$ and $\boldsymbol{w}_{:,i}$ are treated as super components.

but this time, completely different sensors are applied.

A more general model, referred to as matrix innovation model, proposed in [66] which treats the measurement matrices $\boldsymbol{A}_k$'s as a time series given by

$$\boldsymbol{A}_k = (1-\beta)\,\boldsymbol{A}_{k-1} + \beta \boldsymbol{U}_k,\ k \in \{2,3,...,K\}\,, \tag{3.2.3}$$

where $\beta \in [0,1]$ controls the correlation between the matrices at adjacent time instances. They start with a Gaussian random matrix $\boldsymbol{A}_1$ with elements i.i.d. drawn form $\mathcal{N}\left(0, \frac{1}{m}\right)$. Let $\boldsymbol{U}_k$ be a Gaussian random matrix with elements i.i.d. drawn from $\mathcal{N}\left(0, \left(\frac{2}{\beta}-1\right)\frac{1}{m}\right)$ and independent of $\boldsymbol{A}_{k-1}$ ($\boldsymbol{U}_k$ is not required when $\beta = 0$). This model is also a special case of Definition 3.2.2 : for a given finite K, the elements in a row of the $\boldsymbol{\Sigma}_A$ form a geometric series of $1-\beta$. For example, when $K = 3$, we have

$$\boldsymbol{\Sigma}_A = \begin{bmatrix} 1 & 1-\beta & (1-\beta)^2 \\ 1-\beta & 1 & 1-\beta \\ (1-\beta)^2 & 1-\beta & 1 \end{bmatrix}.$$

(Notice: as in Definition 3.2.2, we take the parameter $\frac{1}{m}$ out from $\boldsymbol{\Sigma}_A$.)

Based on Definition 3.2.1 and Definition 3.2.2, the overall system can also be written as the following compact form

$$\mathrm{B}-\mathrm{model}:\ \boldsymbol{y}_B = \boldsymbol{A}_B \boldsymbol{x}_B + \boldsymbol{w}_B, \tag{3.2.4}$$

where $\boldsymbol{y}_B$, $\boldsymbol{x}_B$, $\boldsymbol{w}_B$ have the structure as in Definition 3.2.1 and $\boldsymbol{A}_B$ has the structure as in Definition 3.2.2. (see Fig 3.2.2(b))

Remark 3.2.3. We consider $\mathrm{B}-$model and $\mathrm{D}-$model are mathematically

equivalent. We will most often omit the subscript B and D for simplified notation as long as the model is clear according to the context. With slight abuse of notation, the mean vector and the covariance matrix of $\boldsymbol{A}_{B,i,j}$ are referred to as those of its diagonal vector $\boldsymbol{a}_{:,i,j}$. In addition, the block signal element $\boldsymbol{x}_{:,i}$ and block matrix element $\boldsymbol{A}_{B,i,j}$ both can be referred to as super components.

3.2.2 AMP Algorithm with an MMSE Estimator

Recall that AMP algorithm is proposed in [1] for solving the linear equations (3.2.2) when $K = 1$ (ignore the subscript D) by assuming $\boldsymbol{A}$ is a standard Gaussian random matrix, $\boldsymbol{x}$ is a random vector and $\boldsymbol{w}$ is the additive white Gaussian noise, all containing i.i.d. components. AMP algorithm updates the estimated signal $\boldsymbol{x}^{t+1}$ at the t-th iteration via (2.1.1) and (2.1.2) and the input of $\eta(\cdot)$ can be written as (2.2.1) which is the ground truth signal $\boldsymbol{x}$ plus the equivalent noise $\boldsymbol{w}_e^t$ (more detailed information can be found in Chapter 2). If only the sparsity level of $\boldsymbol{x}$ is given without knowing its actual distribution, a soft-thresholding function (2.1.3) can be chosen for denoising. If the distribution of the signal $\boldsymbol{x}$ (or more precisely, of the signal elements x_i's) is given (e.g. BG distribution), the soft-thresholding function can be replaced by an optimally designed MMSE estimator which usually provides a better performance. Given the statistics of the signal and the equivalent noise, based on (2.2.1), the specific form of the MMSE estimator can be obtained via the following equation, for each element of $\boldsymbol{x}^{t+1}$,

$$x_i^{t+1} = \eta\left(\tilde{x}_i^t\right) \coloneqq \mathrm{E}_{x_i|\tilde{x}_i^t}\left[x_i|\tilde{x}_i^t\right], \forall i \in [n], \tag{3.2.5}$$

then the statistics of the estimation error $\boldsymbol{x}-\boldsymbol{x}^{t+1}$ can be calculated which will be used to update the equivalent noise $\boldsymbol{w}_e^{t+1}$ in the next iteration. (The special situation, in which the sparsity structure of the signal is given, i.e. group sparse signals, has been recently discussed in [76, 77, 78]. Unfortunately, the performances of these AMP based algorithms were not theoretically analysed, i.e. SE technique cannot be used to predict their performances.)

3.2.3 State Evolution of AMP

The SE technique describes the asymptotic performance of the AMP algorithm in the asymptotic region, in which $m, n \to \infty$ and keep $\frac{m}{n} \to \delta$ fixed. The performance of the system can be described by a sequence of $\left\{\left(\sigma_e^t\right)^2\right\}_{t\geq 0}$ with initial condition $\left(\sigma_e^0\right)^2 = \frac{1}{\delta}E\left[X^2\right] + \sigma_w^2$ (X with a density function p_X), for $t > 0$, calculate

$$\left(\sigma_e^t\right)^2 = \frac{1}{\delta}E\left[\left(\eta_t\left(X + \sigma_e^{t-1}Z\right) - X\right)^2\right] + \sigma_w^2, \tag{3.2.6}$$

where $Z \sim \mathcal{N}(0,1)$ is independent of X. The calculation of (3.2.6) is directly based on the (2.2.2) and the Gaussianity of $\boldsymbol{w}_e^t$ plays the key role during the calculation. If AMP algorithm converges, when $t \to \infty$, $\left(\sigma_e^t\right)^2$ converges to a fixed point and the performance of the system can be predicted without applying the AMP algorithm.

The rigorous proof of the correctness of the SE is given in [29] with a more general form listed below, with initial condition $\boldsymbol{q}^0$ and define $\boldsymbol{m}^{-1} = \boldsymbol{0}$,

$$\begin{aligned} \boldsymbol{h}^{t+1} &= \boldsymbol{A}^T\boldsymbol{m}^t - \xi_t\boldsymbol{q}^t,\ \boldsymbol{m}^t = g_t\left(\boldsymbol{b}^t, \boldsymbol{w}\right), \\ \boldsymbol{b}^t &= \boldsymbol{A}\boldsymbol{q}^t - \lambda_t\boldsymbol{m}^{t-1},\ \boldsymbol{q}^t = f_t\left(\boldsymbol{h}^t, \boldsymbol{x}\right), \end{aligned} \tag{3.2.7}$$

where $g_t(\cdot)$, $f_t(\cdot)$ are assumed to be Lipschitz continuous (almost everywhere continuously differentiable, see [29] for the rigorous definition) and applied component-wise, $\xi_t \coloneqq \langle g_t'(\boldsymbol{b}^t, \boldsymbol{w})\rangle$, and $\lambda_t \coloneqq \frac{1}{\delta}\langle f_t'(\boldsymbol{h}^t, \boldsymbol{x})\rangle$. As mentioned in [29] , the AMP algorithm is a special case of (3.2.7) by defining

$$\begin{aligned} \boldsymbol{h}^{t+1} &= \boldsymbol{x} - \left(\boldsymbol{A}^T\boldsymbol{r}^t + \boldsymbol{x}^t\right),\ \boldsymbol{q}^t = \boldsymbol{x}^t - \boldsymbol{x}, \\ \boldsymbol{b}^t &= \boldsymbol{w} - \boldsymbol{r}^t,\ \boldsymbol{m}^t = -\boldsymbol{r}^t. \end{aligned} \tag{3.2.8}$$

with $f_t(\boldsymbol{h}^t, \boldsymbol{x}) = \eta_{t-1}(\boldsymbol{x} - \boldsymbol{h}^t) - \boldsymbol{x}$, $g_t(\boldsymbol{b}^t, \boldsymbol{w}) = \boldsymbol{b}^t - \boldsymbol{w}$ and initial condition $\boldsymbol{q}^0 = -\boldsymbol{x}$. (3.2.8) only can be applied for mathematical analysis as it requires $\boldsymbol{x}$ to initialize $\boldsymbol{q}^0$. The terms $\xi_t\boldsymbol{q}^t$ and $\lambda_t\boldsymbol{m}^{t-1}$ in (3.2.7) play the same role as the Onsager term in (2.1.2) which ensure the Gaussianity of $\boldsymbol{h}^{t+1}$ and $\boldsymbol{b}^t$ at each iteration. For the AMP case, $\boldsymbol{h}^{t+1}$ eventually represents the equivalent noise $\boldsymbol{w}_e^t$.

The key ideal of the proof in [29] is to avoid directly tracking the statistics (conditional distributions) of $\boldsymbol{m}^t$, $\boldsymbol{q}^t$ given $\boldsymbol{A}$ at each iteration. Instead, they calculate the conditional distribution of $\boldsymbol{A}$ given the σ-algebra $\mathfrak{G}_{t1,t2}$ ($(t1,t2) = (t,t)$ or $(t1,t2) = (t+1,t)$) generated by $\{\boldsymbol{b}^t\}_{t\geq 0}$, $\{\boldsymbol{m}^t\}_{t\geq 0}$, $\{\boldsymbol{h}^t\}_{\geq 1}$ and $\{\boldsymbol{q}^t\}_{\geq 1}$. To be more explicit, define

$$\begin{aligned} \underbrace{\left[\boldsymbol{h}^1 + \xi_0\boldsymbol{q}^0 | \ldots | \boldsymbol{h}^t + \xi_{t-1}\boldsymbol{q}^{t-1}\right]}_{\boldsymbol{X}_t} &= \boldsymbol{A}^T \underbrace{\left[\boldsymbol{m}^0 | \ldots | \boldsymbol{m}^{t-1}\right]}_{\boldsymbol{M}_t}, \\ \underbrace{\left[\boldsymbol{b}^0 | \ldots | \boldsymbol{b}^{t-1} + \lambda_{t-1}\boldsymbol{m}^{t-2}\right]}_{\boldsymbol{Y}_t} &= \boldsymbol{A} \underbrace{\left[\boldsymbol{q}^0 | \ldots | \boldsymbol{q}^{t-1}\right]}_{\boldsymbol{Q}_t}, \end{aligned} \tag{3.2.9}$$

the proof in [29] tracks the conditional distribution of $\boldsymbol{A}$ given linear equations

(3.2.9), i.e.

$$\boldsymbol{A}|_{\mathfrak{G}_{t1,t2}} \overset{d}{=} \boldsymbol{A}|_{\boldsymbol{X}_{t2}=\boldsymbol{A}^T\boldsymbol{M}_{t2},\, \boldsymbol{Y}_{t1}=\boldsymbol{A}\boldsymbol{Q}_{t1}}, \tag{3.2.10}$$

where $\overset{d}{=}$ represents equal in distribution. As long as $\boldsymbol{A}|_{\mathfrak{G}_{t1,t2}}$ is achieved, the corresponding conditional distributions of $\boldsymbol{b}^t|_{\mathfrak{G}_{t,t}}$ and $\boldsymbol{h}^{t+1}|_{\mathfrak{G}_{t+1,t}}$ will be calculated and the Gaussianity can be checked.

3.3 AMP for C-DCS

The original AMP algorithm can be extended for the C-DCS model based on (3.2.4) where both the measurement matrices and the signals can be correlated. We call the derived algorithm as AMP-C-DCS which estimates the unknown signal via

$$\begin{aligned}
\boldsymbol{x}_B^{t+1} &= \boldsymbol{\eta}_t\left(\boldsymbol{A}_B^T\boldsymbol{r}_B^t + \boldsymbol{x}_B^t\right), \\
\boldsymbol{r}_B^{t+1} &= \boldsymbol{y}_B - \boldsymbol{A}_B\boldsymbol{x}_B^{t+1} + \frac{1}{\delta}\left(\boldsymbol{I}_m \otimes \left(\boldsymbol{\Sigma}_A \odot \boldsymbol{D}^t\right)\right)\boldsymbol{r}_B^t,
\end{aligned} \tag{3.3.1}$$

where $\otimes$ denotes the Kronecker product, $\odot$ stands for the component wise multiplication, $\boldsymbol{x}_{:,i}^{t+1} = \boldsymbol{\eta}_t\left(\tilde{\boldsymbol{x}}_{:,i}^t\right)$ and $\boldsymbol{D}^t = \frac{1}{n}\sum_{i=1}^n \boldsymbol{\eta}_t'\left(\tilde{\boldsymbol{x}}_{:,i}^t\right)$ where $\boldsymbol{\eta}_t'(\boldsymbol{v}_i) \coloneqq \frac{\partial \boldsymbol{\eta}_t(\boldsymbol{v}_i)}{\partial \boldsymbol{v}}$ denoting the Jacobian matrix of $\boldsymbol{\eta}_t(\boldsymbol{v}_i) : \mathbb{R}^K \to \mathbb{R}^K$. We provide a heuristic derivation of AMP-C-DCS algorithm in Section 3.6.1.

The state evolution technique can also be applied for the AMP-C-DCS algorithm, but this time, we are tracking the covariance matrix of the equivalent noise rather than a scalar value as in (3.2.6). The corresponding equation changes to

$$\boldsymbol{\Sigma}_e^t = \frac{1}{\delta}\left(\boldsymbol{\Sigma}_A\right).^2 \odot E\left[\left(\boldsymbol{\eta}_t\left(\boldsymbol{X} + \boldsymbol{Z}_e^{t-1}\right) - \boldsymbol{X}\right)^2\right] + \sigma_w^2\boldsymbol{I}. \tag{3.3.2}$$

where $\boldsymbol{Z}_e^{t-1} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_e^{t-1}\right)$ is independent of $\boldsymbol{X} \in \mathbb{R}^K$ with initial value $\boldsymbol{\Sigma}_e^0 = \frac{n}{m}\left(\boldsymbol{\Sigma}_A\right).^2 \odot \boldsymbol{\Sigma}_X + \sigma_w^2 \boldsymbol{I}$ where $\boldsymbol{\Sigma}_X$ is the covariance matrix of the random vector $\boldsymbol{X}$ (i.e. the super component of the unknown signal as in Definition 3.2.1). Equation (3.3.2) is achieved based on the equivalent noise (2.2.2) and the following theorem.

Theorem 3.3.1. *Let $\boldsymbol{u} = \left(\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I}\right)\boldsymbol{v} + \boldsymbol{A}^T\boldsymbol{w}$, where $\boldsymbol{A} \in \mathbb{R}^{Km \times Kn}$ is a block matrix as in Definition 3.2.2 with mean zero and covariance matrix $\frac{1}{m}\boldsymbol{\Sigma}_A$, $\boldsymbol{v} \in \mathbb{R}^{Kn}$ is a block random vector as in Definition 3.2.1 with mean vector zero and covariance matrix $\boldsymbol{\Sigma}_v$. $\boldsymbol{w} \in \mathbb{R}^{Km}$ is the white Gaussian noise with σ_w^2, and $\boldsymbol{A}$,$\boldsymbol{v}$ and $\boldsymbol{w}$ are mutually independent. Then the vector $\boldsymbol{u} \in \mathbb{R}^{Kn}$ is also a block vector and the covariance matrix of each super-component $\boldsymbol{u}_{:,j} \in \mathbb{R}^K$, $j \in [n]$ can be calculated by the following equation*

$$\boldsymbol{\Sigma}_u = \frac{n}{m}\left(\boldsymbol{\Sigma}_A\right).^2 \odot \boldsymbol{\Sigma}_v + \sigma_w^2 \boldsymbol{I}, \tag{3.3.3}$$

where $(\cdot).^2$ means component-wise square operation, $\odot$ stands for the component-wise multiplication.

Proof. Consider the $\mathrm{B}-$model of the system, the elements of $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ and $\boldsymbol{A}$ are all super components. Let $\boldsymbol{H}_{j,l} \in \mathbb{R}^{K\times K}$ be the j,l-th super component of $\left(\boldsymbol{A}^T\boldsymbol{A} - \boldsymbol{I}\right) \in \mathbb{R}^{Kn\times Kn}$ with diagonal vector $\boldsymbol{h}_{:,j,l} \in \mathbb{R}^K$ as in Remark 3.2.3. Just like the original AMP algorithm, we can treat $\boldsymbol{h}_{:,j,l}$'s for $j \neq l,\ j,l \in [n]$ as approximately i.i.d. Gaussian vectors with mean $\boldsymbol{0}$ and covariance matrix

$$\begin{aligned}\mathrm{Cov}\left(\boldsymbol{h}_{:,j,l}\right) &= \sum_{i=1}^{m} \mathrm{Cov}\left(\boldsymbol{A}_{i,j}\boldsymbol{a}_{:,i,l}\right), \\ &\overset{a}{=} \frac{1}{m}\left(\boldsymbol{\Sigma}_A\right).^2,\end{aligned}$$

where $\boldsymbol{a}_{:,i,l} \in \mathbb{R}^K$ is the diagonal vector of $\boldsymbol{A}_{i,l} \in \mathbb{R}^{K\times K}$, and $\overset{a}{=}$ holds by Lemma 3.3.2 listed below. Next, we calculate the covariance matrix of the super component $\boldsymbol{u}_{:,j} \in \mathbb{R}^K$, $j \in [n]$:

$$\begin{aligned}\mathrm{Cov}\left(\boldsymbol{u}_{:,j}\right) &= \mathrm{Cov}\left(\sum_{l=1}^{n} \boldsymbol{H}_{j,l}\boldsymbol{v}_{:,l} + \sum_{i=1}^{m} \boldsymbol{A}_{i,j}\boldsymbol{w}_{:,i}\right), \\ &= \sum_{l=1}^{n} \mathrm{Cov}\left(\boldsymbol{H}_{j,l}\boldsymbol{v}_{:,l}\right) + \sum_{i=1}^{m} \mathrm{Cov}\left(\boldsymbol{A}_{i,j}\boldsymbol{w}_{:,i}\right), \\ &= \frac{n}{m}\left(\boldsymbol{\Sigma}_A\right).^2 \odot \boldsymbol{\Sigma}_v + \sigma_w^2 \boldsymbol{I},\end{aligned}$$

which provides (3.3.3). □

Lemma 3.3.2. *Let $\boldsymbol{u} = \mathrm{diag}\left(\boldsymbol{h}\right)\boldsymbol{v}$ where $\boldsymbol{h}, \boldsymbol{v} \in \mathbb{R}^K$ are two independent random vectors with mean zero and covariance matrices $\boldsymbol{\Sigma}_h$ and $\boldsymbol{\Sigma}_v$ respectively. Then the vector $\boldsymbol{u}$ is of mean zero and covariance matrix $\boldsymbol{\Sigma}_h \odot \boldsymbol{\Sigma}_v$.*

For the denoising function, based on the decoupled model in (2.2.1), we have

$$\tilde{\boldsymbol{x}}_B^t = \boldsymbol{x}_B + \boldsymbol{w}_{B,e}^t \tag{3.3.4}$$

(B − model is used for joint estimation), at t-th iteration, the MMSE estimator will act on each super component individually

$$\boldsymbol{x}_{:,i}^{t+1} = \boldsymbol{\eta}_t\left(\tilde{\boldsymbol{x}}_{:,i}^t\right) \coloneqq \mathrm{E}_{x_{:,i}|\tilde{x}_{:,i}^t}\left[\boldsymbol{x}_{:,i}|\tilde{\boldsymbol{x}}_{:,i}^t\right], \tag{3.3.5}$$

based on the statistic information of the equivalent noise $\boldsymbol{w}_e^t$. The covariance matrix ($\boldsymbol{\Sigma}_e^t$) of the equivalent noise can be computed by Theorem 3.3.1.

Then based on the following equation (see section 3.6.7 for the proof):

$$\begin{aligned}\boldsymbol{\Sigma}_{\eta}^{t+1} &= \boldsymbol{\Psi}\left(\boldsymbol{\Sigma}_{e}^{t}\right) \\ &\coloneqq \mathrm{E}_{\tilde{x}_{:,i}^{t}}\left[\mathrm{E}_{x_{:,i}|\tilde{x}_{:,i}^{t}}\left[\boldsymbol{x}_{:,i}\boldsymbol{x}_{:,i}^{T}|\tilde{\boldsymbol{x}}_{:,i}^{t}\right]\right] - \mathrm{E}_{\tilde{x}_{:,i}^{t}}\left[\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}_{:,i}^{t}\right)\boldsymbol{\eta}^{T}\left(\tilde{\boldsymbol{x}}_{:,i}^{t}\right)\right], \qquad (3.3.6)\end{aligned}$$

we are able to estimate the covariance matrix $\boldsymbol{\Sigma}_{\eta}^{t+1}$ of super component $\left(\boldsymbol{x}_{:,i} - \boldsymbol{x}_{:,i}^{t+1}\right)$ which contains the statistic information of the estimated error by function $\boldsymbol{\eta}_t(\cdot)$ in (3.3.5). The matrix $\boldsymbol{\Sigma}_{\eta}^{t+1}$ then can be used to update $\boldsymbol{\Sigma}_{e}^{t+1}$ of the equivalent noise at the next iteration.

Our approach is significantly different from that in [66]. There, the so-called AMP-MMV algorithm was proposed for the MMV model based on the authors' previous work the turbo-AMP in [79]. Though the signal correlations are considered by adding an extra layer for the signal components in the factor graph, there is no inherent part in their algorithm to handle the correlations in the measurement matrices. By contrast, we extend the original AMP by grouping the correlated signal/matrix components to form super components. The derivations of the denoising function $\boldsymbol{\eta}_t(\cdot)$ and the Onsager term are based on the super components.

(Notice: signal estimation is based on (3.3.4) and $\boldsymbol{\Sigma}_A$ only affects the equivalent noise (3.3.2). For the special case when $\boldsymbol{\Sigma}_A = \boldsymbol{I}$, the elements in each super component of the equivalent noise are uncorrelated due to (3.3.2), but the correlations between signals are always required during the estimation of signals at each iteration.)

3.4 Correctness Justification of AMP-C-DCS

3.4.1 Gaussian Conditional Lemma

In the rigorous proof of [29], the Gaussian conditional lemma is the fundamental technique for the whole proof. In our case, it can be written as

Lemma 3.4.1. *[29, Extension of Lemma 11] Let $\boldsymbol{z} \in \mathbb{R}^n$ be a random vector with $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_z)$, and let $\boldsymbol{D} \in \mathbb{R}^{m \times n}$ be a linear operator with full row rank. Then for any constant vector $\boldsymbol{b} \in \mathbb{R}^m$, the distribution of $\boldsymbol{z}$ conditioned on $\boldsymbol{D}\boldsymbol{z} = \boldsymbol{b}$ satisfies:*

$$\boldsymbol{z}|_{\boldsymbol{D}\boldsymbol{z}=\boldsymbol{b}} \overset{d}{=} \boldsymbol{\Sigma}_z \boldsymbol{D}^T \left(\boldsymbol{D}\boldsymbol{\Sigma}_z \boldsymbol{D}^T\right)^{-1} \boldsymbol{b} + \mathcal{P}\ (\tilde{\boldsymbol{z}})$$

where $\tilde{\boldsymbol{z}}$ is an independent copy of $\boldsymbol{z}$, $\mathcal{P}\ (\tilde{\boldsymbol{z}}) \coloneqq \left(\boldsymbol{I} - \boldsymbol{\Sigma}_z \boldsymbol{D}^T \left(\boldsymbol{D}\boldsymbol{\Sigma}_z \boldsymbol{D}^T\right)^{-1} \boldsymbol{D}\right) \tilde{\boldsymbol{z}}$ and $\mathcal{P}\ \coloneqq \boldsymbol{I} - \boldsymbol{\Sigma}_z \boldsymbol{D}^T \left(\boldsymbol{D}\boldsymbol{\Sigma}_z \boldsymbol{D}^T\right)^{-1} \boldsymbol{D}$ is a projector.

Proof. The above lemma can be achieved via the standard version of Gaussian conditional lemma (Lemma 3.4.2) listed below. For the mean value, we let $\boldsymbol{\Sigma}_w \to \mathbf{0}$ which gives $\boldsymbol{\Sigma}_z \boldsymbol{D}^T \left(\boldsymbol{D}\boldsymbol{\Sigma}_z \boldsymbol{D}^T\right)^{-1} \boldsymbol{b}$. For the covariance matrix, we have $\boldsymbol{\Sigma}_z - \boldsymbol{\Sigma}_z \boldsymbol{D}^T \left(\boldsymbol{D}\boldsymbol{\Sigma}_z \boldsymbol{D}^T\right)^{-1} \boldsymbol{\Sigma}_z = \mathcal{P}\ \boldsymbol{\Sigma}_z$. In addition, $\mathrm{Cov}\left(\mathcal{P}\ (\tilde{\boldsymbol{z}})\right) = \mathcal{P}\ \boldsymbol{\Sigma}_z \mathcal{P}^{\ T} = \mathcal{P}\ \boldsymbol{\Sigma}_z$. The above lemma is proved. □

Lemma 3.4.2. *Given $\boldsymbol{b} = \boldsymbol{D}\boldsymbol{z} + \boldsymbol{w}$ where $\boldsymbol{D} \in \mathbb{R}^{m \times n}$ is a deterministic matrix, $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_z)$ and $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_w)$ are Gaussian random vectors. Assume all*

the matrix inverses exist, we have the following consequences

$$\begin{aligned}
\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{b}} &= \boldsymbol{\Sigma}_z \boldsymbol{D}^T \left(\boldsymbol{D} \boldsymbol{\Sigma}_z \boldsymbol{D}^T + \boldsymbol{\Sigma}_w \right)^{-1} \boldsymbol{b} \\
&= \left(\boldsymbol{\Sigma}_z^{-1} + \boldsymbol{D}^T \boldsymbol{\Sigma}_w^{-1} \boldsymbol{D} \right)^{-1} \boldsymbol{D}^T \boldsymbol{\Sigma}_w^{-1} \boldsymbol{b}, \\
\boldsymbol{\Sigma}_{\boldsymbol{z}|\boldsymbol{b}} &= \left(\boldsymbol{\Sigma}_z^{-1} + \boldsymbol{D}^T \boldsymbol{\Sigma}_w^{-1} \boldsymbol{D} \right)^{-1} \\
&= \boldsymbol{\Sigma}_z - \boldsymbol{\Sigma}_z \boldsymbol{D}^T \left(\boldsymbol{D} \boldsymbol{\Sigma}_z \boldsymbol{D}^T + \boldsymbol{\Sigma}_w \right)^{-1} \boldsymbol{\Sigma}_z
\end{aligned}$$

where $\boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{b}}$ is the conditional mean and $\boldsymbol{\Sigma}_{\boldsymbol{z}|\boldsymbol{b}}$ is the conditional covariance matrix.

Proof. The proof can be found in many books talking about multivariate Gaussian density or related papers [80]. □

In [29, Lemma 11], $\boldsymbol{\Sigma}_z = \boldsymbol{I}\sigma_z^2$ thus in their case, $\mathcal{P}(\tilde{\boldsymbol{z}}) = \mathcal{P}_{\{\boldsymbol{D}\boldsymbol{z}=\boldsymbol{0}\}}$ is the orthogonal projection onto the subspace $\{\boldsymbol{D}\boldsymbol{z} = \boldsymbol{0}\}$. For the correlated case where $\boldsymbol{\Sigma}_z \neq \boldsymbol{I}\sigma_z^2$, $\mathcal{P}(\tilde{\boldsymbol{z}})$ is no longer an orthogonal projection and usually the statistic information contained by $\mathcal{P}(\tilde{\boldsymbol{z}})$ is much more complicated compared with the orthogonal projection case. See the following as an example.

Example 3.4.3. Consider the case in which $K = 2$. Let $\boldsymbol{D}_1 = \begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{0}_2 & \boldsymbol{0}_2 \end{bmatrix}$, $\boldsymbol{D}_2 = \begin{bmatrix} \boldsymbol{0}_2 & \boldsymbol{I}_2 & \boldsymbol{0}_2 \end{bmatrix}$ where $\boldsymbol{I}_2 \in \mathbb{R}^{2\times 2}$ is the identity matrix and $\boldsymbol{0}_2 \in \mathbb{R}^{2\times 2}$ contains all zero elements. Assume $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^2$ is given, $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{R}^6$ are two Gaussian vectors with across correlation among super components $\boldsymbol{z}_{:,i} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \in \mathbb{R}^{2\times 2}$. We can write the compact linear

equations as

$$\underbrace{\begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix}}_{\boldsymbol{b}} = \underbrace{\begin{bmatrix} \boldsymbol{D}_1 & \\ & \boldsymbol{D}_2 \end{bmatrix}}_{\boldsymbol{D}} \underbrace{\begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix}}_{\boldsymbol{z}}$$

with $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_z)$ and $\boldsymbol{\Sigma}_z = \boldsymbol{\Sigma} \otimes \boldsymbol{I}_6$. This function is written in the D − model but in order to apply Gaussian conditional lemma, we need to write $\boldsymbol{b}$ and $\boldsymbol{z}$ in the vector form. Based on Lemma 3.4.1, we achieve the covariance matrix after estimation

$$\begin{aligned} \boldsymbol{\Sigma}_P &= \boldsymbol{\Sigma}_z - \boldsymbol{\Sigma}_z \boldsymbol{D}^T \left(\boldsymbol{D} \boldsymbol{\Sigma}_z \boldsymbol{D}^T\right)^{-1} \boldsymbol{D} \boldsymbol{\Sigma}_z \\ &= \begin{bmatrix} \boldsymbol{0}_2 & & & \boldsymbol{0}_2 & & \\ & (1-\rho)\boldsymbol{I}_2 & & & \boldsymbol{0}_2 & \\ & & \boldsymbol{I}_2 & & & \rho \boldsymbol{I}_2 \\ \boldsymbol{0}_2 & & & (1-\rho)\boldsymbol{I}_2 & & \\ & \boldsymbol{0}_2 & & & \boldsymbol{0}_2 & \\ & & \rho \boldsymbol{I}_2 & & & \boldsymbol{I}_2 \end{bmatrix} \end{aligned} \tag{3.4.1}$$

The blue elements in (3.4.1) show that the covariances between corresponding elements in $\boldsymbol{z}$ after estimation can still be represented by $\boldsymbol{\Sigma}$, while the red elements can't, which makes the analysis of state evolution complicated. The exceptions are the cases when $\rho = 1$ and $\rho = 0$.

Next, we need to apply Lemma 3.4.1 to the Gaussian matrix $\boldsymbol{A}$ with restrictions $\boldsymbol{X} = \boldsymbol{A}^T \boldsymbol{M}$, $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{Q}$ as in (3.2.10). But this extension is not straightforward due to the structure of $\boldsymbol{A}$. For simplicity, let's consider only one restriction $\boldsymbol{X} = \boldsymbol{A}^T \boldsymbol{M}$ or equivalently $\boldsymbol{X}^T = \boldsymbol{M}^T \boldsymbol{A}$. The matrix $\boldsymbol{A} = \text{diag}(\boldsymbol{A}_1, ..., \boldsymbol{A}_K)$ has a diagonal structure (consider D − model). If we directly estimate $\boldsymbol{A}$ column by column, we eventually estimate $\boldsymbol{A}_1$, $\boldsymbol{A}_2$,...,$\boldsymbol{A}_K$ separately without considering the correlation effects (described by $\boldsymbol{\Sigma}_A$, here

we ignore the normalization factor $\frac{1}{m}$) between them. The correct way is to jointly estimate i-th column form all $\boldsymbol{A}_k$'s for $i \in [n]$, $k \in [K]$. In order to apply joint estimation, we define the reshuffle and inverse reshuffle operations as follows

Definition 3.4.4. For any block diagonal matrix $\boldsymbol{B}$, define the reshuffle τ_1/τ_2 and the corresponding inverse operations τ_1^{-1}/τ_2^{-1} as follows

$$\underbrace{\begin{bmatrix} \boldsymbol{B}_1 & & \\ & \ddots & \\ & & \boldsymbol{B}_K \end{bmatrix}}_{\boldsymbol{B}} \underset{\tau_1^{-1}(\boldsymbol{B}_C)}{\overset{\tau_1(\boldsymbol{B})}{\rightleftharpoons}} \underbrace{\begin{bmatrix} \boldsymbol{B}_1 \\ \vdots \\ \boldsymbol{B}_K \end{bmatrix}}_{\boldsymbol{B}_C} \tag{3.4.2}$$

$$\underbrace{\begin{bmatrix} \boldsymbol{B}_1 & & \\ & \ddots & \\ & & \boldsymbol{B}_K \end{bmatrix}}_{\boldsymbol{B}} \underset{\tau_2^{-1}(\boldsymbol{B}_R)}{\overset{\tau_2(\boldsymbol{B})}{\rightleftharpoons}} \underbrace{\begin{bmatrix} \boldsymbol{B}_1 & \cdots & \boldsymbol{B}_K \end{bmatrix}}_{\boldsymbol{B}_R} \tag{3.4.3}$$

Thus the mean value of $\boldsymbol{A}|_{\boldsymbol{X}=\boldsymbol{A}^T\boldsymbol{M}}$ is calculated via $\tau_1^{-1}\left(\boldsymbol{\Sigma}_m\boldsymbol{M}\left(\boldsymbol{M}^T\boldsymbol{\Sigma}_m\boldsymbol{M}\right)^{-1}\tau_1\left(\boldsymbol{X}^T\right)\right)$ where $\boldsymbol{\Sigma}_m = \boldsymbol{\Sigma}_A \otimes \boldsymbol{I}_m$ and $\otimes$ is the Kronecker product and the corresponding mean value of $\boldsymbol{A}|_{\boldsymbol{Y}=\boldsymbol{A}\boldsymbol{Q}}$ is $\tau_2^{-1}\left(\tau_2\left(\boldsymbol{Y}\right)\left(\boldsymbol{Q}^T\boldsymbol{\Sigma}_n\boldsymbol{Q}\right)^{-1}\boldsymbol{Q}^T\boldsymbol{\Sigma}_n\right)$ where $\boldsymbol{\Sigma}_n = \boldsymbol{\Sigma}_A \otimes \boldsymbol{I}_n$. Based on [29, Lemma 10], we can achieve the extended version for our case.

Lemma 3.4.5. *[29, Extension of Lemma 10] Given two linear systems $\boldsymbol{Y}_{t1} = \boldsymbol{A}\boldsymbol{Q}_{t1}$ and $\boldsymbol{X}_{t2} = \boldsymbol{A}^T\boldsymbol{M}_{t2}$ where each of them contains K measurement instances denoted by $\boldsymbol{Y}_{k,t1} = \boldsymbol{A}_k\boldsymbol{Q}_{k,t1}$, $\boldsymbol{X}_{k,t2} = \boldsymbol{A}_k^T\boldsymbol{M}_{k,t2}$ for $k \in [K]$, respectively. Define $\boldsymbol{\Sigma}_A$ as the covariance matrix of super components of $\boldsymbol{A}$. The conditional distribution of the random matrix $\boldsymbol{A}$ given the σ-algebra $\mathfrak{G}_{t1,t2}$ will*

be

$$\boldsymbol{A}|_{\mathfrak{G}_{t1,t2}} \stackrel{d}{=} \boldsymbol{E}_{t1,t2} + \boldsymbol{\mathcal{P}}_{t1,t2}\left(\tilde{\boldsymbol{A}}\right),$$

where $\tilde{\boldsymbol{A}} \stackrel{d}{=} \boldsymbol{A}$ *is a random matrix independent of* $\mathfrak{G}_{t1,t2}$ *and*

$$\begin{aligned}\boldsymbol{E}_{t1,t2} = {} & \tau_2^{-1}\left(\tau_2\left(\boldsymbol{Y}_{t1}\right)\left(\boldsymbol{Q}_{t1}^T\boldsymbol{\Sigma}_n\boldsymbol{Q}_{t1}\right)^{-1}\boldsymbol{Q}_{t1}^T\boldsymbol{\Sigma}_n\right) \\ & + \tau_1^{-1}\left(\boldsymbol{\Sigma}_m\boldsymbol{M}_{t2}\left(\boldsymbol{M}_{t2}^T\boldsymbol{\Sigma}_m\boldsymbol{M}_{t2}\right)^{-1}\tau_1\left(\boldsymbol{X}_{t2}^T\right)\right) \\ & - \tau_1^{-1}\left(\boldsymbol{\Sigma}_m\boldsymbol{M}_{t2}\left(\boldsymbol{M}_{t2}^T\boldsymbol{\Sigma}_m\boldsymbol{M}_{t2}\right)^{-1}\right. \\ & \left.\times\tau_1\left(\boldsymbol{M}_{t2}^T\tau_2^{-1}\left(\tau_2\left(\boldsymbol{Y}_{t1}\right)\left(\boldsymbol{Q}_{t1}^T\boldsymbol{\Sigma}_n\boldsymbol{Q}_{t1}\right)^{-1}\boldsymbol{Q}_{t1}^T\boldsymbol{\Sigma}_n\right)\right)\right)\end{aligned} \tag{3.4.4}$$

and

$$\boldsymbol{\mathcal{P}}_{t1,t2}\left(\tilde{\boldsymbol{A}}\right) = \tau_2^{-1}\left(\tau_2\left(\tau_1^{-1}\left(\boldsymbol{P}_{M_{t2}}^{\nparallel}\tau_1\left(\tilde{\boldsymbol{A}}\right)\right)\right)\boldsymbol{P}_{Q_{t1}}^{\nparallel T}\right) \tag{3.4.5}$$

where $\boldsymbol{P}_{M_{t2}}^{\nparallel} = \boldsymbol{I} - \boldsymbol{P}_{M_{t2}}^{\parallel}$ *and* $\boldsymbol{P}_{Q_{t1}}^{\nparallel} = \boldsymbol{I} - \boldsymbol{P}_{Q_{t1}}^{\parallel}$ *and* $\boldsymbol{P}_{M_{t2}}^{\parallel} = \boldsymbol{\Sigma}_m\boldsymbol{M}_{t2}\left(\boldsymbol{M}_{t2}^T\boldsymbol{\Sigma}_m\boldsymbol{M}_{t2}\right)^{-1}\boldsymbol{M}_{t2}^T$ *and* $\boldsymbol{P}_{Q_{t1}}^{\parallel} = \boldsymbol{\Sigma}_n\boldsymbol{Q}_{t1}\left(\boldsymbol{Q}_{t1}^T\boldsymbol{\Sigma}_n\boldsymbol{Q}_{t1}\right)^{-1}\boldsymbol{Q}_{t1}^T$ *are two projectors.*

According to example 3.4.3 and the reshuffle operations listed in Lemma 3.4.5, we found the analysis of tracking the statistics of $\boldsymbol{A}$ is extremely complicated. Thus, in the following analysis, we focus on the two special cases: 1) independent case where $\boldsymbol{\Sigma}_A = \boldsymbol{I}$ and 2) the identical case where $\boldsymbol{\Sigma}_A = \boldsymbol{1}$. We can achieve the following corollary.

Corollary 3.4.6. *For the two special cases: (1)* $\boldsymbol{A}_k$*'s are independent, (2)*

$\boldsymbol{A}_k$'s are identical, $\boldsymbol{E}_{t1,t2}$ and $\boldsymbol{\mathcal{P}}_{t1,t2}\left(\tilde{\boldsymbol{A}}\right)$ can be simplified as

$$
\begin{aligned}
&\boldsymbol{E}_{t1,t2} \\
&= \begin{cases}
\boldsymbol{Y}_{t1}\left(\boldsymbol{Q}_{t1}^T\boldsymbol{Q}_{t1}\right)^{-1}\boldsymbol{Q}_{t1}^T + \boldsymbol{M}_{t2}\left(\boldsymbol{M}_{t2}^T\boldsymbol{M}_{t2}\right)^{-1}\boldsymbol{X}_{t2}^T \\
-\boldsymbol{M}_{t2}\left(\boldsymbol{M}_{t2}^T\boldsymbol{M}_{t2}\right)^{-1}\boldsymbol{M}_{t2}^T\boldsymbol{Y}_{t1}\left(\boldsymbol{Q}_{t1}^T\boldsymbol{Q}_{t1}\right)^{-1}\boldsymbol{Q}_{t1}^T & \text{if } \boldsymbol{\Sigma}_A = \boldsymbol{I} \\
\\
\boldsymbol{I}_K \otimes \left(\boldsymbol{Y}_{R,t1}\left(\boldsymbol{Q}_{R,t1}^T\boldsymbol{Q}_{R,t1}\right)^{-1}\boldsymbol{Q}_{R,t1}^T\right) \\
+\boldsymbol{I}_K \otimes \left(\boldsymbol{M}_{R,t2}\left(\boldsymbol{M}_{R,t2}^T\boldsymbol{M}_{R,t2}\right)^{-1}\boldsymbol{X}_{R,t2}^T\right) \\
-\boldsymbol{I}_K \otimes \left(\boldsymbol{M}_{R,t2}\left(\boldsymbol{M}_{R,t2}^T\boldsymbol{M}_{R,t2}\right)^{-1}\boldsymbol{M}_{R,t2}^T\right. \\
\left.\times\, \boldsymbol{Y}_{R,t1}\left(\boldsymbol{Q}_{R,t1}^T\boldsymbol{Q}_{R,t1}\right)^{-1}\boldsymbol{Q}_{R,t1}^T\right) & \text{if } \boldsymbol{\Sigma}_A = \boldsymbol{1}
\end{cases}
\end{aligned}
$$

and

$$
\boldsymbol{\mathcal{P}}_{t1,t2}\left(\tilde{\boldsymbol{A}}\right) = \begin{cases}
\boldsymbol{P}_{M_{t2}}^{\perp}\left(\tilde{\boldsymbol{A}}\right)\boldsymbol{P}_{Q_{t1}}^{\perp} & \text{if } \boldsymbol{\Sigma}_A = \boldsymbol{I} \\
\boldsymbol{I}_K \otimes \left(\boldsymbol{P}_{M_{R,t2}}^{\perp}\tilde{\boldsymbol{A}}_I\boldsymbol{P}_{Q_{R,t1}}^{\perp}\right) & \text{if } \boldsymbol{\Sigma}_A = \boldsymbol{1}
\end{cases}
$$

where $\otimes$ denotes the Kronecker product, $\tilde{\boldsymbol{A}}$ represents the independent copy of $\boldsymbol{A}$, and $\boldsymbol{A}_I \coloneqq \boldsymbol{A}_1 = ... = \boldsymbol{A}_K$ denotes the identical matrix inside of $\boldsymbol{A}$ for the identical case. $\boldsymbol{P}_{\boldsymbol{V}}^{\perp} = \boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{V}}$, and $\boldsymbol{P}_{\boldsymbol{V}}$ is the orthogonal projector onto the column space of matrix $\boldsymbol{V}$.

Proof. The proofs of the two special cases are given in section 3.6.2 and 3.6.3.

□

3.4.2 Special Cases Analysis

In order to justify the correctness of our proposed algorithm, we need go back to the general form as (3.2.7). The corresponding formulas (apply B − model)

based on (3.3.1) are

$$\begin{aligned} \boldsymbol{h}^{t+1} &= \boldsymbol{A}^T \boldsymbol{m}^t - (\boldsymbol{I} \otimes \boldsymbol{\xi}_t)\, \boldsymbol{q}^t,\ \boldsymbol{m}^t = \boldsymbol{g}_t\left(\boldsymbol{b}^t, \boldsymbol{w}\right), \\ \boldsymbol{b}^t &= \boldsymbol{A}\boldsymbol{q}^t - (\boldsymbol{I} \otimes \boldsymbol{\lambda}_t)\, \boldsymbol{m}^{t-1},\ \boldsymbol{q}^t = \boldsymbol{f}_t\left(\boldsymbol{h}^t, \boldsymbol{x}\right), \end{aligned} \tag{3.4.6}$$

where $\boldsymbol{\xi}_t = \langle \boldsymbol{g}_t'\left(\boldsymbol{b}^t, \boldsymbol{w}\right)\rangle$, $\boldsymbol{\lambda}_t = \frac{1}{\delta}\boldsymbol{\Sigma}_A \odot \langle \boldsymbol{f}_t'\left(\boldsymbol{h}^t, \boldsymbol{x}\right)\rangle \in \mathbb{R}^{K\times K}$ are two matrices, $\boldsymbol{g}_t'$, $\boldsymbol{f}_t'$ are Jacobian matrices of $\boldsymbol{g}_t\left(\boldsymbol{b}^t, \boldsymbol{w}\right) = \boldsymbol{b}^t - \boldsymbol{w}$, $\boldsymbol{f}_t\left(\boldsymbol{h}^t, \boldsymbol{x}\right) = \boldsymbol{\eta}_t\left(\boldsymbol{x} - \boldsymbol{h}^t\right) - \boldsymbol{x}$ with respect to the first argument, respectively, and $\odot$ denotes the element-wise production. $\mathrm{B}-$model is preferred here, because both $\boldsymbol{g}_t$ and $\boldsymbol{f}_t$ are element-wise operations applied for each super component, thus the connect between (3.4.6) and (3.2.7) is more explicit. The inputs of $\boldsymbol{g}_t$ and $\boldsymbol{f}_t$ are (super components) always treated as vectors. In addition, $\boldsymbol{\xi}_t$ and $\boldsymbol{\lambda}_t$ are two matrices that multiplied by each super component of $\boldsymbol{q}^t$ and $\boldsymbol{m}^t$. The direct analysis based on (3.4.6) is not adequate as the Gaussian conditional lemma suffers from reshuffle operations and the analysis is extremely complicated. But for the two special cases, (3.4.6) can be simplified by removing the reshuffle operations.

For the independent case, where $\boldsymbol{\Sigma}_A$ will be an identity matrix, thus both $\boldsymbol{\xi}_t$ and $\boldsymbol{\lambda}_t$ are diagonal matrices. In addition, based on the Corollary 3.4.6, there is no need for joint estimation and we can compute $\boldsymbol{A}_k|_{\boldsymbol{X}_{k,t2}=\boldsymbol{A}_k^T\boldsymbol{M}_{k,t2},\boldsymbol{Y}_{k,t1}=\boldsymbol{A}_k\boldsymbol{Q}_{k,t1}}$ separately for each $k \in [K]$. Under this condition, we can rewrite (3.4.6) based on $\mathrm{D}-$model. For each measurement instance $k \in [K]$, we have

$$\begin{aligned} \boldsymbol{h}_k^{t+1} &= \boldsymbol{A}_k^T \boldsymbol{m}_k^t - \xi_{k,t}\boldsymbol{q}_k^t,\ \boldsymbol{m}_k^t = \left[\boldsymbol{g}_t\left(\boldsymbol{b}^t, \boldsymbol{w}\right)\right]_k, \\ \boldsymbol{b}_k^t &= \boldsymbol{A}_k\boldsymbol{q}_k^t - \lambda_{k,t}\boldsymbol{m}_k^{t-1},\ \boldsymbol{q}_k^t = \left[\boldsymbol{f}_t\left(\boldsymbol{h}^t, \boldsymbol{x}\right)\right]_k, \end{aligned} \tag{3.4.7}$$

where $\xi_{k,t}$ and $\lambda_{k,t}$ represent the k-th diagonal elements of $\boldsymbol{\xi}_t$ and $\boldsymbol{\lambda}_t$ respec-

tively. The notations in (3.4.7) have the same definition as those in (3.2.7) except $\boldsymbol{g}_t$ and $\boldsymbol{f}_t$ operate on the super components. Although joint estimation is applied for $\boldsymbol{m}^t$ and $\boldsymbol{q}^t$, the correlations across $\boldsymbol{m}_k^t$'s and $\boldsymbol{q}_k^t$'s are eventually ignored by the effect of independent measurement matrices $\boldsymbol{A}_k$'s which resulting independent $\boldsymbol{h}_k^{t+1}$'s and $\boldsymbol{b}_k^t$'s (independent Gaussian noise). This property is reflected in (3.3.2) by letting $\boldsymbol{\Sigma}_A = \boldsymbol{I}$. Thus (3.4.7) should have a similar behaviour as (3.2.7). In section 3.6.4 we provide an illustration of proving that the conditional distribution $\boldsymbol{h}^{t+1}$ and $\boldsymbol{b}^t$ will always be Gaussian for $t \geq 0$ by the effects of $(\boldsymbol{I} \otimes \boldsymbol{\xi}_t)\, \boldsymbol{q}^t$ and $(\boldsymbol{I} \otimes \boldsymbol{\lambda}_t)\, \boldsymbol{m}^{t-1}$.

For the identical case, where $\boldsymbol{\Sigma}_A$ is a matrix with all 1 value. Based on the Corollary 3.4.6, we only need to compute the statistics for one measurement matrix $\boldsymbol{A}_I$, and the corresponding two constraints can be replaced by $\boldsymbol{Y}_{R,t1} = \boldsymbol{A}_I \boldsymbol{Q}_{R,t1}$ and $\boldsymbol{X}_{R,t2} = \boldsymbol{A}_I^T \boldsymbol{M}_{R,t2}$. Recall that

$$\boldsymbol{M}_{R,t2} = [\boldsymbol{M}_{1,t2},\, \boldsymbol{M}_{2,t2}, ..., \boldsymbol{M}_{K,t2}]$$

and based on (3.2.9) each $\boldsymbol{M}_{k,t2} = \left[\boldsymbol{m}_k^0 | \ldots | \boldsymbol{m}_k^{t-1}\right]$. Rearrange the columns of $\boldsymbol{M}_{R,t2}$ to form the following form

$$\left[\boldsymbol{m}_1^0 \ldots \boldsymbol{m}_K^0 | \cdots | \boldsymbol{m}_1^{t-1} \ldots \boldsymbol{m}_K^{t-1}\right] =: \left[\boldsymbol{m}_r^0 | \boldsymbol{m}_r^2 | ... | \boldsymbol{m}_r^{t-1}\right]$$

where $\boldsymbol{m}_r^t = [\boldsymbol{m}_1^t | \ldots | \boldsymbol{m}_K^t]$. Apply the corresponding rearrangement for $\boldsymbol{X}_{R,t2}$ (the same as $\boldsymbol{Y}_{R,t1}$ and $\boldsymbol{Q}_{R,t1}$), will change (3.4.6) to

$$\begin{aligned} \boldsymbol{h}_r^{t+1} &= \boldsymbol{A}_I^T \boldsymbol{m}_r^t - \boldsymbol{q}_r^t \boldsymbol{\xi}_t^T,\ \boldsymbol{m}_r^t = \boldsymbol{g}_t \left(\boldsymbol{b}_r^t, \boldsymbol{w}_r\right), \\ \boldsymbol{b}_r^t &= \boldsymbol{A}_I \boldsymbol{q}_r^t - \boldsymbol{m}_r^{t-1} \boldsymbol{\lambda}_t^T,\ \boldsymbol{q}_r^t = \boldsymbol{f}_t \left(\boldsymbol{h}_r^t, \boldsymbol{x}_r\right), \end{aligned} \tag{3.4.8}$$

where $\boldsymbol{q}_r \in \mathbb{R}^{n\times K}$ is treated as an n-dimension vector with each super component as a K-dimension row vector. The definitions for other notations with subscript r are the same. The new form is exactly the same as the one proposed in [37, Proposition 5]. The correctness of state evolution of this form has already been analysed in [37].

3.5 Case Study and Simulations

3.5.1 Bernoulli-Gaussian Prior

In order to practically justify the correctness of our proposed algorithm, we take the well known BG distribution $p_{x_{:,i}}(\boldsymbol{x}) = (1-\epsilon)\,\delta^f_{\boldsymbol{x}=\mathbf{0}} + \epsilon p(\boldsymbol{x};\, \mathbf{0},\, \boldsymbol{\Sigma}_x)$ as an example, where $\epsilon \in (0,\, 1]$ stands for the sparsity level, $\delta^f_{\boldsymbol{x}=\mathbf{0}}$ is the Dirac delta function, $p(\boldsymbol{x};\, \mathbf{0},\, \boldsymbol{\Sigma}_x)$ represents the Gaussian density with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}_x \in \mathbb{R}^{K\times K}$. For each super component, we have corresponding $\boldsymbol{\eta}(\cdot)$(ignoring the superscript t)

$$\begin{aligned}\boldsymbol{\eta}(\tilde{\boldsymbol{x}}_{:,i}) &= \int \boldsymbol{x}_{:,i} p(\boldsymbol{x}_{:,i}|\tilde{\boldsymbol{x}}_{:,i})\, d\boldsymbol{x}_{:,i} \\ &= \frac{\boldsymbol{R}\tilde{\boldsymbol{x}}_{:,i}}{C_1 \exp\left(-\frac{1}{2}\tilde{\boldsymbol{x}}^T_{:,i}\boldsymbol{\Sigma}_e^{-1}\boldsymbol{R}\tilde{\boldsymbol{x}}_{:,i}\right)+1},\end{aligned} \tag{3.5.1}$$

where $\boldsymbol{R} = \left(\boldsymbol{\Sigma}_x^{-1} + \boldsymbol{\Sigma}_e^{-1}\right)^{-1}\boldsymbol{\Sigma}_e^{-1}$ and $C_1 = \frac{(1-\epsilon)}{\epsilon}\left|\boldsymbol{\Sigma}_x\left(\boldsymbol{\Sigma}_x^{-1} + \boldsymbol{\Sigma}_e^{-1}\right)\right|^{\frac{1}{2}}$ (see section 3.6.8 for the detailed proof). The direct computation of $\boldsymbol{\Sigma}_\eta$ based on (3.3.6) will be

$$\Sigma_\eta(p,q) = \epsilon\Sigma_x(p,q) - \Sigma_z(p,q) \tag{3.5.2}$$

where $\Sigma_z(p,q) = \mathrm{E}_{\tilde{x}_{:,i}}\left[\mathrm{E}_{x|\tilde{x}_{:,i}}[x_p|\tilde{\boldsymbol{x}}_{:,i}]\,\mathrm{E}_{x|\tilde{x}_{:,i}}[x_q|\tilde{\boldsymbol{x}}_{:,i}]\right]$ and $\Sigma_z(p,q)$ represents the element at location (p,q) of matrix $\boldsymbol{\Sigma}_z$ and

$$\boldsymbol{\Sigma}_z(p,q) = \int \frac{[\boldsymbol{R}\tilde{\boldsymbol{x}}_{:,i}]_p\,[\boldsymbol{R}\tilde{\boldsymbol{x}}_{:,i}]_q\,\epsilon p_{\mathrm{G}}(\tilde{\boldsymbol{x}}_{:,i};\,\mathbf{0},\,\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e)}{C_1 exp\left(-\frac{1}{2}\tilde{\boldsymbol{x}}_{:,i}^T\boldsymbol{\Sigma}_e^{-1}\boldsymbol{R}\tilde{\boldsymbol{x}}_{:,i}\right)+1}\mathrm{d}\tilde{\boldsymbol{x}}_{:,i}, \tag{3.5.3}$$

where $[\boldsymbol{R}\tilde{\boldsymbol{x}}_{:,i}]_p$ represents the single element of vector $\boldsymbol{R}\tilde{\boldsymbol{x}}_{:,i}$ at position p (proof is given in section 3.6.9). The computation of (3.5.3) will be difficult or even not achievable in practice as it contains high-dimensional integration (with respect to $\tilde{\boldsymbol{x}}_{:,i}$). An alternative calculation of the covariance matrix is based on the following lemma.

Lemma 3.5.1. *Consider a random vector $\boldsymbol{X}\in\mathbb{R}^K$ with a conditional probability density function of the form $p_{X|V}(\boldsymbol{x}|\boldsymbol{v}) \coloneqq \frac{1}{Z(\boldsymbol{v})}exp\left(\phi_0(\boldsymbol{x})+\boldsymbol{x}^T\boldsymbol{\Sigma}_e^{-1}\boldsymbol{v}\right)$ where $Z(\boldsymbol{v})$ is a normalization constant, then we have covariance matrix of $\boldsymbol{X}$ conditioned on $\boldsymbol{v}$ is given by*

$$\boldsymbol{\Sigma}_{X|V=v} = \boldsymbol{D}\left(\mathrm{E}_{X|V}[\boldsymbol{X}|\boldsymbol{V}=\boldsymbol{v}]\right)\boldsymbol{\Sigma}_e, \tag{3.5.4}$$

where $\boldsymbol{D}\left(\mathrm{E}_{X|V}[\boldsymbol{X}|\boldsymbol{V}=\boldsymbol{v}]\right) = \partial\mathrm{E}_{X|V}[\boldsymbol{X}|\boldsymbol{V}=\boldsymbol{v}]/\partial\boldsymbol{v}\in\mathbb{R}^{K\times K}$ with the (l,j)-th component calculated by $\partial\mathrm{E}_{X|V}[\boldsymbol{X}|\boldsymbol{V}=\boldsymbol{v}]_l/\partial\boldsymbol{v}_j$.

Proof. See section 3.6.6. □

Combine Lemma 3.5.1 with the Gaussian part of BG distribution (both mean vector and covariance matrix of non-Gaussian part are zero), we have $\boldsymbol{x}=\boldsymbol{x}_{:,i}$, $\boldsymbol{v}=\tilde{\boldsymbol{x}}_{:,i}$ and $\mathrm{E}_{x_{:,i}|\tilde{x}_{:,i}}[\boldsymbol{x}_{:,i}|\tilde{\boldsymbol{x}}_{:,i}] = \boldsymbol{\eta}(\tilde{\boldsymbol{x}}_{:,i})$. The corresponding covariance matrix of $\boldsymbol{x}_{:,i}$ given $\tilde{\boldsymbol{x}}_{:,i}$, can be calculated by $\boldsymbol{\eta}'(\tilde{\boldsymbol{x}}_{:,i})\,\boldsymbol{\Sigma}_e$ and the average

covariance matrix for the whole system, can be calculated by

$$\boldsymbol{\Psi}\left(\boldsymbol{\Sigma}_e\right) \;=\; \boldsymbol{D}\boldsymbol{\Sigma}_e, \tag{3.5.5}$$

where $\boldsymbol{D} = \mathrm{E}_{\tilde{x}_{:,i}}\left[\boldsymbol{\eta}'\left(\tilde{\boldsymbol{x}}_{:,i}\right)\right]$ and for the large system limit, $\boldsymbol{D} = \frac{1}{n}\sum_i^n \boldsymbol{\eta}'\left(\tilde{\boldsymbol{x}}_{:,i}\right)$. The (l,j)-th component of $\boldsymbol{\eta}'\left(\tilde{\boldsymbol{x}}_{:,i}\right)$ is calculated by

$$\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}_{:,i}\right)'_{l,j} = \frac{R_{l,j}}{C_2+1} + \frac{\left[\boldsymbol{R}\tilde{\boldsymbol{x}}_{:,i}\right]_l}{\left(C_2+1\right)^2} C_2 \left[\boldsymbol{\Sigma}_e^{-1}\boldsymbol{R}\tilde{\boldsymbol{x}}_{:,i}\right]_j, \tag{3.5.6}$$

where $C_2 = C_1 \exp\left(-\frac{1}{2}\tilde{\boldsymbol{x}}_{:,i}^T\boldsymbol{\Sigma}_e^{-1}\boldsymbol{R}\tilde{\boldsymbol{x}}_{:,i}\right)$ and $R_{l,j}$ is the (l,j)-th component of $\boldsymbol{R}$. The pseudo code of AMP-C-DCS is give in Algorithm 3.1 and the corresponding state evolution is given in Algorithm 3.2.

3.5.2 Gaussian Prior

Now consider Gaussian signals for the DCS and MMV cases. Then MMSE estimator (3.5.1) and the associated covariance matrix of estimation error have nice closed forms:

$$\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}_{:,i}\right) \;=\; \mathrm{E}_{x_{:,i}|\tilde{x}_{:,i}}\left[\boldsymbol{x}_{:,i}|\tilde{\boldsymbol{x}}_{:,i}\right] = \boldsymbol{R}\tilde{\boldsymbol{x}}_{:,i}, \tag{3.5.7}$$

$$\boldsymbol{\Psi}\left(\boldsymbol{\Sigma}_e\right) \;=\; \left(\boldsymbol{\Sigma}_x^{-1} + \boldsymbol{\Sigma}_e^{-1}\right)^{-1}. \tag{3.5.8}$$

which can be achieved via the standard result of an MMSE estimator for a Gaussian prior signal or derived form BG analysis by setting $\epsilon = 1$ (see detailed proof in section 3.6.10). When the signal covariance matrix $\boldsymbol{\Sigma}_x$ are of special forms, i.e., either $\boldsymbol{\Sigma}_x = \sigma_x^2\boldsymbol{I}$ or $\boldsymbol{\Sigma}_x = \sigma_x^2\mathbf{1}$ where all the entries of the matrix $\mathbf{1} \in \mathbb{R}^{K\times K}$ are one, the state evolution admits simple closed forms.

Algorithm 3.1 Pseudo code of AMP-C-DCS algorithm.

Input: Measurement (block) matrix $\boldsymbol{A}$, observation (block) vector $\boldsymbol{y}$, covariance matrices $\boldsymbol{\Sigma}_A$, $\boldsymbol{\Sigma}_x$, σ_w^2, sparsity level ϵ.

Output: Estimated signal $\boldsymbol{x}^t$.

Initialization: $\boldsymbol{r}^0 = \boldsymbol{y}$, $\boldsymbol{x}^0 = \boldsymbol{0}$, $\boldsymbol{\Sigma}_\eta^0 = \epsilon\boldsymbol{\Sigma}_x$, $t = 0$.

Iteration: In the t-th iteration, do

1. Compute
$$\tilde{\boldsymbol{x}}^t = \boldsymbol{A}^T\boldsymbol{r}^t + \boldsymbol{x}^t,$$
2. Compute covariance matrix of equivalent noise
$$\boldsymbol{\Sigma}_e^t = \frac{n}{m}\left(\boldsymbol{\Sigma}_A\right).^2 \odot \boldsymbol{\Sigma}_\eta^t + \sigma_w^2\boldsymbol{I},$$
3. Estimate signal
$$\boldsymbol{x}_{:,i}^{t+1} = \boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}_{:,i}^t\right), \forall i \in [n]$$
and calculate $\boldsymbol{D}^t$,
4. Update covariance matrix by fast calculation
$$\boldsymbol{\Sigma}_\eta^{t+1} = \boldsymbol{D}^t\boldsymbol{\Sigma}_e^t,$$
5. Update residual
$$\boldsymbol{r}^{t+1} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{t+1} + \frac{1}{\delta}\left(\boldsymbol{I}_m \otimes \left(\boldsymbol{\Sigma}_A \odot \boldsymbol{D}^t\right)\right)\boldsymbol{r}^t,$$
6. $t = t + 1$,
7. Go back to step 1 unless the stopping criteria are satisfied.

Algorithm 3.2 State evolution of AMP-C-DCS algorithm.

Input: Covariance matrices $\mathbf{\Sigma}_A$, $\mathbf{\Sigma}_x$, σ_w^2, sparsity level ϵ.

Output: MSE of estimation.

Initialization: $\mathbf{\Sigma}_\eta^0 = \epsilon\mathbf{\Sigma}_x$, $t = 0$.

Iteration: In the t-th iteration, do

1. Compute covariance matrix of equivalent noise

$$\mathbf{\Sigma}_e^t = \frac{n}{m}\left(\mathbf{\Sigma}_A\right).^2 \odot \mathbf{\Sigma}_\eta^t + \sigma_w^2\boldsymbol{I},$$

2. Update theoretical value of covariance matrix

$$\mathbf{\Sigma}_\eta^{t+1} = \mathbf{\Psi}\left(\mathbf{\Sigma}_e^t\right),$$

3. $t = t + 1$,

4. Go back to step 1 unless the stopping criteria are satisfied.

Define the average recovery distortion at the steady state by $d^\infty = \frac{1}{Kn}\left\|\boldsymbol{x} - \boldsymbol{x}^\infty\right\|_2^2$. It can be verified, via steady state analysis, that when $\mathbf{\Sigma}_x = \sigma_x^2\boldsymbol{I}$ (independent signals),

$$d_{MMV}^\infty = d_{DS}^\infty = \frac{\delta}{2}\left(\left(\frac{1-\delta}{\delta}\sigma_x^2 - \sigma_w^2\right) + \sqrt{\left(\frac{1-\delta}{\delta}\sigma_x^2 + \sigma_w^2\right)^2 + 4\sigma_x^2\sigma_w^2}\right), \quad (3.5.9)$$

and when $\mathbf{\Sigma}_x = \sigma_x^2\mathbf{1}$ (repeated signals: $\boldsymbol{x}_1 = \cdots = \boldsymbol{x}_K$),

$$d_{MMV}^\infty = \frac{\delta}{2}\left(\left(\frac{1-\delta}{\delta}\sigma_x^2 - \frac{\sigma_w^2}{K}\right) + \sqrt{\left(\frac{1-\delta}{\delta}\sigma_x^2 + \frac{\sigma_w^2}{K}\right)^2 + \frac{4\sigma_w^2\sigma_x^2}{K}}\right), \quad (3.5.10)$$

$$d_{DS}^\infty = \frac{\delta}{2}\left(\left(\frac{1-K\delta}{\delta}\sigma_x^2 - \sigma_w^2\right) + \sqrt{\left(\frac{1-K\delta}{\delta}\sigma_x^2 + \sigma_w^2\right)^2 + 4K\sigma_x^2\sigma_w^2}\right). \quad (3.5.11)$$

(see detailed proof in section 3.6.11.)

It is interesting to observe that the same results can be obtained by solely applying random matrix theory. Specifically, consider a linear system $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}+\boldsymbol{w}$ where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is Gaussian random matrix with i.i.d. entries drawn from $\mathcal{N}\left(0, \frac{1}{m}\right)$, $\boldsymbol{x} \in \mathbb{R}^n$ is the signal drawn from $\mathcal{N}(\boldsymbol{0}, \sigma_x^2\boldsymbol{I})$, and $\boldsymbol{w} \in \mathbb{R}^m$ is the noise drawn from $\mathcal{N}(\boldsymbol{0}, \sigma_w^2\boldsymbol{I})$. Let $(m, n) \to \infty$ simultaneously with $\frac{m}{n} \to \delta$. The empirical distribution of the eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$ converges to the Marchenko-Pastur distribution weakly. Based on this fact, the average distortion of MMSE estimate, i.e., $\lim_{n\to\infty} \frac{1}{n} \|\boldsymbol{x} - \hat{\boldsymbol{x}}_{MMSE}\|_2^2$, can be computed as [81, 82, 29]

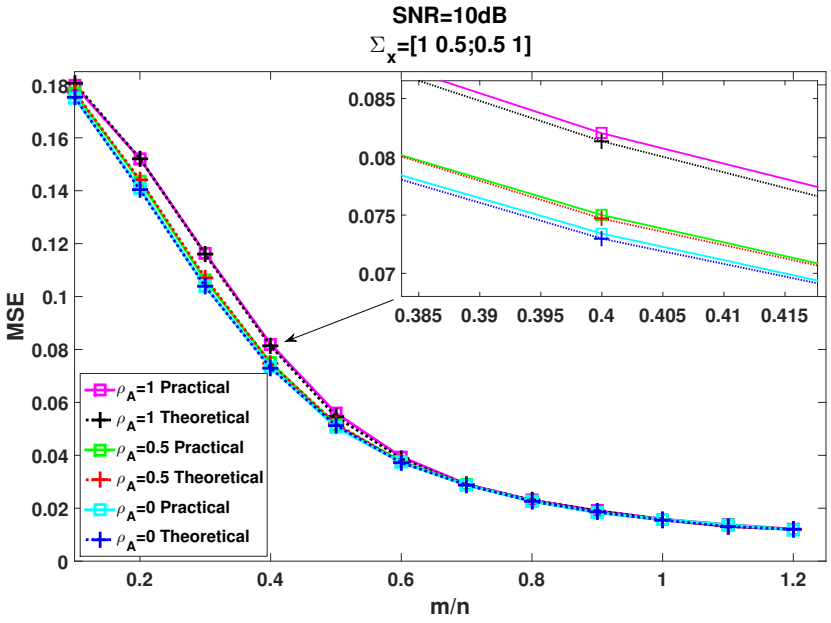
$$f\left(\delta, \sigma_x^2, \sigma_w^2\right) = \frac{1}{n}\mathrm{tr}\left(\boldsymbol{\Sigma}_{x|y}\right) = \frac{1}{n}\mathrm{tr}\left(\left(\sigma_x^{-2}\boldsymbol{I} + \sigma_w^{-2}\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\right)$$
$$= \frac{\delta}{2}\left(\left(\frac{1-\delta}{\delta}\sigma_x^2 - \sigma_w^2\right) + \sqrt{\left(\frac{1-\delta}{\delta}\sigma_x^2 + \sigma_w^2\right)^2 + 4\sigma_x^2\sigma_w^2}\right), \tag{3.5.12}$$

where $\mathrm{tr}(\boldsymbol{M})$ calculates the trace of a square matrix $\boldsymbol{M}$.

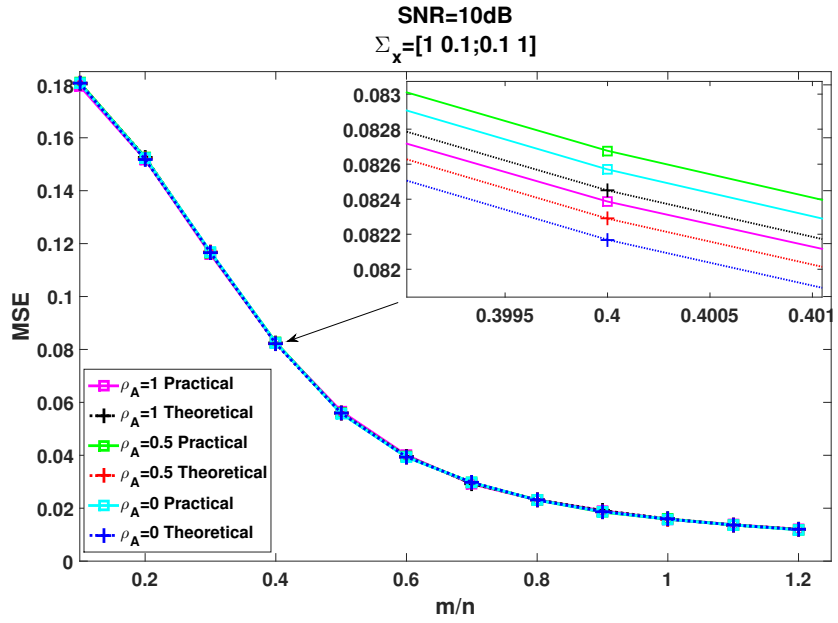
(A recent paper [83] proved that random matrix theory can also be used to calculate the MMSE for non-Gaussian signal priors. The performance comparison between random matrix theory and SE technique is not included here but should be an interesting future research topic)

3.5.3 Numerical Results

In Fig. 3.5.1, we focus on the case $K = 2$. Consider the block signal and matrix models specified in Definition 3.2.1 and 3.2.2 respectively. Assume $\boldsymbol{\Sigma}_A = [1\ \rho_A;\ \rho_A\ 1]$ and $\boldsymbol{\Sigma}_x = [1\ \rho_x;\ \rho_x\ 1]$. In the simulations, we numerically study the performance of AMP-C-DCS as a function of ρ_A when $\rho_x = 0.5$ (in Fig. 3.5.1a) and $\rho_x = 0.1$ (in Fig. 3.5.1b). The signal sparsity level is set to $\epsilon =$

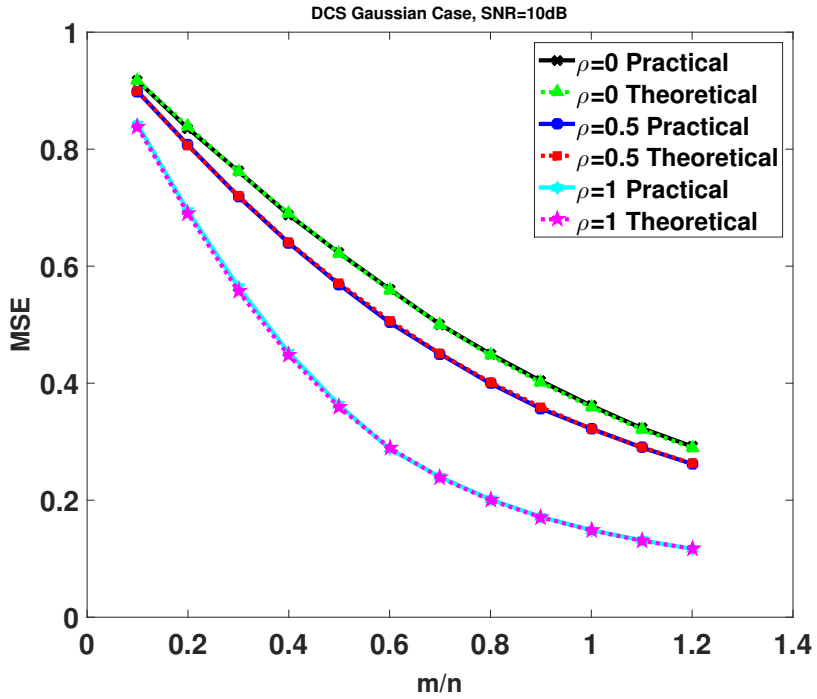


(a) Signal correlated coefficient :$\rho_x = 0.5$

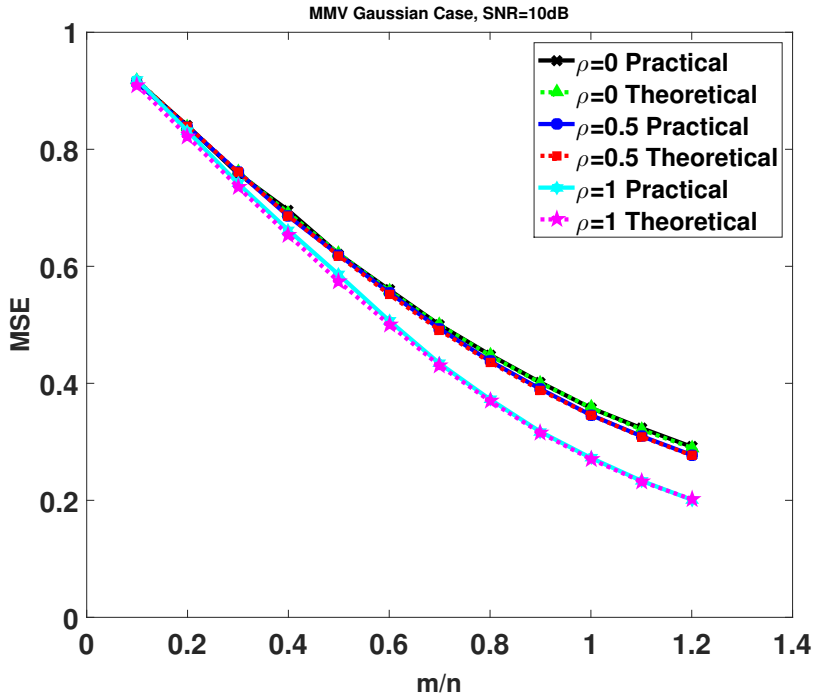


(b) Signal correlated coefficient : $\rho_x = 0.1$

Figure 3.5.1: Simulation results. For AMP-C-DCS with BG prior where $K = 2$, the off diagonal elements of $\mathbf{\Sigma}_x$ are denoted by ρ_x which controls the correlation between signals, the off diagonal element of $\mathbf{\Sigma}_A$ is denoted by ρ_A which controls the correlation between matrices.



(a) DCS model with Gaussian prior: $\rho_A = 0$



(b) MMV model with Gaussian prior: $\rho_A = 1$

Figure 3.5.2: Simulation results. For AMP-C-DCS with Gaussian prior where $K = 2$, the off diagonal elements of $\mathbf{\Sigma}_x$ are denoted by ρ which controls the correlation between signals.

0.2 and the signal dimension is given by $n = 3000$. The simulated curves are obtained from the average of 250 trials. We add the theoretical curve achieved by state evolution (Algorithm 3.2) to judge the performance of AMP-C-DCS. The difference between them is less than 1% and noticeable only when the curves are zoomed in. Furthermore, from Fig. 3.5.1a, where signal components are more correlated, it can be observed that AMP-C-DCS performs better when the measurement matrices become more independent. Such gains become tiny in Fig. 3.5.1b when the signal components are nearly independent. This is consistent with the results in Theorem 3.3.1 which suggests the gain of AMP-C-DCS comes from the independence of the measurement matrices and the correlation of the signals. The special cases of DCS and MMV systems with Gaussian prior ($\epsilon = 1$) are illustrated in Fig 3.5.2. For empirical study, the dimension of the signal $\boldsymbol{x}_k$ is set by $n = 1000$. The numerical results are obtained from the average of 100 trials. Similar comments can be made with a more significant performance gain. Although the correctness of the proposed algorithm for the general case where $0 < \rho_A < 1$ has not be rigorously analysed, simulation results validate the accuracy of SE .

The above observed phenomenon can be explained by the following example: assume in the noise free case, we take several photos of an object (identical signals) with a same camera (identical measurement matrices). No additional information will be achieved compared with a single snapshot of the object. If we take photos with different kinds of cameras, for example, one is an optical camera which produces a regular colour image of the object and another is the so called depth camera which provides a depth image of the object (less correlated measurement matrices). Each pixel of the depth image represents the distance between a target point and the camera. With the information pro-

vided by the colour image and the depth image, we may create the 3D model of the object as more information is provided with less correlated measurement matrices.

3.6 Proof

3.6.1 AMP-C-DCS Algorithm: A Heuristic Derivation

In this section we provide the heuristic derivation of our proposed AMP-C-DCS algorithm. The steps are mainly based on the heuristic derivation of the original AMP algorithm proposed in [60, Section 5.2] and based on B − model. To simplify representation, we omit the subscript B in the following analysis. We start from the following message passing algorithm, at t-th iteration,

$$\boldsymbol{r}_{a\to i}^{t} = \boldsymbol{y}_a - \sum_{j\in[n]\backslash i} \boldsymbol{A}_{aj}\boldsymbol{x}_{j\to a}^{t}, \tag{3.6.1}$$

$$\boldsymbol{x}_{i\to a}^{t+1} = \boldsymbol{\eta}_t\left(\sum_{b\in[m]\backslash a} \boldsymbol{A}_{bi}\boldsymbol{r}_{b\to i}^{t}\right), \tag{3.6.2}$$

where subscript $a \to i$ in (3.6.1) denotes the message passing from factor node $a \in [m]$ to variable node $i \in [n]$, subscript $i \to a$ in (3.6.2) denotes the message passing from variable node i to factor node a and $[n]\backslash i$ represents the set $[n]$ without element i. Recall, in the C-DCS model, the elements (e.g. $\boldsymbol{y}_a \in \mathbb{R}^K$, $\boldsymbol{A}_{aj} \in \mathbb{R}^{K\times K}$, $\boldsymbol{x}_{j\to a}^{t} \in \mathbb{R}^K$, $\boldsymbol{r}_{a\to i}^{t} \in \mathbb{R}^K$) are no longer scalars, instead, they are super components. A natured guess is that $\boldsymbol{r}_{a\to i}^{t} = \boldsymbol{r}_a^t + \overrightarrow{O}\left(n^{-1/2}\right)$ and $\boldsymbol{x}_{i\to a}^{t} = \boldsymbol{x}_i^t + \overrightarrow{O}\left(m^{-1/2}\right)$, then setting

$$\boldsymbol{r}_{a\to i}^{t} = \boldsymbol{r}_a^t + \triangle\boldsymbol{r}_{a\to i}^{t},\ \boldsymbol{x}_{i\to a}^{t} = \boldsymbol{x}_i^t + \triangle\boldsymbol{x}_{i\to a}^{t}. \tag{3.6.3}$$

Substituting in (3.6.1), (3.6.2) and ignore the terms $\boldsymbol{A}_{ai}\triangle\boldsymbol{x}_{i\to a}^{t}$ and $\boldsymbol{A}_{ai}\triangle\boldsymbol{r}_{a\to i}^{t}$ which are of order $1/n$ (the magnitude of each element inside the super component), provides

$$\begin{aligned}\boldsymbol{r}_{a}^{t}+\triangle\boldsymbol{r}_{a\to i}^{t}&=\boldsymbol{y}_{a}-\sum_{j\in[n]}\boldsymbol{A}_{aj}\left(\boldsymbol{x}_{j}^{t}+\triangle\boldsymbol{x}_{j\to a}^{t}\right)+\boldsymbol{A}_{ai}\boldsymbol{x}_{i}^{t},\\ \boldsymbol{x}_{i}^{t+1}+\triangle\boldsymbol{x}_{i\to a}^{t+1}&=\boldsymbol{\eta}_{t}\left(\sum_{b\in[m]}\boldsymbol{A}_{bi}\left(\boldsymbol{r}_{b}^{t}+\triangle\boldsymbol{r}_{b\to i}^{t}\right)-\boldsymbol{A}_{ai}\boldsymbol{r}_{a}^{t}\right).\end{aligned}\tag{3.6.4}$$

By applying first order Taylor's expansion to the $\boldsymbol{\eta}_t$ function, we have the following approximation

$$\begin{aligned}\boldsymbol{x}_{i}^{t+1}+\triangle\boldsymbol{x}_{i\to a}^{t+1}=\boldsymbol{\eta}_{t}&\left(\sum_{b\in[m]}\boldsymbol{A}_{bi}\left(\boldsymbol{r}_{b}^{t}+\triangle\boldsymbol{r}_{b\to i}^{t}\right)\right)\\&-\boldsymbol{\eta}_{t}'\left(\sum_{b\in[m]}\boldsymbol{A}_{bi}\left(\boldsymbol{r}_{b}^{t}+\triangle\boldsymbol{r}_{b\to i}^{t}\right)\right)\boldsymbol{A}_{ai}\boldsymbol{r}_{a}^{t}\end{aligned}\tag{3.6.5}$$

where $\boldsymbol{\eta}_t'$ is the Jacobian matrix of $\boldsymbol{\eta}_t$. Now compare (3.6.3) with (3.6.4) and (3.6.5), a reasonable decomposition will be

$$\boldsymbol{r}_{a}^{t}=\boldsymbol{y}_{a}-\sum_{j\in[n]}\boldsymbol{A}_{aj}\left(\boldsymbol{x}_{j}^{t}+\triangle\boldsymbol{x}_{j\to a}^{t}\right),\tag{3.6.6}$$

$$\triangle\boldsymbol{r}_{a\to i}^{t}=\boldsymbol{A}_{ai}\boldsymbol{x}_{i}^{t},\tag{3.6.7}$$

$$\boldsymbol{x}_{i}^{t+1}=\boldsymbol{\eta}_{t}\left(\sum_{b\in[m]}\boldsymbol{A}_{bi}\left(\boldsymbol{r}_{b}^{t}+\triangle\boldsymbol{r}_{b\to i}^{t}\right)\right)\tag{3.6.8}$$

$$\triangle\boldsymbol{x}_{i\to a}^{t+1}=-\boldsymbol{\eta}_{t}'\left(\sum_{b\in[m]}\boldsymbol{A}_{bi}\left(\boldsymbol{r}_{b}^{t}+\triangle\boldsymbol{r}_{b\to i}^{t}\right)\right)\boldsymbol{A}_{ai}\boldsymbol{r}_{a}^{t}\tag{3.6.9}$$

Substituting (3.6.7) in (3.6.8) and $\sum_{b\in[m]} \boldsymbol{A}_{bi}^2 \to \boldsymbol{I}$ from the definition of the block matrix model in Definition 3.2.2, we have

$$\boldsymbol{x}^{t+1} = \boldsymbol{\eta}_t \left(\boldsymbol{A}^T \boldsymbol{r}^t + \boldsymbol{x}^t \right)$$

which is the first equation of our proposed algorithm.

Substituting (3.6.9) in (3.6.6) yields

$$\begin{aligned}
\boldsymbol{r}_a^t &= \boldsymbol{y}_a - \sum_{j\in[n]} \boldsymbol{A}_{aj} \boldsymbol{x}_j^t - \sum_{j\in[n]} \boldsymbol{A}_{aj} \triangle \boldsymbol{x}_{j\to a}^t \\
&= \boldsymbol{y}_a - \sum_{j\in[n]} \boldsymbol{A}_{aj} \boldsymbol{x}_j^t \\
&+ \sum_{j\in[n]} \boldsymbol{A}_{aj} \boldsymbol{\eta}_t' \left(\sum_{b\in[m]} \boldsymbol{A}_{bj} \boldsymbol{r}_b^{t-1} + \boldsymbol{x}_j^{t-1} \right) \boldsymbol{A}_{aj} \boldsymbol{r}_a^{t-1}
\end{aligned}$$

Define $\boldsymbol{D}_j^{t-1} \coloneqq \boldsymbol{\eta}_{t-1}' \left(\sum_{b\in[m]} \boldsymbol{A}_{bj} \boldsymbol{r}_b^{t-1} + \boldsymbol{x}_j^{t-1} \right)$ and $\boldsymbol{D}^{t-1} = \frac{1}{n} \sum_{j\in[n]} \boldsymbol{D}_j^{t-1}$, we have

$$\begin{aligned}
\boldsymbol{r}_a^t &= \boldsymbol{y}_a - \sum_{j\in[n]} \boldsymbol{A}_{aj} \boldsymbol{x}_j^t + \sum_{j\in[n]} \boldsymbol{A}_{aj} \boldsymbol{D}_j^{t-1} \boldsymbol{A}_{aj} \boldsymbol{r}_a^{t-1} \\
&\approx \boldsymbol{y}_a - \sum_{j\in[n]} \boldsymbol{A}_{aj} \boldsymbol{x}_j^t + nE \left[\boldsymbol{A}_{aj} \boldsymbol{D}_j^{t-1} \boldsymbol{A}_{aj} \right] \boldsymbol{r}_a^{t-1} \\
&= \boldsymbol{y}_a - \sum_{j\in[n]} \boldsymbol{A}_{aj} \boldsymbol{x}_j^t + \frac{1}{\delta} \left(\boldsymbol{\Sigma}_A \odot \boldsymbol{D}^{t-1} \right) \boldsymbol{r}_a^{t-1}
\end{aligned}$$

which provides the second equation of our proposed algorithm. This finishes our derivation.

3.6.2 Independent Case Where $\boldsymbol{\Sigma}_A = \boldsymbol{I}_K$

Apply D − model and we have $\boldsymbol{\Sigma}_m = \boldsymbol{I}_{Km}$ and $\boldsymbol{\Sigma}_n = \boldsymbol{I}_{Kn}$. The first term in (3.4.4) becomes

$$\begin{aligned}
&\tau_2^{-1}\left(\tau_2\left(\boldsymbol{Y}\right)\left(\boldsymbol{Q}^T\boldsymbol{\Sigma}_n\boldsymbol{Q}\right)^{-1}\boldsymbol{Q}^T\boldsymbol{\Sigma}_n\right)\\
=&\tau_2^{-1}\left(\tau_2\left(\boldsymbol{Y}\right)\left(\boldsymbol{Q}^T\boldsymbol{Q}\right)^{-1}\boldsymbol{Q}^T\right)\\
=&\boldsymbol{Y}\left(\boldsymbol{Q}^T\boldsymbol{Q}\right)^{-1}\boldsymbol{Q}^T.
\end{aligned}$$

Similarly, the second term in (3.4.4) becomes

$$\begin{aligned}
&\tau_1^{-1}\left(\boldsymbol{\Sigma}_m\boldsymbol{M}\left(\boldsymbol{M}^T\boldsymbol{\Sigma}_m\boldsymbol{M}\right)^{-1}\tau_1\left(\boldsymbol{X}^T\right)\right)\\
=&\boldsymbol{M}\left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\boldsymbol{X}^T,
\end{aligned}$$

then the third term in (3.4.4) is

$$\begin{aligned}
&\tau_1^{-1}\left(\boldsymbol{M}\left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\tau_1\left(\boldsymbol{M}^T\tau_2^{-1}\left(\tau_2\left(\boldsymbol{Y}\right)\left(\boldsymbol{Q}^T\boldsymbol{Q}\right)^{-1}\boldsymbol{Q}^T\right)\right)\right)\\
=&\tau_1^{-1}\left(\boldsymbol{M}\left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\tau_1\left(\boldsymbol{M}^T\boldsymbol{Y}\left(\boldsymbol{Q}^T\boldsymbol{Q}\right)^{-1}\boldsymbol{Q}^T\right)\right)\\
=&\boldsymbol{M}\left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\boldsymbol{M}^T\boldsymbol{Y}\left(\boldsymbol{Q}^T\boldsymbol{Q}\right)^{-1}\boldsymbol{Q}^T.
\end{aligned}$$

For the projection (3.4.5), first calculate the left projection

$$\begin{aligned}
&\tau_1^{-1}\left(\boldsymbol{P}_M^{\natural}\tau_1\left(\tilde{\boldsymbol{A}}\right)\right)\\
=&\tau_1^{-1}\left(\left(\boldsymbol{I}-\boldsymbol{M}\left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\boldsymbol{M}^T\right)\tau_1\left(\tilde{\boldsymbol{A}}\right)\right)\\
=&\left(\boldsymbol{I}-\boldsymbol{M}\left(\boldsymbol{M}^T\boldsymbol{M}\right)^{-1}\boldsymbol{M}^T\right)\left(\tilde{\boldsymbol{A}}\right)\\
:=&\boldsymbol{P}_M^{\perp}\left(\tilde{\boldsymbol{A}}\right),
\end{aligned}$$

then calculate the right projection

$$\begin{aligned}
&\tau_2^{-1}\left(\tau_2\left(\boldsymbol{P}_M^{\perp}\left(\tilde{\boldsymbol{A}}\right)\right)\boldsymbol{P}_Q^{\nparallel T}\right)\\
=&\tau_2^{-1}\left(\tau_2\left(\boldsymbol{P}_M^{\perp}\left(\tilde{\boldsymbol{A}}\right)\right)\left(\boldsymbol{I}-\boldsymbol{Q}\left(\boldsymbol{Q}^T\boldsymbol{Q}\right)^{-1}\boldsymbol{Q}^T\right)\right)\\
:=&\boldsymbol{P}_M^{\perp}\left(\tilde{\boldsymbol{A}}\right)\boldsymbol{P}_Q^{\perp}.
\end{aligned}$$

The proof finished for the independent case.

3.6.3 Identical Case Where $\boldsymbol{\Sigma}_A = 1_K$

Apply D − model and we have $\boldsymbol{\Sigma}_m = 1_{\mathrm{K}} \otimes \boldsymbol{I}_m$ and $\boldsymbol{\Sigma}_n = 1_{\mathrm{K}} \otimes \boldsymbol{I}_n$. Follow the same steps as in section 3.6.2. The first and second terms in (3.4.4) can be easily achieved as

$$\begin{aligned}
&\tau_2^{-1}\left(\tau_2\left(\boldsymbol{Y}\right)\left(\boldsymbol{Q}^T\boldsymbol{\Sigma}_n\boldsymbol{Q}\right)^{-1}\boldsymbol{Q}^T\boldsymbol{\Sigma}_n\right)\\
=&\boldsymbol{I}_K\otimes\left(\boldsymbol{Y}_R\left(\boldsymbol{Q}_R^T\boldsymbol{Q}_R\right)^{-1}\boldsymbol{Q}_R^T\right)\\
&\tau_1^{-1}\left(\boldsymbol{\Sigma}_m\boldsymbol{M}\left(\boldsymbol{M}^T\boldsymbol{\Sigma}_m\boldsymbol{M}\right)^{-1}\tau_1\left(\boldsymbol{X}^T\right)\right)\\
=&\boldsymbol{I}_K\otimes\left(\boldsymbol{M}_R\left(\boldsymbol{M}_R^T\boldsymbol{M}_R\right)^{-1}\boldsymbol{X}_R^T\right),
\end{aligned}$$

and the third term is

$$\begin{aligned}
&\tau_1^{-1}\Big(\boldsymbol{\Sigma}_m\boldsymbol{M}\left(\boldsymbol{M}_R^T\boldsymbol{M}_R\right)^{-1}\tau_1\left(\boldsymbol{M}^T\Big[\boldsymbol{I}_K\otimes\Big(\boldsymbol{Y}_R\left(\boldsymbol{Q}_R^T\boldsymbol{Q}_R\right)^{-1}\boldsymbol{Q}_R^T\Big)\Big]\right)\Big)\\
=&\tau_1^{-1}\Big(\boldsymbol{\Sigma}_m\boldsymbol{M}\left(\boldsymbol{M}_R^T\boldsymbol{M}_R\right)^{-1}\tau_1\left(\boldsymbol{M}^T\right)\Big)\Big[\boldsymbol{I}_K\otimes\Big(\boldsymbol{Y}_R\left(\boldsymbol{Q}_R^T\boldsymbol{Q}_R\right)^{-1}\boldsymbol{Q}_R^T\Big)\Big]\\
=&\Big[\boldsymbol{I}_K\otimes\Big(\boldsymbol{M}_R\left(\boldsymbol{M}_R^T\boldsymbol{M}_R\right)^{-1}\boldsymbol{M}_R^T\Big)\Big]\Big[\boldsymbol{I}_K\otimes\Big(\boldsymbol{Y}_R\left(\boldsymbol{Q}_R^T\boldsymbol{Q}_R\right)^{-1}\boldsymbol{Q}_R^T\Big)\Big]\\
=&\,\boldsymbol{I}_K\otimes\Big(\boldsymbol{M}_R\left(\boldsymbol{M}_R^T\boldsymbol{M}_R\right)^{-1}\boldsymbol{M}_R^T\boldsymbol{Y}_R\left(\boldsymbol{Q}_R^T\boldsymbol{Q}_R\right)^{-1}\boldsymbol{Q}_R^T\Big).
\end{aligned}$$

For the projection (3.4.5), first calculate the left projection

$$
\begin{aligned}
&\tau_1^{-1}\left(\boldsymbol{P}_M^{\nparallel}\tau_1\left(\tilde{\boldsymbol{A}}\right)\right)\\
=&\tau_1^{-1}\left(\tau_1\left(\tilde{\boldsymbol{A}}\right)-\boldsymbol{\Sigma}_m\boldsymbol{M}\left(\boldsymbol{M}_R^T\boldsymbol{M}_R\right)^{-1}\boldsymbol{M}^T\tau_1\left(\tilde{\boldsymbol{A}}\right)\right)\\
=&\tau_1^{-1}\left(\mathbf{1}_{K\times 1}\otimes\tilde{\boldsymbol{A}}_I-\boldsymbol{\Sigma}_m\boldsymbol{M}\left(\boldsymbol{M}_R^T\boldsymbol{M}_R\right)^{-1}\tau_1\left(\boldsymbol{M}^T\tilde{\boldsymbol{A}}\right)\right)\\
=&\tau_1^{-1}\left(\mathbf{1}_{K\times 1}\otimes\tilde{\boldsymbol{A}}_I-\boldsymbol{\Sigma}_m\boldsymbol{M}\left(\boldsymbol{M}_R^T\boldsymbol{M}_R\right)^{-1}\boldsymbol{M}_R^T\tilde{\boldsymbol{A}}_I\right)\\
=&\boldsymbol{I}_K\otimes\left[\left(\boldsymbol{I}-\boldsymbol{M}_R\left(\boldsymbol{M}_R^T\boldsymbol{M}_R\right)^{-1}\boldsymbol{M}_R^T\right)\tilde{\boldsymbol{A}}_I\right]\\
=&\boldsymbol{I}_K\otimes\left(\boldsymbol{P}_{M_R}^{\perp}\tilde{\boldsymbol{A}}_I\right),
\end{aligned}
$$

where $\mathbf{1}_{K\times 1}\in\mathbb{R}^{K\times 1}$ with all one value. then calculate the right projection

$$
\begin{aligned}
&\tau_2^{-1}\left(\tau_2\left(\boldsymbol{I}_K\otimes\left(\boldsymbol{P}_{M_R}^{\perp}\tilde{\boldsymbol{A}}_I\right)\right)\boldsymbol{P}_Q^{\nparallel T}\right)\\
=&\,\tau_1^{-1}\left(\left(\boldsymbol{I}-\boldsymbol{\Sigma}_n\boldsymbol{Q}\left(\boldsymbol{Q}_R^T\boldsymbol{Q}_R\right)^{-1}\boldsymbol{Q}^T\right)\tau_1\left(\boldsymbol{I}_K\otimes\left(\boldsymbol{P}_{M_R}^{\perp}\tilde{\boldsymbol{A}}_I\right)^T\right)\right)^T\\
=&\,\boldsymbol{I}_K\otimes\left(\left(\boldsymbol{I}-\boldsymbol{Q}_R\left(\boldsymbol{Q}_R^T\boldsymbol{Q}_R\right)^{-1}\boldsymbol{Q}_R^T\right)\tilde{\boldsymbol{A}}_I^T\boldsymbol{P}_{M_R}^{\perp}\right)^T\\
=&\,\boldsymbol{I}_K\otimes\left(\boldsymbol{P}_{M_R}^{\perp}\tilde{\boldsymbol{A}}_I\boldsymbol{P}_{Q_R}^{\perp}\right).
\end{aligned}
$$

The proof finished for the identical case.

3.6.4 Gaussianity Analysis

The rigorous proof requires to check all the steps appeared in [29], but here we just provide an illustration to analyse the Gaussianity of $\boldsymbol{h}^{t+1}$ and $\boldsymbol{b}^t$ based on the conditional distribution $\boldsymbol{A}|_{\mathfrak{G}_{t1,t2}}$ and assume the induction hypothesis used in [29] where $K=1$ can be extended to our case where $K>1$ and $\boldsymbol{\Sigma}_A=\boldsymbol{I}$. We use $\stackrel{h}{=}$ to denote the requirements of certain induction hypothesis. Recall, according to Section 3.4.2, for the independent $\boldsymbol{A}_k$'s, we can consider

all the super components in (3.4.6) and $\boldsymbol{\xi}_t$, $\boldsymbol{\lambda}_t$ as diagonal matrices. The analysis in Corollary 3.4.6 is based on D − model, but we can easily rewrite it to B − model without changing the formula (Note: we can't do this operation in the identical case) and the following analysis in this section will mainly based on B − model, unless mentioned otherwise. Define the following

$$\begin{aligned}\boldsymbol{m}_{||}^t &= \sum_{i=0}^{t-1} \boldsymbol{\alpha}_i \oslash \boldsymbol{m}^i,\ \boldsymbol{q}_{||}^t = \sum_{i=0}^{t-1} \boldsymbol{\beta}_i \oslash \boldsymbol{q}^i \\ \overrightarrow{\boldsymbol{\alpha}} &= \overrightarrow{\boldsymbol{\alpha}}_t = [\boldsymbol{\alpha}_0, ..., \boldsymbol{\alpha}_{t-1}]^T \\ \overrightarrow{\boldsymbol{\beta}} &= \overrightarrow{\boldsymbol{\beta}}_t = [\boldsymbol{\beta}_0, ..., \boldsymbol{\beta}_{t-1}]^T\end{aligned}$$

where $\boldsymbol{\alpha}_i$, $\boldsymbol{\beta}_i \in \mathbb{R}^{K\times K}$ are diagonal matrices, $\oslash$ is the operation that multiple each super components of $\boldsymbol{m}^i$ and $\boldsymbol{q}^i$ with the same diagonal matrices $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$, i.e $\boldsymbol{\alpha}_i \oslash \boldsymbol{m}^i = (\boldsymbol{I} \otimes \boldsymbol{\alpha}_i)\, \boldsymbol{m}^i = \boldsymbol{m}^i \oslash \boldsymbol{\alpha}_i$. Let $\boldsymbol{v}, \boldsymbol{u} \in \mathbb{R}^{mK\times K}$ as Definition 3.2.1 but with diagonal super components, define the following inter product

$$\langle \boldsymbol{v}, \boldsymbol{u} \rangle \coloneqq \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{v}_i \boldsymbol{u}_i.$$

We are going to show:

$$\boldsymbol{h}^{t+1}|_{\mathfrak{G}_{t+1,t}} \overset{d}{=} \sum_{i=0}^{t-1} \boldsymbol{\alpha}_i \oslash \boldsymbol{h}^{i+1} + \tilde{\boldsymbol{A}}^T \boldsymbol{m}_{\perp}^t + \tilde{\boldsymbol{Q}}_{t+1} \vec{\boldsymbol{o}}_{t+1}\,(1) \tag{3.6.10}$$

$$\boldsymbol{b}^{t}|_{\mathfrak{G}_{t,t}} \overset{d}{=} \sum_{i=0}^{t-1} \boldsymbol{\beta}_i \oslash \boldsymbol{b}^{i} + \tilde{\boldsymbol{A}} \boldsymbol{q}_{\perp}^t + \tilde{\boldsymbol{M}}_t \vec{\boldsymbol{o}}_t\,(1) \tag{3.6.11}$$

where $\vec{\boldsymbol{o}}_t\,(1) \in \mathbb{R}^{tK\times K}$ is a t-dimensional vector contains diagonal super components with all the diagonal elements converges to 0 almost surely as $n \to \infty$.

Based on (3.2.9) and (3.4.6), we have

$$\boldsymbol{X}_t = \boldsymbol{H}_t + \boldsymbol{Q}_t \boldsymbol{\Xi}_t,\ \boldsymbol{Y}_t = \boldsymbol{B}_t + [\boldsymbol{0}|\boldsymbol{M}_{t-1}]\, \boldsymbol{\Lambda}_t \tag{3.6.12}$$

where $\boldsymbol{H}_t = [\boldsymbol{h}^1|...|\boldsymbol{h}^t]$, $\boldsymbol{\Xi}_t = \text{diag}\,(\boldsymbol{\xi}_0, ..., \boldsymbol{\xi}_{t-1})$, $\boldsymbol{B}_t = [\boldsymbol{b}^0|...|\boldsymbol{b}^{t-1}]$ and $\boldsymbol{\Lambda}_t = \text{diag}\,(\boldsymbol{\lambda}_0, ..., \boldsymbol{\lambda}_{t-1})$. As the conditional distribution $\boldsymbol{A}|_{\mathfrak{S}_{t1,t2}}$ has the same form as in [29], just changing the scalar elements to the block diagonal super components. Let $\boldsymbol{m}^t = \boldsymbol{m}^t_{||} + \boldsymbol{m}^t_{\perp}$ and $\boldsymbol{q}^t = \boldsymbol{q}^t_{||} + \boldsymbol{q}^t_{\perp}$, we can get the same results as [29, Lemma 12]:

$$\begin{aligned}\boldsymbol{E}^T_{t+1,t}\boldsymbol{m}^t &= \boldsymbol{X}_t \left(\boldsymbol{M}_t^T \boldsymbol{M}_t\right)^{-1} \boldsymbol{M}\boldsymbol{m}^t_{||} \\ &\quad + \boldsymbol{Q}_{t+1}\left(\boldsymbol{Q}_{t+1}^T\boldsymbol{Q}_{t+1}\right)^{-1}\boldsymbol{Y}_{t+1}^T\boldsymbol{m}^t_{\perp}, \qquad (3.6.13)\\ \boldsymbol{E}_{t,t}\boldsymbol{q}^t &= \boldsymbol{Y}_t\left(\boldsymbol{Q}_t^T\boldsymbol{Q}_t\right)^{-1}\boldsymbol{Q}\boldsymbol{q}^t_{||} \\ &\quad + \boldsymbol{M}_t\left(\boldsymbol{M}_t^T\boldsymbol{M}_t\right)^{-1}\boldsymbol{X}_t^T\boldsymbol{q}^t_{\perp}. \qquad (3.6.14)\end{aligned}$$

Now focus on $\boldsymbol{b}^t|_{\mathfrak{S}_{t,t}}$, with (3.6.14) and Corollary 3.4.6, we have

$$\begin{aligned}&\boldsymbol{b}^t|_{\mathfrak{S}_{t,t}} \\ &\overset{d}{=} \boldsymbol{A}\boldsymbol{q}^t|\mathfrak{S}_{t,t} - \boldsymbol{m}^{t-1} \oslash \boldsymbol{\lambda}_t \\ &= \boldsymbol{Y}_t\left(\boldsymbol{Q}_t^T\boldsymbol{Q}_t\right)^{-1}\boldsymbol{Q}_t^T\boldsymbol{q}^t_{||} + \boldsymbol{M}_t\left(\boldsymbol{M}_t^T\boldsymbol{M}_t\right)^{-1}\boldsymbol{X}_t^T\boldsymbol{q}^t_{\perp} \\ &+ \boldsymbol{P}^{\perp}_{M_t}\tilde{\boldsymbol{A}}\boldsymbol{P}^{\perp}_{Q_t}\boldsymbol{q}^t - \boldsymbol{m}^{t-1}\oslash\boldsymbol{\lambda}_t \\ &\overset{a}{=} \boldsymbol{B}_t\left(\boldsymbol{Q}_t^T\boldsymbol{Q}_t\right)^{-1}\boldsymbol{Q}_t^T\boldsymbol{q}^t_{||} + \boldsymbol{P}^{\perp}_{M_t}\tilde{\boldsymbol{A}}\boldsymbol{P}^{\perp}_{Q_t}\boldsymbol{q}^t \\ &+ [\boldsymbol{0}|\boldsymbol{M}_{t-1}]\,\boldsymbol{\Lambda}_t\left(\boldsymbol{Q}_t^T\boldsymbol{Q}_t\right)^{-1}\boldsymbol{Q}_t^T\boldsymbol{q}^t_{||} \\ &+ \boldsymbol{M}_t\left(\boldsymbol{M}_t^T\boldsymbol{M}_t\right)^{-1}\boldsymbol{H}_t^T\boldsymbol{q}^t_{\perp} - \boldsymbol{m}^{t-1}\oslash\boldsymbol{\lambda}_t \qquad (3.6.15)\end{aligned}$$

where $\overset{a}{=}$ is achieved based on (3.6.12) and the following

$$\begin{aligned}\boldsymbol{X}_t^T\boldsymbol{q}^t_{\perp} &= \boldsymbol{H}_t^T\boldsymbol{q}^t_{\perp} + \boldsymbol{\Xi}_t^T\boldsymbol{Q}_t^T\boldsymbol{q}^t_{\perp} \\ &= \boldsymbol{H}_t^T\boldsymbol{q}^t_{\perp}\end{aligned}$$

as $\boldsymbol{q}_{||}^t = \boldsymbol{Q}_t \overrightarrow{\boldsymbol{\beta}}$ and $\boldsymbol{Q}_t^T \boldsymbol{q}_{\perp}^t = \mathbf{0}$. Next we are going to show

$$
\begin{aligned}
&[\mathbf{0}|\boldsymbol{M}_{t-1}]\boldsymbol{\Lambda}_t \left(\boldsymbol{Q}_t^T \boldsymbol{Q}_t\right)^{-1} \boldsymbol{Q}_t^T \boldsymbol{q}_{||}^t \\
&\qquad + \boldsymbol{M}_t \left(\boldsymbol{M}_t^T \boldsymbol{M}_t\right)^{-1} \boldsymbol{H}_t^T \boldsymbol{q}_{\perp}^t - \boldsymbol{m}^{t-1} \oslash \boldsymbol{\lambda}_t = \boldsymbol{M}_t \vec{\boldsymbol{o}}_t \,(1)\,.
\end{aligned}
$$

The left-hand side can be treated as a linear combination of vectors $\boldsymbol{m}^0, ..., \boldsymbol{m}^{t-1}$. For any $l = 1, ..., t$ we are going to show that coefficients (denoted by diagonal matrices) of $\boldsymbol{m}^{l-1}$ converges to $\mathbf{0}$. The coefficient is

$$
\left[\left(\boldsymbol{M}_t^T \boldsymbol{M}_t\right)^{-1} \boldsymbol{H}_t^T \boldsymbol{q}_{\perp}^t\right]_l - \boldsymbol{\lambda}_l \left(-\boldsymbol{\beta}_l\right)^{[\![l \neq t}
\tag{3.6.16}
$$

where $[\![_{l \neq t}$ is the indicator function. Note: the subscript l is the index of the super component. Let $\boldsymbol{G} \coloneqq \frac{\boldsymbol{M}_t^T \boldsymbol{M}_t}{m}$ for simplicity, then

$$
\begin{aligned}
&\left[\left(\boldsymbol{M}_t^T \boldsymbol{M}_t\right)^{-1} \boldsymbol{H}_t^T \boldsymbol{q}_{\perp}^t\right]_l \\
=& \sum_{r=1}^{t} \left(\frac{\boldsymbol{M}_t^T \boldsymbol{M}_t}{m}\right)_{l,r}^{-1} \frac{n}{m} \left\langle \boldsymbol{h}^r, \boldsymbol{q}^t - \boldsymbol{q}_{||}^t \right\rangle \\
=& \sum_{r=1}^{t} \boldsymbol{G}_{l,r}^{-1} \frac{1}{\delta} \left\langle \boldsymbol{h}^r, \boldsymbol{q}^t - \boldsymbol{Q}_t \overrightarrow{\boldsymbol{\beta}} \right\rangle \\
=& \sum_{r=1}^{t} \boldsymbol{G}_{l,r}^{-1} \frac{1}{\delta} \left\langle \boldsymbol{h}^r, \boldsymbol{q}^t - \sum_{s=0}^{t-1} \boldsymbol{q}^s \oslash \boldsymbol{\beta}_s \right\rangle .
\end{aligned}
$$

We have

$$
\begin{aligned}
&\lim_{n\to\infty} \frac{1}{\delta} \left\langle \boldsymbol{h}^r, \boldsymbol{q}^t - \sum_{s=0}^{t-1} \boldsymbol{q}^s \oslash \boldsymbol{\beta}_s \right\rangle \\
&\overset{h}{=} \lim_{n\to\infty} \frac{1}{\delta} \left\langle \boldsymbol{h}^r, \boldsymbol{q}^t \right\rangle - \sum_{s=0}^{t-1} \lim_{n\to\infty} \frac{1}{\delta} \left\langle \boldsymbol{h}^r, \boldsymbol{q}^s \right\rangle \boldsymbol{\beta}_s
\end{aligned}
$$

and recall $\boldsymbol{q}^t = \boldsymbol{f}_t\left(\boldsymbol{h}^t, \boldsymbol{x}\right)$, then

$$\begin{aligned}
\lim_{n\to\infty} \frac{1}{\delta}\left\langle \boldsymbol{h}^r, \boldsymbol{q}^t \right\rangle &= \lim_{n\to\infty} \frac{1}{\delta}\left\langle \boldsymbol{h}^r, \boldsymbol{f}_t\left(\boldsymbol{h}^t, \boldsymbol{x}\right)\right\rangle \\
&\overset{a}{=} \lim_{n\to\infty} \frac{1}{\delta}\left\langle \boldsymbol{h}^r, \boldsymbol{h}^t \right\rangle \left(\boldsymbol{I} \odot \left\langle \boldsymbol{f}_t'\left(\boldsymbol{h}^t, \boldsymbol{x}\right)\right\rangle\right) \\
&\overset{h}{=} \lim_{n\to\infty} \frac{1}{\delta}\left\langle \boldsymbol{m}^{r-1}, \boldsymbol{m}^{t-1} \right\rangle \left(\boldsymbol{I} \odot \left\langle \boldsymbol{f}_t'\left(\boldsymbol{h}^t, \boldsymbol{x}\right)\right\rangle\right) \\
&\overset{b}{=} \lim_{n\to\infty} \boldsymbol{G}_{r,t}\boldsymbol{\lambda}_t
\end{aligned}$$

where $\overset{a}{=}$ holds based on Lemma 3.6.1 but only focus on the diagonal elements, $\overset{b}{=}$ holds due to the definition of $\boldsymbol{\lambda}_t = \frac{1}{\delta}\boldsymbol{\Sigma}_A \odot \langle \boldsymbol{f}_t'\left(\boldsymbol{h}^t, \boldsymbol{x}\right)\rangle$ and $\boldsymbol{\Sigma}_A = \boldsymbol{I}$. Thus, the coefficient of $\boldsymbol{m}^{l-1}$ can be computed via

$$\begin{aligned}
&\lim_{n\to\infty} \left\{ \sum_{r=1}^{t} \left(\boldsymbol{G}^{-1}\right)_{l,r} \left[\boldsymbol{G}_{r,t}\boldsymbol{\lambda}_t\right] - \sum_{r=1}^{t} \left(\boldsymbol{G}^{-1}\right)_{l,r} \left[\sum_{s=0}^{t-1} \boldsymbol{G}_{r,s}\boldsymbol{\lambda}_s\boldsymbol{\beta}_s\right] \right. \\
&\left. -\boldsymbol{\lambda}_l\left(-\boldsymbol{\beta}_l\right)^{[\![l\neq t}\right\} \\
&\overset{a.s}{=} \lim_{n\to\infty} \left\{ \boldsymbol{\lambda}_t\, [\![_{l=t} - \sum_{s=0}^{t-1} \boldsymbol{\lambda}_s\boldsymbol{\beta}_s\, [\![_{l=s} - \boldsymbol{\lambda}_l\left(-\boldsymbol{\beta}_l\right)^{[\![l\neq t} \right\} \\
&\overset{a.s}{=} \mathbf{0}.
\end{aligned}$$

where $\overset{a.s}{=}$ means the equality holds almost surely. Thus (3.6.15) can be simplified as

$$\boldsymbol{b}^t|_{\mathfrak{G}_{t,t}} \overset{d}{=} \boldsymbol{B}_t\left(\boldsymbol{Q}_t^T\boldsymbol{Q}_t\right)^{-1}\boldsymbol{Q}_t^T\boldsymbol{q}_{||}^t + \boldsymbol{P}_{\boldsymbol{M}_t}^{\perp}\tilde{\boldsymbol{A}}\boldsymbol{P}_{\boldsymbol{Q}_t}^{\perp}\boldsymbol{q}^t + \boldsymbol{M}_t\vec{\boldsymbol{o}}_t\left(1\right).$$

based on [29, Corollary 2], we can write

$$\begin{aligned}\overrightarrow{\boldsymbol{\beta}} &= \left(\frac{\boldsymbol{Q}_t^T\boldsymbol{Q}_t}{n}\right)^{-1}\frac{\boldsymbol{Q}_t^T\boldsymbol{q}^t}{n}\\ &= \left(\frac{\boldsymbol{Q}_t^T\boldsymbol{Q}_t}{n}\right)^{-1}\frac{\boldsymbol{Q}_t^T\boldsymbol{q}_{||}^t}{n}\end{aligned}$$

as $\boldsymbol{Q}_t^T\boldsymbol{q}_\perp^t = \mathbf{0}$ then we reach

$$\begin{aligned}\boldsymbol{b}^t|_{\mathfrak{G}_{t,t}} &\overset{d}{=} \sum_{i=0}^{t-1}\boldsymbol{\beta}_i \oslash \boldsymbol{b}^i + \boldsymbol{P}_{\boldsymbol{M}_t}^{\perp}\tilde{\boldsymbol{A}}\boldsymbol{P}_{\boldsymbol{Q}_t}^{\perp}\boldsymbol{q}^t + \boldsymbol{M}_t\vec{\boldsymbol{o}}_t\left(1\right).\\ &= \sum_{i=0}^{t-1}\boldsymbol{\beta}_i \oslash \boldsymbol{b}^i + \left(\boldsymbol{I}-\boldsymbol{P}_{\boldsymbol{M}_t}\right)\tilde{\boldsymbol{A}}\boldsymbol{q}_\perp^t + \boldsymbol{M}_t\vec{\boldsymbol{o}}_t\left(1\right)\\ &= \sum_{i=0}^{t-1}\boldsymbol{\beta}_i \oslash \boldsymbol{b}^i + \tilde{\boldsymbol{A}}\boldsymbol{q}_\perp^t - \boldsymbol{P}_{\boldsymbol{M}_t}\tilde{\boldsymbol{A}}\boldsymbol{q}_\perp^t + \boldsymbol{M}_t\vec{\boldsymbol{o}}_t\left(1\right)\\ &= \sum_{i=0}^{t-1}\boldsymbol{\beta}_i \oslash \boldsymbol{b}^i + \tilde{\boldsymbol{A}}\boldsymbol{q}_\perp^t + \boldsymbol{M}_t\vec{\boldsymbol{o}}_t\left(1\right)\end{aligned}$$

base on the fact that $\boldsymbol{P}_{\boldsymbol{M}_t}\tilde{\boldsymbol{A}}\boldsymbol{q}_\perp^t \overset{d}{=} \tilde{\boldsymbol{M}}_t\vec{\boldsymbol{o}}_t\left(1\right)$, which can be proved by rewriting $\boldsymbol{P}_{\boldsymbol{M}_t}\tilde{\boldsymbol{A}}\boldsymbol{q}_\perp^t$ in the $\mathrm{D}-$model then apply Lemma 3.6.2(c) to each $\boldsymbol{P}_{\boldsymbol{M}_{K,t}}\tilde{\boldsymbol{A}}_k\boldsymbol{q}_{k\perp}^t$ for $k\in[K]$, separately. Proof of (3.6.10) is similar.

3.6.5 Additional Lemmas

Lemma 3.6.1 (Stein's Lemma [37, Lemma 5])**.** *For jointly Gaussian random vectors $\boldsymbol{x}_1,\boldsymbol{x}_2\in\mathbb{R}^K$ with zero mean and any function $\boldsymbol{f}:\ \mathbb{R}^K\to\mathbb{R}^K$ where $E\left[\frac{\partial\boldsymbol{f}(\boldsymbol{x}_1)}{\partial\boldsymbol{X}}\right]$ and $E\left[\boldsymbol{x}_1\boldsymbol{f}\left(\boldsymbol{x}_2\right)^T\right]$ exist, the following holds*

$$E\left[\boldsymbol{x}_1\boldsymbol{f}\left(\boldsymbol{x}_2\right)^T\right] = \mathrm{Cov}\left(\boldsymbol{x}_1,\boldsymbol{x}_2\right)E\left[\frac{\partial\boldsymbol{f}\left(\boldsymbol{x}_2\right)}{\partial\boldsymbol{X}}\right]^T$$

Lemma 3.6.2 ([29, Lemma 2])**.** *For any deterministic $\boldsymbol{u}\in\mathbb{R}^n$ and $\boldsymbol{v}\in\mathbb{R}^m$*

with $\|\boldsymbol{u}\| = \|\boldsymbol{v}\| = 1$ *and a Gaussian matrix* $\tilde{\boldsymbol{A}}$ *distributed as* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *we have*

1. $\boldsymbol{v}^T \tilde{\boldsymbol{A}} \boldsymbol{u} \overset{d}{=} Z/\sqrt{m}$ *where* $Z \sim \mathcal{N}(0,1)$.

2. $\lim_{n\to\infty} \left\|\tilde{\boldsymbol{A}}\boldsymbol{u}\right\|^2 = 1$ *almost surely.*

3. *Consider, for* $d \leq m$, *a* d*-dimensional subspace* W *of* $\mathbb{R}^m$, *an orthogonal basis* $\boldsymbol{w}_1, ..., \boldsymbol{w}_d$ *of* W *with* $\|\boldsymbol{w}_i\|^2 = m$ *for* $i = 1, ..., d$, *and the orthogonal projection* $\boldsymbol{P}_W$ *onto* W. *Then for* $\boldsymbol{D} = [\boldsymbol{w}_1|...|\boldsymbol{w}_d]$, *we have* $\boldsymbol{P}_W \boldsymbol{A}\boldsymbol{u} \overset{d}{=} \boldsymbol{D}\boldsymbol{x}$ *with* $\boldsymbol{x} \in \mathbb{R}^d$ *that satisfies:* $\lim_{m\to\infty} \|\boldsymbol{x}\| \overset{a.s.}{=} 0$ *(the limit being taken with* d *fixed). Note that* $\boldsymbol{x}$ *is* $\vec{\boldsymbol{o}}_d(1)$ *as well.*

3.6.6 Proof of Lemma 3.5.1

Define the following probability function

$$p_{X|V}(\boldsymbol{x}|\boldsymbol{v}) \coloneqq \frac{1}{Z(\boldsymbol{v})} \exp\left(\phi_0(\boldsymbol{x}) + \boldsymbol{x}^T \boldsymbol{\Sigma}_e^{-1} \boldsymbol{v}\right),$$
$$Z(\boldsymbol{v}) \coloneqq \int \exp\left(\phi_0(\boldsymbol{x}) + \boldsymbol{x}^T \boldsymbol{\Sigma}_e^{-1} \boldsymbol{v}\right) d\boldsymbol{x},$$

where $Z(\boldsymbol{v})$ is a normalization constant, $\boldsymbol{x}$ and $\boldsymbol{v}$ are vectors with a same dimension and $\boldsymbol{\Sigma}_e$ denotes a positive semi-definite matrix.

Firstly, calculate the derivative of $Z(\boldsymbol{v})$:

$$\begin{aligned}
\frac{\partial Z(\boldsymbol{v})}{\partial \boldsymbol{v}} &= \frac{\partial}{\partial \boldsymbol{v}} \int \exp\left(\phi_0(\boldsymbol{x}) + \boldsymbol{x}^T \boldsymbol{\Sigma}_e^{-1} \boldsymbol{v}\right) d\boldsymbol{x}, \\
&= \left[\int \boldsymbol{x}^T \exp\left(\phi_0(\boldsymbol{x}) + \boldsymbol{x}^T \boldsymbol{\Sigma}_e^{-1} \boldsymbol{v}\right) d\boldsymbol{x}\right] \boldsymbol{\Sigma}_e^{-1}, \\
&= Z(\boldsymbol{v})\, \mathrm{E}_{X|V}[\boldsymbol{x}|\boldsymbol{v}]^T \boldsymbol{\Sigma}_e^{-1},
\end{aligned}$$

then rearrange the formula which provides

$$\begin{aligned}\mathrm{E}_{X|V}\left[\boldsymbol{x}|\boldsymbol{v}\right] &= \frac{\boldsymbol{\Sigma}_e}{Z\left(\boldsymbol{v}\right)}\left[\frac{\partial Z\left(\boldsymbol{v}\right)}{\partial \boldsymbol{v}}\right]^T, \\ &= \int \boldsymbol{x} p_{X|V}\left(\boldsymbol{x}|\boldsymbol{v}\right) d\boldsymbol{x}.\end{aligned}$$

Secondly, calculate the derivative of $\mathrm{E}_{X|V}\left[\boldsymbol{x}|\boldsymbol{v}\right]$:

$$\begin{aligned}&\frac{\partial \mathrm{E}_{X|V}\left(\boldsymbol{x}|\boldsymbol{v}\right)}{\partial \boldsymbol{v}} \\ =&\int \frac{\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{\Sigma}_e^{-1}}{Z\left(\boldsymbol{v}\right)}\exp\left(\phi_0\left(\boldsymbol{x}\right)+\boldsymbol{x}^T\boldsymbol{\Sigma}_e^{-1}\boldsymbol{v}\right)d\boldsymbol{x} \\ &-\int \frac{\boldsymbol{x}Z\left(\boldsymbol{v}\right)\mathrm{E}_{X|V}\left[\boldsymbol{x}|\boldsymbol{v}\right]^T\boldsymbol{\Sigma}_e^{-1}}{Z^2\left(\boldsymbol{v}\right)}\exp\left(\phi_0\left(\boldsymbol{x}\right)+\boldsymbol{x}^T\boldsymbol{\Sigma}_e^{-1}\boldsymbol{v}\right)d\boldsymbol{x}, \\ =&\left[\int \frac{\boldsymbol{x}\boldsymbol{x}^T}{Z\left(\boldsymbol{v}\right)}\exp\left(\phi_0\left(\boldsymbol{x}\right)+\boldsymbol{x}^T\boldsymbol{\Sigma}_e^{-1}\boldsymbol{v}\right)d\boldsymbol{x}\right]\boldsymbol{\Sigma}_e^{-1} \\ &-\left[\int \frac{\boldsymbol{x}}{Z\left(\boldsymbol{v}\right)}\exp\left(\phi_0\left(\boldsymbol{x}\right)+\boldsymbol{x}^T\boldsymbol{\Sigma}_e^{-1}\boldsymbol{v}\right)d\boldsymbol{x}\right]\mathrm{E}_{X|V}\left[\boldsymbol{x}|\boldsymbol{v}\right]^T\boldsymbol{\Sigma}_e^{-1}, \\ =&\mathrm{E}_{X|V}\left[\boldsymbol{x}\boldsymbol{x}^T|\boldsymbol{v}\right]\boldsymbol{\Sigma}_e^{-1}-\mathrm{E}_{X|V}\left[\boldsymbol{x}|\boldsymbol{v}\right]\mathrm{E}_{X|V}\left[\boldsymbol{x}|\boldsymbol{v}\right]^T\boldsymbol{\Sigma}_e^{-1}.\end{aligned}$$

Finally, multiply $\boldsymbol{\Sigma}_e$ on both sides, we achieve

$$\begin{aligned}\frac{\partial \mathrm{E}_{X|V}\left(\boldsymbol{x}|\boldsymbol{v}\right)}{\partial \boldsymbol{v}}\boldsymbol{\Sigma}_e &= \mathrm{E}_{X|V}\left[\boldsymbol{x}\boldsymbol{x}^T|\boldsymbol{v}\right]-\mathrm{E}_{X|V}\left[\boldsymbol{x}|\boldsymbol{v}\right]\mathrm{E}_{X|V}\left[\boldsymbol{x}|\boldsymbol{v}\right]^T, \\ &= \mathrm{Cov}\left(\boldsymbol{x}|\boldsymbol{v}\right).\end{aligned}$$

3.6.7 Proof of Estimation Error of an MMSE Estimator

Define the estimator $\hat{\boldsymbol{x}} = \boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}\right) = \mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}|\tilde{\boldsymbol{x}}\right]$, the covariance matrix of $\boldsymbol{x} - \hat{\boldsymbol{x}}$ which is defined as $\boldsymbol{\Sigma}_\eta$ can be calculated via

$$\begin{aligned}
\boldsymbol{\Sigma}_\eta &= \mathrm{E}\left[\left(\boldsymbol{x} - \hat{\boldsymbol{x}}\right)\left(\boldsymbol{x} - \hat{\boldsymbol{x}}\right)^T\right] \\
&= \mathrm{E}\left[\boldsymbol{x}\boldsymbol{x}^T - 2\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}|\tilde{\boldsymbol{x}}\right]\boldsymbol{x}^T + \mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}|\tilde{\boldsymbol{x}}\right]\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}|\tilde{\boldsymbol{x}}\right]^T\right] \\
&= \mathrm{E}_{\tilde{x}}\left[\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}\boldsymbol{x}^T - 2\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}|\tilde{\boldsymbol{x}}\right]\boldsymbol{x}^T + \mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}|\tilde{\boldsymbol{x}}\right]\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}|\tilde{\boldsymbol{x}}\right]^T\right]\right] \\
&= \mathrm{E}_{\tilde{x}}\left[\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}\boldsymbol{x}^T|\tilde{\boldsymbol{x}}\right]\right] - \mathrm{E}_{\tilde{x}}\left[\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}|\tilde{\boldsymbol{x}}\right]\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}|\tilde{\boldsymbol{x}}\right]^T\right] \\
&= \mathrm{E}_{\tilde{x}}\left[\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}\boldsymbol{x}^T|\tilde{\boldsymbol{x}}\right]\right] - \mathrm{E}_{\tilde{x}}\left[\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}\right)\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}\right)^T\right]
\end{aligned}$$

3.6.8 Proof of $\boldsymbol{\eta}\left(\cdot\right)$ with Bernoulli-Gaussian Prior

Recall the BG distribution:

$$p_x\left(\boldsymbol{x}\right) = \left(1-\epsilon\right)\delta^f_{\boldsymbol{x}=\boldsymbol{0}} + \epsilon p\left(\boldsymbol{x};\, \boldsymbol{0},\, \boldsymbol{\Sigma}_x\right), \tag{3.6.17}$$

where

$$p\left(\boldsymbol{x};\, \boldsymbol{0},\, \boldsymbol{\Sigma}_x\right) = \left|2\pi\boldsymbol{\Sigma}_x\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}_x^{-1}\boldsymbol{x}\right). \tag{3.6.18}$$

Define the system model

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{w}_e, \tag{3.6.19}$$

where $\boldsymbol{w}_e$ is the additive Gaussian noise, thus we have

$$p_{w_e}\left(\boldsymbol{w}_e; \boldsymbol{0},\, \boldsymbol{\Sigma}_e\right) = p\left(\boldsymbol{w}_e;\, \boldsymbol{0},\, \boldsymbol{\Sigma}_e\right), \tag{3.6.20}$$

The joint probability of $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ has the following formula,

$$p_{x,\tilde{x}}\left(\boldsymbol{x},\tilde{\boldsymbol{x}}\right)=p\left(\tilde{\boldsymbol{x}}-\boldsymbol{x};\mathbf{0},\,\boldsymbol{\Sigma}_e\right)\left(1-\epsilon\right)\delta^f_{\boldsymbol{x}=\mathbf{0}}+\epsilon p\left(\tilde{\boldsymbol{x}}-\boldsymbol{x};\,\mathbf{0},\,\boldsymbol{\Sigma}_e\right)p\left(\boldsymbol{x};\mathbf{0},\,\boldsymbol{\Sigma}_x\right),$$

and based on (3.6.19), we have

$$p_{\tilde{x}}\left(\tilde{\boldsymbol{x}}\right)=\left(1-\epsilon\right)p\left(\tilde{\boldsymbol{x}};\mathbf{0},\,\boldsymbol{\Sigma}_e\right)+\epsilon p\left(\tilde{\boldsymbol{x}};\mathbf{0},\,\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e\right).$$

Let $\hat{\boldsymbol{x}}=\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}|\tilde{\boldsymbol{x}}\right]$ and based on Bayes' theorem, we have

$$\begin{aligned}\hat{\boldsymbol{x}}&=\int\boldsymbol{x}\frac{p_{x,\tilde{x}}\left(\boldsymbol{x},\tilde{\boldsymbol{x}}\right)}{p_{\tilde{x}}\left(\tilde{\boldsymbol{x}}\right)}d\boldsymbol{x}\\&=\frac{\int\boldsymbol{x}p\left(\tilde{\boldsymbol{x}}-\boldsymbol{x};\,\mathbf{0},\,\boldsymbol{\Sigma}_e\right)p\left(\boldsymbol{x};\mathbf{0},\,\boldsymbol{\Sigma}_x\right)d\boldsymbol{x}}{\frac{(1-\epsilon)}{\epsilon}p\left(\tilde{\boldsymbol{x}};\mathbf{0},\,\boldsymbol{\Sigma}_e\right)+p\left(\tilde{\boldsymbol{x}};\mathbf{0},\,\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e\right)}.\end{aligned}\tag{3.6.21}$$

Based on Lemma 3.4.2, we have

$$\frac{\int\boldsymbol{x}p\left(\tilde{\boldsymbol{x}}-\boldsymbol{x};\,\mathbf{0},\,\boldsymbol{\Sigma}_e\right)p\left(\boldsymbol{x};\mathbf{0},\,\boldsymbol{\Sigma}_x\right)d\boldsymbol{x}}{p\left(\tilde{\boldsymbol{x}};\mathbf{0},\,\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e\right)}=\left(\boldsymbol{\Sigma}_x^{-1}+\boldsymbol{\Sigma}_e^{-1}\right)^{-1}\boldsymbol{\Sigma}_e^{-1}\tilde{\boldsymbol{x}}\tag{3.6.22}$$

which is the MMSE estimate for a Gaussian prior signal with an identity measurement matrix. Apply the fact (3.6.22) in (3.6.21), we will reach (3.5.1).

3.6.9 Proof of $\boldsymbol{\Sigma}_\eta$ with Bernoulli-Gaussian Prior

Define $\boldsymbol{\Sigma}_\eta=\mathrm{E}_{\tilde{x}}\left[\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}\boldsymbol{x}^T|\tilde{\boldsymbol{x}}\right]\right]-\mathrm{E}_{\tilde{x}}\left[\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}\right)\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}\right)^T\right]$ where $\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}\right)=\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}|\tilde{\boldsymbol{x}}\right]$. Assume the system model as (3.6.19) with signal and noise distributions as (3.6.17) and (3.6.20). Note that $\mathrm{E}_{\tilde{x}}\left[\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}\boldsymbol{x}^T|\tilde{\boldsymbol{x}}\right]\right]=\mathrm{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right]=\epsilon\boldsymbol{\Sigma}_x$ based on the law of total expectation. We focus on the calculation of the (p,q)-th elements of $\mathrm{E}_{\tilde{x}}\left[\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}\right)\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}\right)^T\right]$ and , let $\boldsymbol{R}=\left(\boldsymbol{\Sigma}_x^{-1}+\boldsymbol{\Sigma}_e^{-1}\right)^{-1}\boldsymbol{\Sigma}_e^{-1}$, based on

(3.6.21), we have

$$
\begin{aligned}
&\mathrm{E}_{\tilde{x}}\left[\left[\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}\right)\right]_p\left[\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}\right)\right]_q\right] \\
&= \mathrm{E}_{\tilde{x}}\left[\frac{\epsilon^2\int x_p p\left(\tilde{\boldsymbol{x}}-\boldsymbol{x};\,\mathbf{0},\,\boldsymbol{\Sigma}_e\right)p\left(\boldsymbol{x};\mathbf{0},\,\boldsymbol{\Sigma}_x\right)d\boldsymbol{x}\int x_q p\left(\tilde{\boldsymbol{x}}-\boldsymbol{x};\,\mathbf{0},\,\boldsymbol{\Sigma}_e\right)p\left(\boldsymbol{x};\mathbf{0},\,\boldsymbol{\Sigma}_x\right)d\boldsymbol{x}}{\left(\left(1-\epsilon\right)p\left(\tilde{\boldsymbol{x}};\mathbf{0},\,\boldsymbol{\Sigma}_e\right)+\epsilon p\left(\tilde{\boldsymbol{x}};\mathbf{0},\,\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e\right)\right)^2}\right] \\
&= \int\frac{\epsilon^2\int x_p p\left(\tilde{\boldsymbol{x}}-\boldsymbol{x};\,\mathbf{0},\,\boldsymbol{\Sigma}_e\right)p\left(\boldsymbol{x};\mathbf{0},\,\boldsymbol{\Sigma}_x\right)d\boldsymbol{x}\int x_q p\left(\tilde{\boldsymbol{x}}-\boldsymbol{x};\,\mathbf{0},\,\boldsymbol{\Sigma}_e\right)p\left(\boldsymbol{x};\mathbf{0},\,\boldsymbol{\Sigma}_x\right)d\boldsymbol{x}}{\left(1-\epsilon\right)p\left(\tilde{\boldsymbol{x}};\mathbf{0},\,\boldsymbol{\Sigma}_e\right)+\epsilon p\left(\tilde{\boldsymbol{x}};\mathbf{0},\,\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e\right)}d\tilde{\boldsymbol{x}} \\
&= \epsilon\int\frac{\frac{\int x_p p(\tilde{\boldsymbol{x}}-\boldsymbol{x};\mathbf{0},\boldsymbol{\Sigma}_e)p(\boldsymbol{x};\mathbf{0},\boldsymbol{\Sigma}_x)d\boldsymbol{x}}{p(\tilde{\boldsymbol{x}};\mathbf{0},\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e)}\frac{\int x_q p(\tilde{\boldsymbol{x}}-\boldsymbol{x};\mathbf{0},\boldsymbol{\Sigma}_e)p(\boldsymbol{x};\mathbf{0},\boldsymbol{\Sigma}_x)d\boldsymbol{x}}{p(\tilde{\boldsymbol{x}};\mathbf{0},\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e)}}{\frac{(1-\epsilon)p(\tilde{\boldsymbol{x}};\mathbf{0},\boldsymbol{\Sigma}_e)}{\epsilon p(\tilde{\boldsymbol{x}};\mathbf{0},\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e)}+\frac{1}{p(\tilde{\boldsymbol{x}};\mathbf{0},\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e)}}d\tilde{\boldsymbol{x}} \\
&\overset{(a)}{=} \int\frac{\left[\boldsymbol{R}\tilde{\boldsymbol{x}}\right]_p\left[\boldsymbol{R}\tilde{\boldsymbol{x}}\right]_q\epsilon p\left(\tilde{\boldsymbol{x}};\,0\,\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e\right)}{\frac{(1-\epsilon)}{\epsilon}\left|\boldsymbol{\Sigma}_x\left(\boldsymbol{\Sigma}_x^{-1}+\boldsymbol{\Sigma}_e^{-1}\right)\right|^{\frac{1}{2}}\exp\left(-\frac{1}{2}\tilde{\boldsymbol{x}}^T\boldsymbol{\Sigma}_e^{-1}\boldsymbol{R}\tilde{\boldsymbol{x}}\right)+1}d\tilde{\boldsymbol{x}}
\end{aligned}
$$

where $\overset{(a)}{=}$ is based on (3.6.22). The final $\boldsymbol{\Sigma}_\eta$ can be written in a compact form as

$$
\begin{aligned}
\boldsymbol{\Sigma}_\eta &= \mathrm{E}_{\tilde{x}}\left[\mathrm{E}_{x|\tilde{x}}\left[\boldsymbol{x}\boldsymbol{x}^T|\tilde{\boldsymbol{x}}\right]\right]-\mathrm{E}_{\tilde{x}}\left[\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}\right)\boldsymbol{\eta}\left(\tilde{\boldsymbol{x}}\right)^T\right] \\
&= \epsilon\boldsymbol{\Sigma}_x-\int\frac{\left[\boldsymbol{R}\tilde{\boldsymbol{x}}\right]\left[\boldsymbol{R}\tilde{\boldsymbol{x}}\right]^T\epsilon p\left(\tilde{\boldsymbol{x}};\,0\,\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e\right)}{\frac{(1-\epsilon)}{\epsilon}\left|\boldsymbol{\Sigma}_x\left(\boldsymbol{\Sigma}_x^{-1}+\boldsymbol{\Sigma}_e^{-1}\right)\right|^{\frac{1}{2}}\exp\left(-\frac{1}{2}\tilde{\boldsymbol{x}}^T\boldsymbol{\Sigma}_e^{-1}\boldsymbol{R}\tilde{\boldsymbol{x}}\right)+1}d\tilde{\boldsymbol{x}} \quad (3.6.23)
\end{aligned}
$$

3.6.10 Proof of $\boldsymbol{\eta}\left(\cdot\right)$ and $\boldsymbol{\Sigma}_\eta$ with Gaussian Prior

The results can be directly achieved from Lemma 3.4.2, but here, we derive the results based on the consequences from BG prior.

For the estimator $\boldsymbol{\eta}\left(\cdot\right)$, we start from (3.6.21), by setting $\epsilon=1$, we directly achieve

$$
\hat{\boldsymbol{x}}=\frac{\int\boldsymbol{x}p\left(\tilde{\boldsymbol{x}}-\boldsymbol{x};\,\mathbf{0},\,\boldsymbol{\Sigma}_e\right)p\left(\boldsymbol{x};\mathbf{0},\,\boldsymbol{\Sigma}_x\right)d\boldsymbol{x}}{p\left(\tilde{\boldsymbol{x}};\mathbf{0},\,\boldsymbol{\Sigma}_x+\boldsymbol{\Sigma}_e\right)},
$$

and based on (3.6.22) we can prove (3.5.7)

For the $\boldsymbol{\Sigma}_\eta$, we start form (3.6.23), by setting $\epsilon = 1$, we have

$$\begin{aligned}\boldsymbol{\Sigma}_\eta &= \boldsymbol{\Sigma}_x - \boldsymbol{R}\left[\int \tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^T \epsilon p\left(\tilde{\boldsymbol{x}};\, 0\,\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_e\right) d\tilde{\boldsymbol{x}}\right]\boldsymbol{R}^T, \\ &= \boldsymbol{\Sigma}_x - \boldsymbol{R}\mathrm{E}\left[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^T\right]\boldsymbol{R}^T. \end{aligned} \tag{3.6.24}$$

Recall that

$$\begin{aligned}\boldsymbol{R} &= \left(\boldsymbol{\Sigma}_x^{-1} + \boldsymbol{\Sigma}_e^{-1}\right)^{-1}\boldsymbol{\Sigma}_e^{-1}, \\ &= \boldsymbol{\Sigma}_x\left(\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_e\right)^{-1}, \end{aligned} \tag{3.6.25}$$

substituting (3.6.25) into (3.6.24) will give

$$\begin{aligned}\boldsymbol{\Sigma}_\eta &= \boldsymbol{\Sigma}_x - \boldsymbol{\Sigma}_x\left(\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_e\right)^{-1}\boldsymbol{\Sigma}_x, \\ &\overset{\text{(WMI)}}{=} \left(\boldsymbol{\Sigma}_x^{-1} + \boldsymbol{\Sigma}_e^{-1}\right)^{-1}, \end{aligned}$$

where $\overset{\text{(WMI)}}{=}$ is based on Woodbury Matrix Identity (WMI) listed below. The achieved result coincides with Lemma 3.4.2.

Lemma 3.6.3 (Woodbury matrix identity (WMI)[84])**.** *For matrices $\boldsymbol{A}$, $\boldsymbol{U}$, $\boldsymbol{C}$ and $\boldsymbol{V}$ of the correct sizes, assuming all the matrices have inverses, then we have the following*

$$\left(\boldsymbol{A} + \boldsymbol{UCV}\right)^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}\left(\boldsymbol{C}^{-1} + \boldsymbol{VA}^{-1}\boldsymbol{U}\right)^{-1}\boldsymbol{VA}^{-1}.$$

3.6.11 Heuristic State Evolution Analysis for Gaussian Signals

For the C-DCS model, we are tracking the covariance matrices of the equivalent noise ($\mathbf{\Sigma}_e^t$) and the estimation error ($\mathbf{\Sigma}_\eta^t$). The SE analysis is based on the following equations:

$$\mathbf{\Sigma}_e^t = \frac{1}{\delta}\left(\mathbf{\Sigma}_A\right).^2 \odot \mathbf{\Sigma}_\eta^t + \sigma_w^2 \boldsymbol{I}, \tag{3.6.26}$$

$$\mathbf{\Sigma}_\eta^t = \left(\mathbf{\Sigma}_x^{-1} + \left(\mathbf{\Sigma}_e^{t-1}\right)^{-1}\right)^{-1}. \tag{3.6.27}$$

where $\delta = \frac{m}{n}$.

Case 1: Assume $\mathbf{\Sigma}_x = \sigma_x^2 \boldsymbol{I}$ (independent signals), for both MMV and DCS cases, $\mathbf{\Sigma}_e^t$ and $\mathbf{\Sigma}_\eta^t$ will keep as diagonal matrices. Thus, (3.6.26) will degenerate into the following scalar version

$$\left(\sigma_e^t\right)^2 = \frac{1}{\delta}\frac{\sigma_x^2\left(\sigma_e^{t-1}\right)^2}{\sigma_x^2 + \left(\sigma_e^{t-1}\right)^2} + \sigma_w^2 \tag{3.6.28}$$

and by assumption $\sigma_e^{t-1} = \sigma_e^t = \sigma_e^\infty$ at the steady state point, we have equivalent noise with variance

$$\left(\sigma_e^\infty\right)^2 = \frac{\left(\frac{1-\delta}{\delta}\sigma_x^2 + \sigma_w^2\right) + \sqrt{\left(\frac{1-\delta}{\delta}\sigma_x^2 + \sigma_w^2\right)^2 + 4\sigma_w^2\sigma_x^2}}{2}$$

where the negative part has been ignored. The corresponding MSE can be calculated via $\delta\left(\left(\sigma_e^\infty\right)^2 - \sigma_w^2\right)$ which provides (3.5.9)

Case 2: Assume $\mathbf{\Sigma}_x = \sigma_x^2 \mathbf{1}$(identical signals) and consider DCS case where $\mathbf{\Sigma}_A = \boldsymbol{I}$. The inverse of $\mathbf{\Sigma}_x$ is not exist, thus, we consider the pseudo inverse $\mathbf{\Sigma}_x^+$ instead. The Singular Value Decomposition (SVD) of $\mathbf{\Sigma}_x$ and $\mathbf{\Sigma}_x^+$ can be

written as

$$\boldsymbol{\Sigma}_x = \sigma_x^2 \boldsymbol{U} \begin{bmatrix} K & \\ & \mathbf{0}_{K-1} \end{bmatrix} \boldsymbol{U}^T,\ \boldsymbol{\Sigma}_x^+ = \frac{1}{\sigma_x^2} \boldsymbol{U} \begin{bmatrix} \frac{1}{K} & \\ & \boldsymbol{\infty}_{K-1} \end{bmatrix} \boldsymbol{U}^T$$

where $\boldsymbol{U}$ is an orthogonal matrix, $\mathbf{0}_{K-1}$, $\boldsymbol{\infty}_{K-1} \in \mathbb{R}^{(K-1)\times(K-1)}$ are diagonal matrices with diagonal elements 0 and ∞, respectively. Due to the property of $\boldsymbol{\Sigma}_A$, $\boldsymbol{\Sigma}_e^t = \left(\sigma_e^t\right)^2 \boldsymbol{I}$ will always be a diagonal matrix, thus we can write the following decomposition

$$\boldsymbol{\Sigma}_e^t = \left(\sigma_e^t\right)^2 \boldsymbol{U}\boldsymbol{U}^T,\ \left(\boldsymbol{\Sigma}_e^t\right)^{-1} = \frac{1}{\left(\sigma_e^t\right)^2} \boldsymbol{U}\boldsymbol{U}^T,$$

and $\boldsymbol{\Sigma}_\eta^t$ can be achieved via

$$\begin{aligned} \left(\boldsymbol{\Sigma}_x^+ + \left(\boldsymbol{\Sigma}_e^t\right)^{-1}\right)^{-1} &= \left(\boldsymbol{U} \begin{bmatrix} \frac{1}{\sigma_x^2 K} + \frac{1}{\left(\sigma_e^t\right)^2} & \\ & \boldsymbol{\infty}_{K-1} \end{bmatrix} \boldsymbol{U}^T\right)^{-1} \\ &= \frac{\left(\sigma_e^t\right)^2}{\sigma_x^2 K + \left(\sigma_e^t\right)^2} \boldsymbol{\Sigma}_x. \end{aligned}$$

We only focus on the diagonal elements of $\left(\boldsymbol{\Sigma}_x^+ + \left(\boldsymbol{\Sigma}_e^t\right)^{-1}\right)^{-1}$which should be $\frac{\left(\sigma_e^t\right)^2 \sigma_x^2}{\sigma_x^2 K + \left(\sigma_e^t\right)^2}$, the state evolution will also degenerate into the following scalar form

$$\left(\sigma_e^{t+1}\right)^2 = \frac{1}{\delta} \frac{\left(\sigma_e^t\right)^2 \sigma_x^2}{\sigma_x^2 K + \left(\sigma_e^t\right)^2} + \sigma_w^2 \tag{3.6.29}$$

notice the difference between (3.6.29) and (3.6.28). Following the rest analysis steps in Case 1, we will achieve (3.5.11)

Case 3: Assume $\boldsymbol{\Sigma}_x = \sigma_x^2 \mathbf{1}$(identical signals) and consider MMV case where $\boldsymbol{\Sigma}_A = \mathbf{1}$. The analysis is the same as in DCS case, the only difference is

that $\mathbf{\Sigma}_e^t$ is no longer a diagonal matrix. Instead, we have

$$\alpha^t \mathbf{1} = \alpha^t \boldsymbol{U} \begin{bmatrix} K & & \\ & & \\ & & \mathbf{0}_{K-1} \end{bmatrix} \boldsymbol{U}^T \coloneqq (\mathbf{\Sigma}_A) .^2 \odot \mathbf{\Sigma}_\eta^t$$

then

$$\begin{aligned} \mathbf{\Sigma}_e^t &= \frac{1}{\delta}\alpha^t \mathbf{1} + \sigma_w^2 \boldsymbol{U}\boldsymbol{U}^T \\ &= \boldsymbol{U} \begin{bmatrix} \frac{1}{\delta}\alpha^t K + \sigma_w^2 & & \\ & & \\ & & \sigma_w^2 \boldsymbol{I}_{K-1} \end{bmatrix} \boldsymbol{U}^T \end{aligned} \tag{3.6.30}$$

and $\mathbf{\Sigma}_\eta^t$ can be achieved via

$$\begin{aligned} \left(\mathbf{\Sigma}_x^+ + \left(\mathbf{\Sigma}_e^t\right)^{-1}\right)^{-1} &= \left(\boldsymbol{U} \begin{bmatrix} \frac{1}{\sigma_x^2 K} + \frac{1}{\frac{1}{\delta}\alpha^t K + \sigma_w^2} & & \\ & & \\ & & \boldsymbol{\infty}_{K-1} \end{bmatrix} \boldsymbol{U}^T\right)^{-1} \\ &= \frac{\left(\frac{1}{\delta}\alpha^t K + \sigma_w^2\right)}{\frac{1}{\delta}\alpha^t K + \sigma_w^2 + \sigma_x^2 K} \mathbf{\Sigma}_x = \mathbf{\Sigma}_\eta^{t+1} \end{aligned}$$

which indicates

$$\alpha^{t+1} = \frac{\left(\frac{1}{\delta}\alpha^t K + \sigma_w^2\right)\sigma_x^2}{\frac{1}{\delta}\alpha^t K + \sigma_w^2 + \sigma_x^2 K}.$$

We still focus on the diagonal elements of $\left(\mathbf{\Sigma}_x^+ + \left(\mathbf{\Sigma}_e^t\right)^{-1}\right)^{-1}$, define

$$\left(\sigma_e^t\right)^2 \boldsymbol{I} \coloneqq \mathrm{Diag}\left(\mathbf{\Sigma}_e^t\right)$$

where $\mathrm{Diag}(\cdot)$ is the operator that keeps the diagonal elements of a matrix

unchanged but set others to zero and based on (3.6.30), we have

$$\begin{aligned} \left(\sigma_e^t\right)^2 &= \frac{1}{\delta}\alpha^t + \sigma_w^2, \\ \alpha^t &= \delta\left(\sigma_e^t\right)^2 - \delta\sigma_w^2, \end{aligned} \tag{3.6.31}$$

then α^{t+1} can be rewritten as a function of σ_e^t, which gives

$$\alpha^{t+1} = \frac{K\left(\sigma_e^t\right)^2\sigma_x^2 - (K-1)\,\sigma_w^2\sigma_x^2}{K\left(\sigma_e^t\right)^2 + K\sigma_x^2 - (K-1)\,\sigma_w^2}. \tag{3.6.32}$$

Substituting (3.6.32) into (3.6.31) will provide

$$\left(\sigma_e^{t+1}\right)^2 = \frac{1}{\delta}\frac{K\left(\sigma_e^t\right)^2\sigma_x^2 - (K-1)\,\sigma_w^2\sigma_x^2}{K\left(\sigma_e^t\right)^2 + K\sigma_x^2 - (K-1)\,\sigma_w^2} + \sigma_w^2.$$

The following steps are the same as in **Case 1** and finally achieve (3.5.10).

Chapter 4

Number of Measurements Selection via AMP

In this chapter, we consider a practical signal transmission/receiving application with fixed energy budget such as radar/sonar. The system is modelled by linear equations with the assumption that the total energy that can be allocated to signals is fixed and thermal noise is the dominant noise source. Under this circumstances, we discover that the signal energy per measurement decreases linearly and the noise energy per measurement increases approximately linearly with the increase of the number of measurements. Thus the SNR decreases quadratically with the number of measurements. This model suggests an optimal operation point different from the common wisdom where more measurements often mean a better performance. Our analysis shows that there is an optimal number of measurements, neither too few nor too many, to minimize the mean squared error of the estimate. The analysis is based on a state evolution technique which is proposed for the approximate message passing algorithm. We consider the Gaussian, BG and LF distributions (when the

soft-thresholding function is chosen as the estimator) in both real and complex domains. Numerical results justify the correctness of our analysis.

4.1 Introduction

This chapter focuses on a system design inspired by practical scenarios where the total energy budget of the linear measurements is fixed, the signal energy per measurement decreases linearly and the noise energy per measurement increases approximately linearly with the number of measurements. This scenario arises in many active sensing applications where measuring means observing the responses of a physical system to the stimulants that we actively put in. One example is radar systems. The number of measurements could correspond to the number of pulses per unit time (pulse frequency) or the number of sub-channels in the entire spectrum. When the number of measurements is increasing, the signal energy per measurement (per pulse/sub-channel) is decreasing linearly with the number of measurements. For the measurement noise, we adopt the commonly used additive white Gaussian noise model. Based on the famous thermal noise effect of a sampling circuit [85, 86, 87], which shows that the noise in a sampling circuit increases with the increase of the sampling rate, we assume that the noise variance increases approximately linearly with the increase of the number of measurements. With the above assumptions, the SNR per measurement should decrease quadratically if we add more measurements. Our goal in this chapter is to address this trade-off and determine the optimal number of measurements. It is worth noting that although this chapter focuses on sparse signals, the same trade-off exists for non-sparse signals as we show in Section 4.3.

The main contribution of this work is that we analyse the quadratic decreasing SNR model and find the exact optimal number of measurements required to minimize the MSE under certain mathematical assumptions. For the purpose of analysis, we assume a Gaussian measurement matrix, i.e., the elements in the measurement matrix are independently drawn from a Gaussian distribution. Let m be the number of measurements, n be the dimension of the unknown signal, and S be the number of non-zero elements. Further, let $m,\ n,\ S \to \infty$ with constant ratios $\delta := \frac{m}{n}$ (normalized number of measurements, under-sampling ratio) and $\epsilon := \frac{S}{n}$ (sparsity level). By characterizing the asymptotic distortion as a function of the normalized number of measurements δ, one can find the optimal number of measurements $\delta^\dagger$ that minimizes the distortion. The $\delta^\dagger$ may be directly achieved by a closed-form formula or by numerical calculation which depends on the statistics of the unknown signal. In order to provide intuition about the value of $\delta^\dagger$ for different unknown signals, we study upper bounds on $\delta^\dagger$ for three typical signal models: Gaussian, BG and LF distributions in both real and complex domains. The first two signal models are commonly used for non-sparse and sparse signal analysis, respectively. The third model is used for worst case analysis meaning the resulting MSE performance is an upper bound on that of signals with arbitrary distribution with the same sparsity level. The worst case analysis result is pessimistic in general but at the same time universal. Our analysis shows that for all three models, in both real and complex domains, the optimal value $\delta^\dagger$ is upper bounded by 2.

Our results are based on the AMP algorithm and the associated state evolution analysis. It is noteworthy that though the rigorous derivation of state evolution of AMP requires a random Gaussian matrix, numerical results

in [28] show that the same results are relatively accurate for partial Fourier and Rademacher matrices when the sizes of these matrices are sufficiently large.

4.2 Problem Formulation

4.2.1 System Model

Consider a signal transmission/receiving system with fixed energy budget such as radar/sonar. If we sample a signal with more measurements m then the energy allocated to each single measurement is reduced. Thus, we assume the signal variance is proportional to m^{-1}. This effect can be modelled by multiplying each measurement by a factor of $1/\sqrt{m}$. In addition, based on [88], when considering the noise power of a receiving system especially for radar, an adequate assumption is that the receiver system is ideal and only consider the thermal noise. It is known that the thermal noise in a sampling circuit increases with the sampling rate [85, 86, 87], we assume a linear relationship between the noise variance (σ_w^2) and the number of measurements (m). Let $\mathbb{H}$ represents the real ($\mathbb{R}$) or complex ($\mathbb{C}$) domain. The system is modelled as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}, \tag{4.2.1}$$

where $\boldsymbol{y} \in \mathbb{H}^m$ denotes the observation vector; $\boldsymbol{A} \in \mathbb{H}^{m\times n}$ is the standard Gaussian random matrix with elements scaled by $1/\sqrt{m}$; $\boldsymbol{x} \in \mathbb{H}^n$ represents the unknown signal and $\boldsymbol{w} \in \mathbb{H}^m$ is additive Gaussian noise with mean zero. In addition, based on the assumption of the linear relationship between σ_w^2 and m, we define

$$\sigma_w^2 \coloneqq \delta\sigma_0^2, \tag{4.2.2}$$

where $\delta \coloneqq \frac{m}{n}$ and σ_0^2 denotes the noise base level which is a constant. Let $\hat{\boldsymbol{x}}$ be the estimated signal. The performance of the system is given by the MSE

$$\text{Err} \coloneqq \lim_{n\to\infty} \frac{1}{n} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 . \tag{4.2.3}$$

In particular, we are interested in the value of δ that minimizes the MSE (4.2.3). We consider the system model (4.2.1) for both non-sparse and sparse signals.

4.2.2 Non-Sparse Setting

For the non-sparse setting, we consider the widely used Gaussian signal as an example. The asymptotic MSE analysis for the traditional problem is well known. Assume that $\boldsymbol{A}$ is a Gaussian random matrix with i.i.d. elements drawn from $\mathcal{N}\left(0, \frac{1}{m}\right)$ when $\mathbb{H} = \mathbb{R}$ (or $\mathcal{CN}\left(0, \frac{1}{m}\right)$ when $\mathbb{H} = \mathbb{C}$), $\boldsymbol{x}$ is drawn from $\mathcal{N}\left(\boldsymbol{0}, \sigma_x^2 \boldsymbol{I}\right)$ when $\mathbb{H} = \mathbb{R}$ (or $\mathcal{CN}\left(\boldsymbol{0}, \sigma_x^2 \boldsymbol{I}\right)$ when $\mathbb{H} = \mathbb{C}$) and the noise $\boldsymbol{w}$ is drawn from $\mathcal{N}\left(\boldsymbol{0}, \sigma_w^2 \boldsymbol{I}\right)$ when $\mathbb{H} = \mathbb{R}$ (or $\mathcal{CN}\left(\boldsymbol{0}, \sigma_w^2 \boldsymbol{I}\right)$ when $\mathbb{H} = \mathbb{C}$). The asymptotic MSE of the MMSE estimator can be directly calculated based on random matrix theory [29]. Denoting $c = \frac{(1-\delta)}{\delta}$, we have

$$\lim_{n\to\infty} \frac{1}{n} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 = \frac{\delta}{2} \left[\left(c\sigma_x^2 - \sigma_w^2 \right) + \sqrt{\left(c\sigma_x^2 + \sigma_w^2 \right)^2 + 4\sigma_w^2 \sigma_x^2} \right] . \tag{4.2.4}$$

By replacing the noise variance with our model (4.2.2), a trade-off between MSE and δ is achieved. Figure 4.2.1 plots an example, where we set $\sigma_x^2 = 1$ and vary the value of σ_0^2. For each given σ_0^2, by increasing the number of measurements, the MSE first decreases until reaches the optimal point; further increasing the number of measurements, the MSE becomes larger.

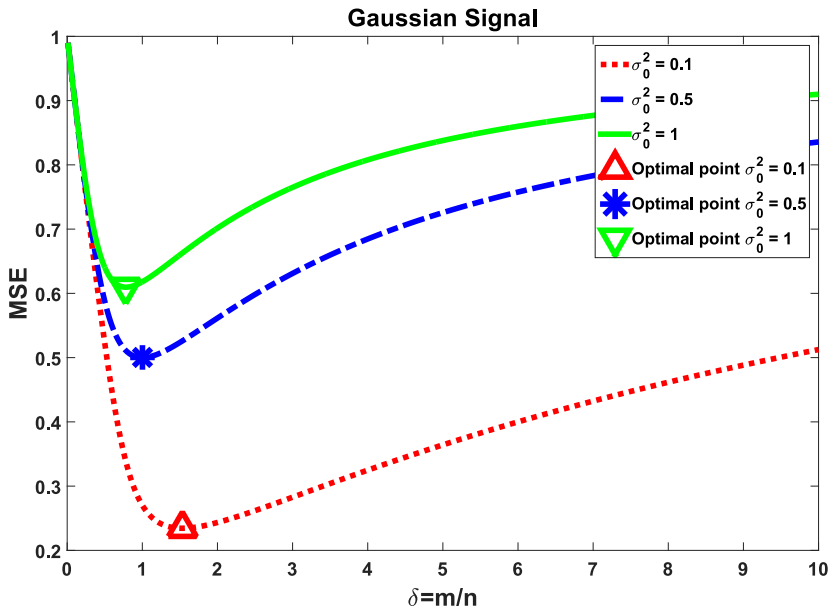


Figure 4.2.1: Trade-off for Gaussian signals. σ_0^2 is the noise base level. The optimal δ decreases with increasing σ_0^2.

For sparse signals, we want to find a similar relationship, taking into account the non-linear property of the sparse decoder. Instead, we use the state evolution technique of the well known AMP algorithm [29, 1]. Although AMP was originally proposed for solving CS problems in which the unknown signals are assumed to be S-sparse, we will show that the same analysis is also valid for non-sparse signals in Section 4.3.3. The background of AMP has been given in Chapter 2. Here we just recall the key parts. At each iteration, AMP estimates the signal based on (2.2.1) which requires the information of the ground truth signal and the equivalent noise. When the actual distribution of $\boldsymbol{x}$ is unknown, a worse case analysis will be applied (see Section 2.1). When the distribution of $\boldsymbol{x}$ is given, then the corresponding MMSE estimator will be used (see Section 3.2.2). The statistics of the equivalent noise is calculated based on (2.2.2) which requires the knowledge of the estimation error of the

previously iteration. When the measurement matrix is a standard Gaussian random matrix, the performance of AMP can be described by SE which tracks the variance of the equivalent noise via

$$\left(\sigma_e^t\right)^2 = \frac{1}{\delta}\mathrm{Err}_t + \sigma_w^2, \tag{4.2.5}$$

where Err_t represents the estimation error of the previously iteration which is defined by (2.2.4). Thus, in order to apply AMP algorithm and the corresponding SE technique, we need to choose the estimator $\eta(\cdot)$ and calculate Err_t, properly. When AMP algorithm converges, we have $t \to \infty$, $\sigma_e^t = \sigma_e^{t+1} = \sigma_e^\infty$ and $\mathrm{Err}_t = \mathrm{Err}_{t+1} = \mathrm{Err}_\infty$. In the following sections, we will study the relationship between MSE (Err_∞) and δ according to some specific signal distributions such as LF and BG distributions in both real and complex domains.

4.3 Analysis in Real Domain

In this section, we analyse the relationship between MSE and δ (or equivalently m) in the real domain for both LF and BG distributions. Then we extend the analysis to the complex domain in the next section. We first consider the situation that the actual distribution of the unknown signal is not given and we only know the sparsity level $\epsilon = \frac{S}{n}$. The analysis in this case will provide a worst case universal solution. A designed decoder based on a given signal distribution should outperform the universal decoder. To study this case, we consider the BG distribution.

4.3.1 Least-Favourite Distribution (Worst Case Analysis)

For the worst case analysis, the soft-thresholding function (2.1.3) will be chosen as the $\eta(\cdot)$ function and the corresponding LF distribution is defined by (2.1.5). The optimal threshold value θ^t and Err_{t+1}, at each iteration, are calculated via

$$\theta^t := \alpha^\dagger \sigma_e^t,\ \alpha^\dagger = \arg\min_{\alpha \in \mathbb{R}_+} M(\epsilon, \alpha)$$

and

$$\mathrm{Err}_{t+1} = M\left(\epsilon, \alpha^\dagger\right)\left(\sigma_e^t\right)^2. \tag{4.3.1}$$

where $M\left(\epsilon, \alpha^\dagger\right)$ is given by (2.2.7).

We apply the above results to our system model and achieve the following theorem

Theorem 4.3.1. *For a linear measurement system (4.2.1) with signal model (2.1.5) and additive white Gaussian noise with variance (4.2.2), apply AMP algorithm with estimator (2.1.3). By the convergence assumption of (4.2.5), we have*

$$\delta^\dagger = 2M\left(\epsilon, \alpha^\dagger\right), \tag{4.3.2}$$

which is independent of the noise variance.

Proof. By the convergence assumption, when $t \to \infty$, we have $\sigma_e^{t+1} = \sigma_e^t = \sigma_e^\infty$ and $\mathrm{Err}_{t+1} = \mathrm{Err}_t = \mathrm{Err}_\infty$. Substituting (4.2.5) into (4.3.1) provides $\mathrm{Err}_\infty = M\left(\epsilon, \alpha^\dagger\right)\left(\frac{1}{\delta}\mathrm{Err}_\infty + \delta\sigma_0^2\right)$ which can be rewritten as $\mathrm{Err}_\infty = \frac{M\left(\epsilon,\alpha^\dagger\right)\delta^2\sigma_0^2}{\delta - M\left(\epsilon,\alpha^\dagger\right)}$. Now Err_∞ is a function of δ. Take the derivative of Err_∞ with respect to δ and set it equal to zero, for $\delta > M\left(\epsilon, \alpha^\dagger\right)$ (which ensures that Err_∞ is a positive value),

we have the only saddle point $\delta^\dagger = 2M\left(\epsilon, \alpha^\dagger\right)$. As $\delta \to \infty$, we have $\mathrm{Err}_\infty \to \infty$, thus, $\delta^\dagger$ is a local minima which is the required solution. In addition, $\delta^\dagger$ does not depend on the base noise level σ_0^2, that is because $\mathrm{Err}_\infty = \frac{M\left(\epsilon,\alpha^\dagger\right)\delta^2\sigma_0^2}{\delta - M\left(\epsilon,\alpha^\dagger\right)}$ and σ_0^2 works as a scalar factor which will not affect $\delta^\dagger$. □

4.3.2 Bernoulli-Gaussian Distribution

Next we consider the BG prior [89, 72, 43] with probability density given by

$$p_x = (1-\epsilon)\,\delta^f_{x=0} + \epsilon p_G\left(x;\, 0,\, \sigma_x^2\right), \tag{4.3.3}$$

where $p_G\left(x;\, 0,\, \sigma_x^2\right)$ represents the Gaussian density with mean 0 and variance σ_x^2.

The $\eta\left(\cdot\right)$ function can be designed based on the prior information of $\boldsymbol{x}$. Let $R^t \coloneqq \sigma_x^2 / \left(\left(\sigma_e^t\right)^2 + \sigma_x^2\right)$ and define

$$I\left(R^t, \epsilon\right) \coloneqq \int \frac{\phi\left(x\right)}{1 + \frac{1-\epsilon}{\epsilon}\frac{1}{\sqrt{1-R^t}}\exp\left(-\frac{R^t}{1-R^t}\frac{x^2}{2}\right)} x^2 dx. \tag{4.3.4}$$

The component-wise function $\eta\left(\cdot\right)$ can be chosen as the MMSE estimator, for each element of $\tilde{\boldsymbol{x}}^t$:

$$\eta\left(\tilde{x}_i^t\right) \coloneqq \frac{p_G\left(\tilde{x}_i^t; 0, \left(\sigma_e^t\right)^2 + \sigma_x^2\right)}{p\left(\tilde{x}_i^t\right)} \epsilon R^t \tilde{x}_i^t, \tag{4.3.5}$$

with $p\left(\tilde{x}_i^t\right) \coloneqq (1-\epsilon)\, p_G\left(\tilde{x}_i^t; 0, \left(\sigma_e^t\right)^2\right) + \epsilon p_G\left(\tilde{x}_i^t; 0, \left(\sigma_e^t\right)^2 + \sigma_x^2\right)$. For simplified notation, define

$$v_1 \coloneqq \frac{1-\epsilon}{\epsilon}\sqrt{\frac{\left(\sigma_e^t\right)^2 + \sigma_x^2}{\left(\sigma_e^t\right)^2}},\; v_2 \coloneqq \frac{R^t}{\left(\sigma_e^t\right)^2},\; v_3 \coloneqq v_1 \exp\left(-\frac{1}{2} v_2 \left(\tilde{x}_i^t\right)^2\right),$$

we have $\eta'\left(x_i^t|\tilde{x}_i^t\right) = R^t/\left(v_3+1\right) + R^t v_3 v_2 \left(\tilde{x}_i^t\right)^2/\left(v_3+1\right)^2$ and

$$\mathrm{Err}_{t+1} \coloneqq \left[\frac{R^t\epsilon}{1-R^t}\left(1-R^t I\left(R^t,\epsilon\right)\right)\right]\left(\sigma_e^t\right)^2. \tag{4.3.6}$$

(See Section 4.6.1 for the detailed proof.)

Fast calculation of Err_{t+1}: We can increase the efficiency of AMP algorithm by avoiding the integration of (4.3.4). Based on Lemma 4.3.2, Err_{t+1} can be approximately calculated by

$$\mathrm{Err}_{t+1} \approx \left[\frac{1}{n}\sum_{i=1}^{n}\eta'\left(\tilde{x}_i^t\right)\right]\left(\sigma_e^t\right)^2. \tag{4.3.7}$$

Proof. As mentioned before, the input of $\eta\left(\cdot\right)$ can be written as $\tilde{x} = x + w_e$ (we ignore the subscript i and superscript t for simplification). Consider the conditional probability

$$p\left(x|\tilde{x}\right) = \frac{p\left(x,\tilde{x}\right)}{p\left(\tilde{x}\right)} = \frac{\left(1-\epsilon\right)p_G\left(\tilde{x}-x;0,\sigma_e^2\right)\delta_{x=0}^f}{p\left(\tilde{x}\right)} + \frac{\epsilon p_G\left(x;0,\sigma_x^2\right)p_G\left(\tilde{x}-x;0,\sigma_e^2\right)}{p\left(\tilde{x}\right)},$$

in which we only care about the second term (the first term has no contributions to $\mathbb{E}\left[x|\tilde{x}\right]$ and $\mathrm{var}\left[x|\tilde{x}\right]$ due to $\delta_{x=0}^f$). The numerator of the second term can be written as

$$\begin{aligned}\epsilon p_G\left(x;0,\sigma_x^2\right)p_G\left(\tilde{x}-x;0,\sigma_e^2\right) &= \epsilon\frac{1}{\sqrt{2\pi\sigma_x^2}}\exp\left(-\frac{x^2}{2\sigma_x^2}\right)\frac{1}{\sqrt{2\pi\sigma_e^2}}\exp\left(-\frac{\left(\tilde{x}-x\right)^2}{2\sigma_e^2}\right)\\ &= \epsilon\frac{1}{2\pi\sigma_x\sigma_e}\exp\left(-\frac{\tilde{x}^2}{2\sigma_e^2}\right)\exp\left(\frac{-\sigma_e^2-\sigma_x^2}{2\sigma_x^2\sigma_e^2}x^2+\frac{x\tilde{x}}{\sigma_e^2}\right).\end{aligned}$$

The term $\epsilon\frac{1}{2\pi\sigma_x\sigma_e}\exp\left(-\frac{\tilde{x}^2}{2\sigma_e^2}\right)$ can be moved to the denominator. Compare the remaining part $\exp\left(\frac{-\sigma_e^2-\sigma_x^2}{2\sigma_x^2\sigma_e^2}x^2+\frac{x\tilde{x}}{\sigma_e^2}\right)$ with the term $\exp\left(\phi_0\left(u\right)+uv\right)$ in Lemma

4.3.2 listed below, we have $u = \frac{x}{\sigma_e^2}$ and $v = \tilde{x}$. Based on (4.3.5) and Lemma 4.3.2, we have

$$\mathbb{E}\left[U|V=\tilde{x}\right] = \frac{\mathbb{E}\left[X|V=\tilde{x}\right]}{\sigma_e^2} = \frac{\eta\left(\tilde{x}\right)}{\sigma_e^2},$$
$$\mathrm{var}\left(U|V=\tilde{x}\right) = \frac{\eta'\left(\tilde{x}\right)}{\sigma_e^2} = \mathrm{var}\left(\frac{X}{\sigma_e^2}|V=\tilde{x}\right) = \frac{1}{\sigma_e^4}\mathrm{var}\left(X|V=\tilde{x}\right),$$

which provides

$$\mathrm{var}\left(X|V=\tilde{x}\right) = \eta'\left(\tilde{x}\right)\sigma_e^2.$$

Recall that the MSE considers the average value of $\mathrm{var}\left(X|V=\tilde{x}\right)$ with respect to different $\tilde{x}$'s, thus (4.3.7) is proved. □

Lemma 4.3.2. *[30, Lemma 2] Consider a random variable U with a conditional probability density function of the form $p_{U|V}\left(u|v\right) \coloneqq \frac{1}{Z(v)}\exp\left(\phi_0\left(u\right)+uv\right)$, where $Z\left(v\right)$ is a normalization constant, Then,*

$$\frac{\partial}{\partial v}\log Z\left(v\right) = \mathbb{E}\left[U|V=v\right]$$
$$\frac{\partial^2}{\partial v^2}\log Z\left(v\right) = \frac{\partial}{\partial v}\mathbb{E}\left[U|V=v\right] = \mathrm{var}\left(U|V=v\right).$$

We do not have a closed form of Err_t, thus we cannot directly achieve the optimal $\delta^\dagger$ as in Theorem 4.3.1. On the other hand, when AMP converges, Err_t and σ_e^t will converge to fixed points Err_∞ and σ_e^∞, respectively. Based on the relationship between Err_∞ and σ_e^∞ given in (4.3.6), the optimal $\delta^\dagger$ can be obtained by the following theorem.

Theorem 4.3.3. *For a linear measurement system (4.2.1) with signal model (4.3.3) and additive white Gaussian noise with variance (4.2.2), apply AMP algorithm with estimator (4.3.5). For any given set of parameters $\{\epsilon,\ \sigma_e^\infty,\ \sigma_x^2,\ \sigma_0^2\}$*

such that $(\sigma_e^\infty)^4 - 4\sigma_0^2 \mathrm{Err}_\infty \geq 0$*, by the convergence assumption of (4.2.5), we have*

$$\delta = \frac{(\sigma_e^\infty)^2 \pm \sqrt{(\sigma_e^\infty)^4 - 4\sigma_0^2 \mathrm{Err}_\infty}}{2\sigma_0^2}. \tag{4.3.8}$$

The optimal value $\delta^\dagger = \frac{(\sigma_e^\infty)^2}{2\sigma_0^2}$ *is achieved when* $(\sigma_e^\infty)^4 = 4\sigma_0^2 \mathrm{Err}_\infty$.

Proof. If AMP algorithm converges, in which $t \to \infty$, $\sigma_e^{t+1} = \sigma_e^t = \sigma_e^\infty$ and $\mathrm{Err}_{t+1} = \mathrm{Err}_t = \mathrm{Err}_\infty$. Based on (4.2.5), we have the following equation

$$(\sigma_e^\infty)^2 = \frac{1}{\delta}\mathrm{Err}_\infty + \delta\sigma_0^2, \tag{4.3.9}$$

where Err_∞ is a function of σ_e^∞ (4.3.6). Treat δ as the only unknown variable, we achieve the corresponding solutions (4.3.8). For any given set of parameters $\{\epsilon,\ \sigma_e^\infty,\ \sigma_x^2,\ \sigma_0^2\}$ such that δ is a positive real value, we say that this is a valid parameter set. Because $\sqrt{(\sigma_e^\infty)^4 - 4\sigma_0^2 \mathrm{Err}_\infty} \leq (\sigma_e^\infty)^2$, the only constraint is that $(\sigma_e^\infty)^4 - 4\sigma_0^2 \mathrm{Err}_\infty \geq 0$. The optimal δ will have a unique solution when $(\sigma_e^\infty)^4 = 4\sigma_0^2 \mathrm{Err}_\infty$, and the optimal value is $\delta^\dagger = \frac{(\sigma_e^\infty)^2}{2\sigma_0^2}$. □

This can be explained by Figure 4.2.1, when σ_e^∞ is set relatively large (Err_∞ will also be large), there are two possible δ's that satisfy (4.3.9). Eventually, the conclusions from Theorem 4.3.3 can be used to derive the result of Theorem 4.3.1. Recall that in the worst case analysis, based on (4.3.1), we have $\mathrm{Err}_\infty = M\left(\epsilon, \alpha^\dagger\right)(\sigma_e^\infty)^2$. In Theorem 4.3.3, the optimal δ is achieved when $(\sigma_e^\infty)^4 = 4\sigma_0^2 \mathrm{Err}_\infty = 4\sigma_0^2 M\left(\epsilon, \alpha^\dagger\right)(\sigma_e^\infty)^2$, which provides $(\sigma_e^\infty)^2 = 4\sigma_0^2 M\left(\epsilon, \alpha^\dagger\right)$. The optimal value is $\delta^\dagger = \frac{(\sigma_e^\infty)^2}{2\sigma_0^2} = \frac{4\sigma_0^2 M\left(\epsilon,\alpha^\dagger\right)}{2\sigma_0^2} = 2M\left(\epsilon, \alpha^\dagger\right)$ which coincides with the solution achieved in Theorem 4.3.1.

4.3.3 Non-Sparse Case (Gaussian)

The state evolution analysis for sparse signals is also valid for the non-sparse case by considering the BG prior with $\epsilon = 1$. In this case, (4.3.4) will degenerate to the variance of a standard Gaussian distribution which is a constant with value equal to 1. The estimated error (4.3.6) then has a closed form

$$\mathrm{Err}_t = R^t\left(\sigma_e^t\right)^2. \tag{4.3.10}$$

Substituting (4.3.10) into (4.2.5) and setting $\sigma_e^t = \sigma_e^{t+1} = \sigma_e^\infty$, leads to

$$(\sigma_e^\infty)^2 = \frac{(c\sigma_x^2 + \delta\sigma_0^2) + \sqrt{(c\sigma_x^2 + \delta\sigma_0^2)^2 + 4\sigma_x^2\delta\sigma_0^2}}{2},$$

where $c \coloneqq \frac{(1-\delta)}{\delta}$. We ignore the negative value due to the non-negative property of the error. The final estimation error at the fixed point will be

$$\mathrm{Err}_\infty = \delta\left((\sigma_e^\infty)^2 - \delta\sigma_0^2\right)$$

which is exactly the same as (4.2.4) (i.e. AMP achieves the optimal MMSE).

4.4 Analysis in Complex Domain

The analysis in the complex domain follows the same line as it in the real domain but we need to take care of the modifications.

For LF distribution: The Complex AMP (CAMP) algorithm for LF distribution has been analysed in [33] providing a new Onsager term. The LF distribution becomes $p_{|x|} = (1-\epsilon)\,\delta^f_{|x|=0} + \epsilon\delta^f_{|x|=+\infty}$ with the assumption that

the phase of x is isotropic and based on [33], the $\eta(\cdot)$ function will be,

$$\eta\left(\tilde{x}_i^t, \theta^t\right) \coloneqq \left(\tilde{x}_i^t - \frac{\theta^t\left(\tilde{x}_i^t\right)}{|\tilde{x}_i^t|}\right) \mathbb{I}_{\{|\tilde{x}_i^t|>\theta^t\}} \tag{4.4.1}$$

where $\mathbb{I}_{\{|\tilde{x}_i^t|>\theta^t\}}$ denotes the indicator function. The formula of $\mathrm{Err}_{C,t}$ will be the same as in real case but with a new $M_C(\epsilon, \alpha)$ function:

$$M_C(\epsilon, \alpha) \coloneqq \epsilon\left(1+\alpha^2\right) + (1-\epsilon)\left[\sqrt{2\pi}\phi\left(\sqrt{2}\alpha\right) - 2\alpha\sqrt{\pi}\Phi\left(-\sqrt{2}\alpha\right)\right]. \tag{4.4.2}$$

Compare (4.4.2) with (2.2.7), we can find the estimation error of non-zero components of signal are the same (first term). The difference between them comes from the de-noising for the zero components of signal (second term). For the complete derivation of new Onsager term and calculation of $\eta'\left(\tilde{x}_i^t, \theta^t\right)$, please refer to [33].

For BG distribution: We assume that the real part and imaginary part of a complex variable share the same mean and variance and they are uncorrelated. For example, let $x \sim \mathcal{CN}\left(\mu, \sigma_x^2\right)$, then we have $(x)^R, (x)^I \sim \mathcal{N}\left(\mu, \frac{\sigma_x^2}{2}\right)$. Under this assumption, we have

$$\begin{aligned} p_{CG}\left(x; \mu, \sigma_x^2\right) &= p_G\left((x)^R; \mu, \frac{\sigma_x^2}{2}\right) p_G\left((x)^I; \mu, \frac{\sigma_x^2}{2}\right) \\ &= \frac{1}{\pi\sigma_x^2}\exp\left(-\frac{|x-\mu_c|^2}{\sigma_x^2}\right), \end{aligned} \tag{4.4.3}$$

where $\mu_c = \mu + \sqrt{-1}\mu$ and the BG distribution in the complex domain becomes $p(x) = (1-\epsilon)\,\delta^f_{|x|=0} + \epsilon p_{CG}(x)$. For the estimate function $\eta(\cdot)$, we just replace the p_G probability in (4.3.5) with p_{CG} defined above. Now let $p^t_{\tilde{x},1} \coloneqq p_{CG}\left(\tilde{x}_i^t; 0, \left(\sigma_e^t\right)^2 + \sigma_x^2\right)$, $p^t_{\tilde{x},2} \coloneqq p_{CG}\left(\tilde{x}_i^t; 0, \left(\sigma_e^t\right)^2\right)$, $p^t_{\tilde{x},3} \coloneqq (1-\epsilon)\,p^t_{\tilde{x},2} + \epsilon p^t_{\tilde{x},1}$ and

$p_o^t \coloneqq -\frac{2}{\sigma_x^2+(\sigma_e^t)^2}p_{\tilde{x},3}^t + \frac{2(1-\epsilon)}{\sigma_w^2}p_{\tilde{x},2}^t + \frac{2\epsilon}{\sigma_x^2+(\sigma_e^t)^2}p_{\tilde{x},1}^t$, the four derivatives of $\eta(\cdot)$ can be calculated based on the following formulas:

$$\frac{\partial \eta^R\left(\tilde{x}_j^t\right)}{\partial\left(\tilde{x}_j^t\right)^R} = \frac{p_o^t}{\left(p_{\tilde{x},3}^t\right)^2}p_{\tilde{x},,1}^t \epsilon R\left(\left(\tilde{x}_j^t\right)^R\right)^2 + \frac{p_{\tilde{x},1}^t}{p_{\tilde{x},3}^t}\epsilon R \tag{4.4.4}$$

$$\frac{\partial \eta^R\left(\tilde{x}_j^t\right)}{\partial\left(\tilde{x}_j^t\right)^I} = \frac{\partial \eta^I\left(\tilde{x}_j^t\right)}{\partial\left(\tilde{x}_j^t\right)^R} = \frac{p_o^t}{\left(p_{\tilde{x},3}^t\right)^2}p_{\tilde{x},1}^t \epsilon R\left(\tilde{x}_j^t\right)^R\left(\tilde{x}_j^t\right)^I \tag{4.4.5}$$

$$\frac{\partial \eta^I\left(\tilde{x}_j^t\right)}{\partial\left(\tilde{x}_j^t\right)^I} = \frac{p_o^t}{\left(p_{\tilde{x},3}^t\right)^2}p_{\tilde{x},1}^t \epsilon R\left(\left(\tilde{x}_j^t\right)^I\right)^2 + \frac{p_{\tilde{x},1}^t}{p_{\tilde{x},3}^t}\epsilon R, \tag{4.4.6}$$

Finally, (4.3.6) will be replaced by

$$\mathrm{Err}_{C,t+1} = \left[\frac{R^t\epsilon}{1-R^t}\left(1-R^t I_C\left(R^t,\epsilon\right)\right)\right]\left(\sigma_e^t\right)^2, \tag{4.4.7}$$

$$I_C\left(R^t,\epsilon\right) = \int_{x^R}\int_{x^I}\frac{\phi_C(x)}{1+\frac{1-\epsilon}{\epsilon}\frac{1}{1-R^t}\exp\left(-\frac{R^t}{1-R^t}|x|^2\right)}|x|^2\,dx^I dx^R \tag{4.4.8}$$

where $\phi_C(x) = p_{CG}(x;0,1)$ is the standard complex normal distribution. (4.4.7) and (4.3.6) are exactly the same except the integration terms (see Section 4.6.1 for the detailed proof).

Fast calculation of $\mathrm{Err}_{C,t+1}$**:** The same as in the real domain, we can efficiently calculate Equation (4.4.7) by focusing on the real part of the signal only,

$$\mathrm{Err}_{C,t+1}^R \approx \frac{1}{n}\sum_{i=1}^n\left(\frac{\partial \eta^R\left(\tilde{x}_j^t\right)}{\partial\left(\tilde{x}_j^t\right)^R}\right)\frac{\left(\sigma_e^t\right)^2}{2} \tag{4.4.9}$$

$$\mathrm{Err}_{C,t+1} \approx 2\mathrm{Err}_{C,t}^R = \frac{1}{n}\sum_{i=1}^n\left(\frac{\partial \eta^R\left(\tilde{x}_j^t\right)}{\partial\left(\tilde{x}_j^t\right)^R}\right)\left(\sigma_e^t\right)^2 \tag{4.4.10}$$

which based on the assumption that the real part and imaginary part of the

complex random variable are i.i.d..

The optimal $\delta^\dagger$ can be achieved by using the same theorems in section 4.3, just replacing $M\left(\epsilon, \alpha^\dagger\right)$ and Err_{t+1} functions with $M_C\left(\epsilon, \alpha^\dagger\right)$ and $\text{Err}_{C,t+1}$, respectively.

4.5 Discussion and Numerical Justification

4.5.1 Discussion on the Optimal of $\delta^\dagger$

The optimal $\delta^\dagger$ for LF distribution can be directly achieved by (4.3.2), while for BG, $\delta^\dagger$ is achieved by numerically calculation. In order to find the common attribute of $\delta^\dagger$ and get an intuition into the values of $\delta^\dagger$ for different kinds of signals, we analyse the upper bounds for Gaussian, BG and LF distributions.

We first focus on the real domain. For the Gaussian case, based on (4.2.4), we are able to achieve $\delta^\dagger = \left(\sqrt{\sigma_x^4/\sigma_0^4 + 16\sigma_x^2/\sigma_0^2} - \sigma_x^2/\sigma_0^2\right)/4$, which is a monotonically decreasing function of $\sigma_0 \in (0, \infty]$, and $\delta^\dagger$ is upper bounded by 2. For the LF distribution, based on Theorem 4.3.1 and the fact that $M\left(\epsilon, \alpha^\dagger\right) \in (0, 1]$, the same upper bound applies. For the BG case, based on Theorem 4.3.3, $\delta^\dagger$ is an increasing function of Err_∞, which is upper bounded by the Gaussian case. Thus, $\delta^\dagger$ is also upper bounded by 2. The same results can be shown also in the complex domain (the detailed proof is given in Section 4.6).

4.5.2 Numerical Justification

For the simulation, we set $n = 1000$, $\sigma_0^2 = 0.01$ as constants. Each simulation point is the average of 100 independent trials. The simulation results provided in Fig. 4.5.1 and Fig. 4.5.2 show the relationship between the MSE and the

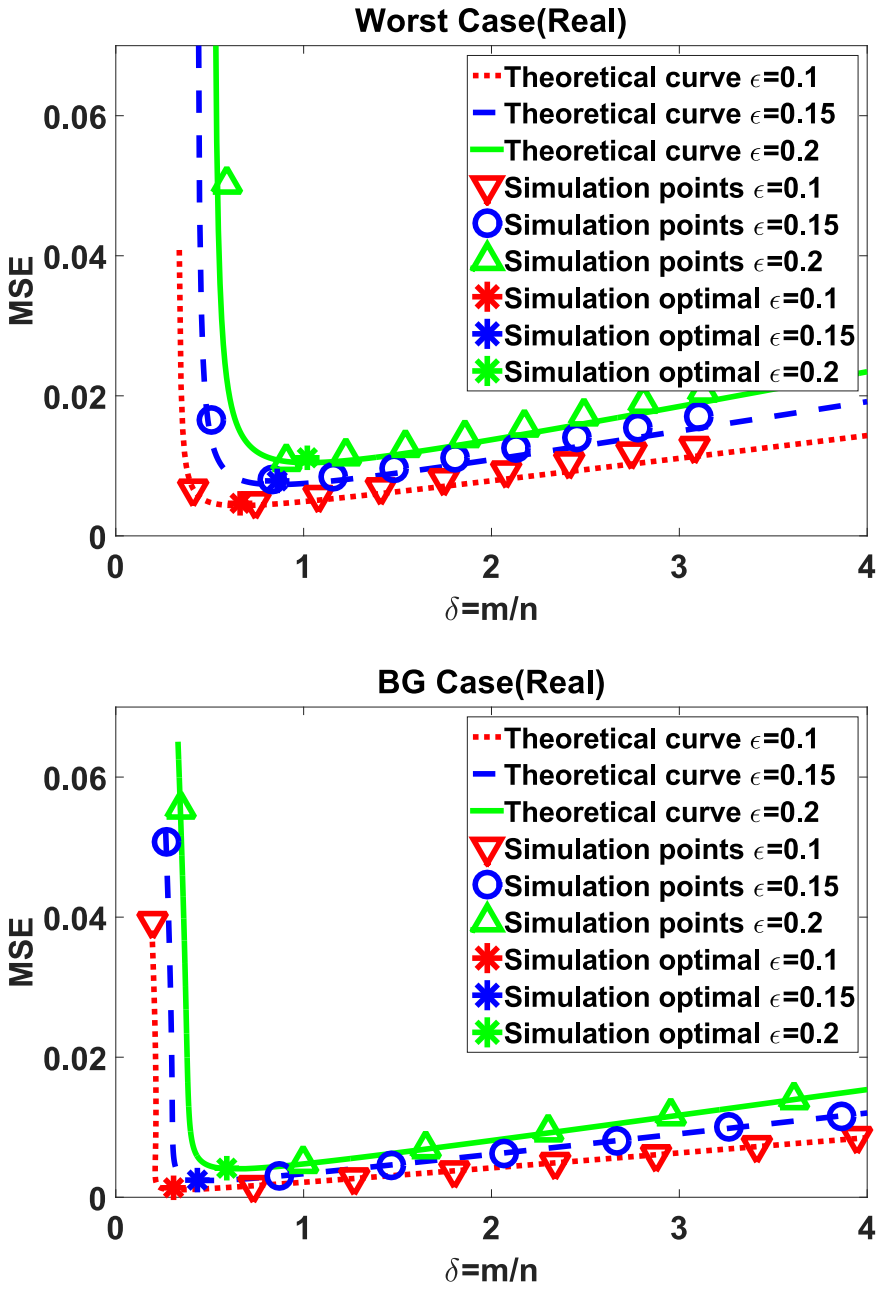


Figure 4.5.1: MSE (Err_∞) vs δ for real case. For the same sparsity and noise levels, the MMSE estimator provides better performances.

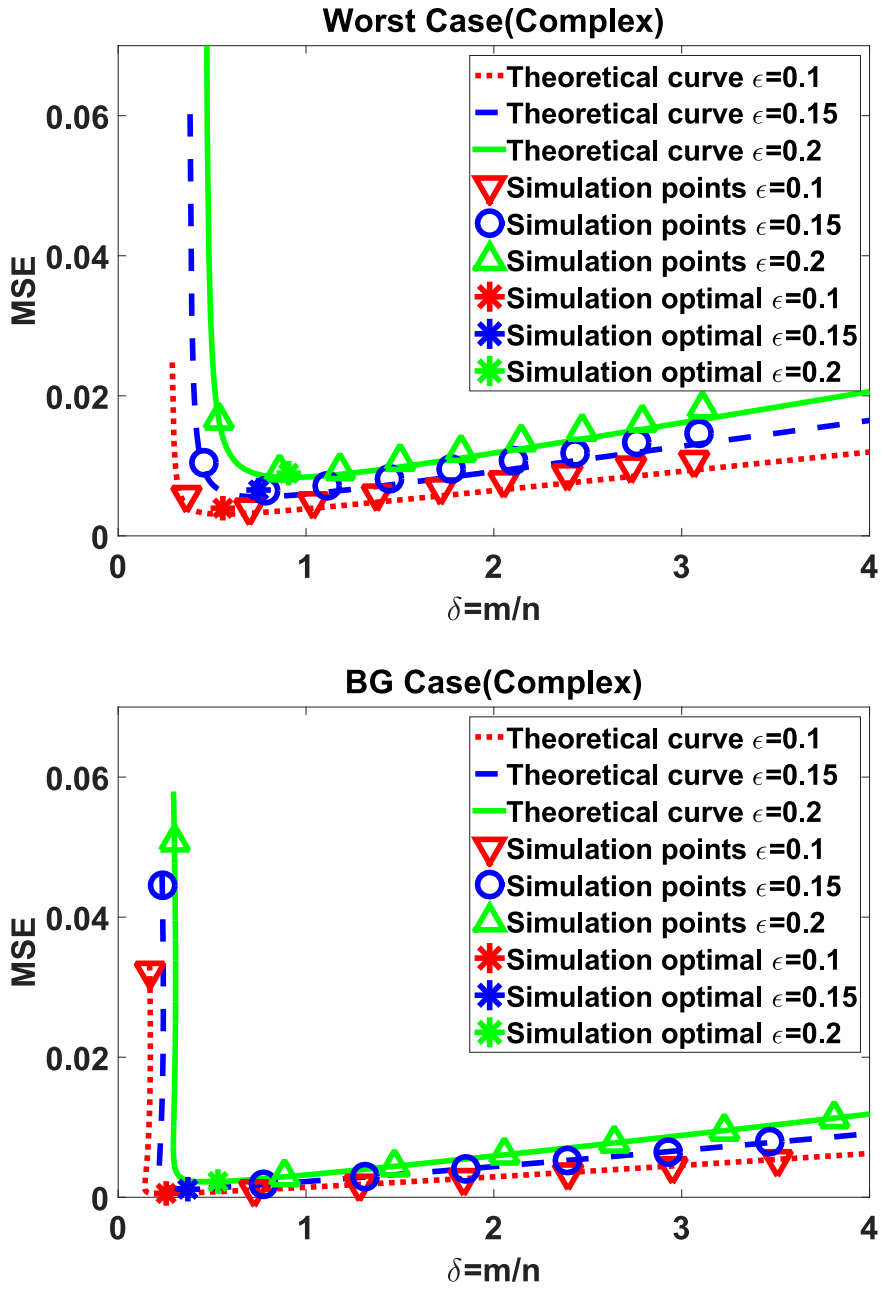


Figure 4.5.2: MSE (Err_∞) vs δ for complex case. For the same sparsity and noise levels, the MMSE estimator provides better performances.

measurement ratio δ for a given sparsity level ϵ. From the figure, one can observe that, when δ increases, initially, the MSE decreases dramatically until it reaches a minimum. After that, further increase in δ will increase the MSE. This phenomenon verifies our presumption that there exists an optimal $\delta^\dagger$ (or $m^\dagger$) for a limited energy transmission system. The overall performance of BG distribution is better than the one of LF distribution which coincides with our explanation at the beginning of Section 4.4. The numerical results of BG signals match the theoretical curves quite well but for the LF distribution, the numerical results are slightly larger than the theoretical curves. The main reason is that for the theoretical analysis in this case, we assume that the values of the non-zero coefficients are $\pm\infty$, but in simulations, these values can only be set as certain large numbers which results in a lower SNR compared with the one in the theoretical case. For both signal distributions, the trends in $\delta^\dagger$ for different ϵ values coincide with above optimal analysis.

The relationship between optimal $\delta^\dagger$ and the sparsity level ϵ for different types of distributions are listed in Fig. 4.5.3 and Fig. 4.5.4. For LF distribution, at the same sparsity level, the optimal δ in the complex case is smaller than the value in the real case. For BG distribution, at the same sparsity level and same noise base level, a similar phenomenon can be observed. In addition, if the sparsity level is relatively small (e.g. $\epsilon = 0.1$, depends on the current noise base level), when the noise base level increases, the optimal δ will move up which means we need to slightly increase the number of measurements; if the sparsity level is relatively large (e.g. $\epsilon = 0.35$, depends on the current noise base level), when the noise base level increases, the optimal δ may move down which means we may need to decrease the number of measurements.

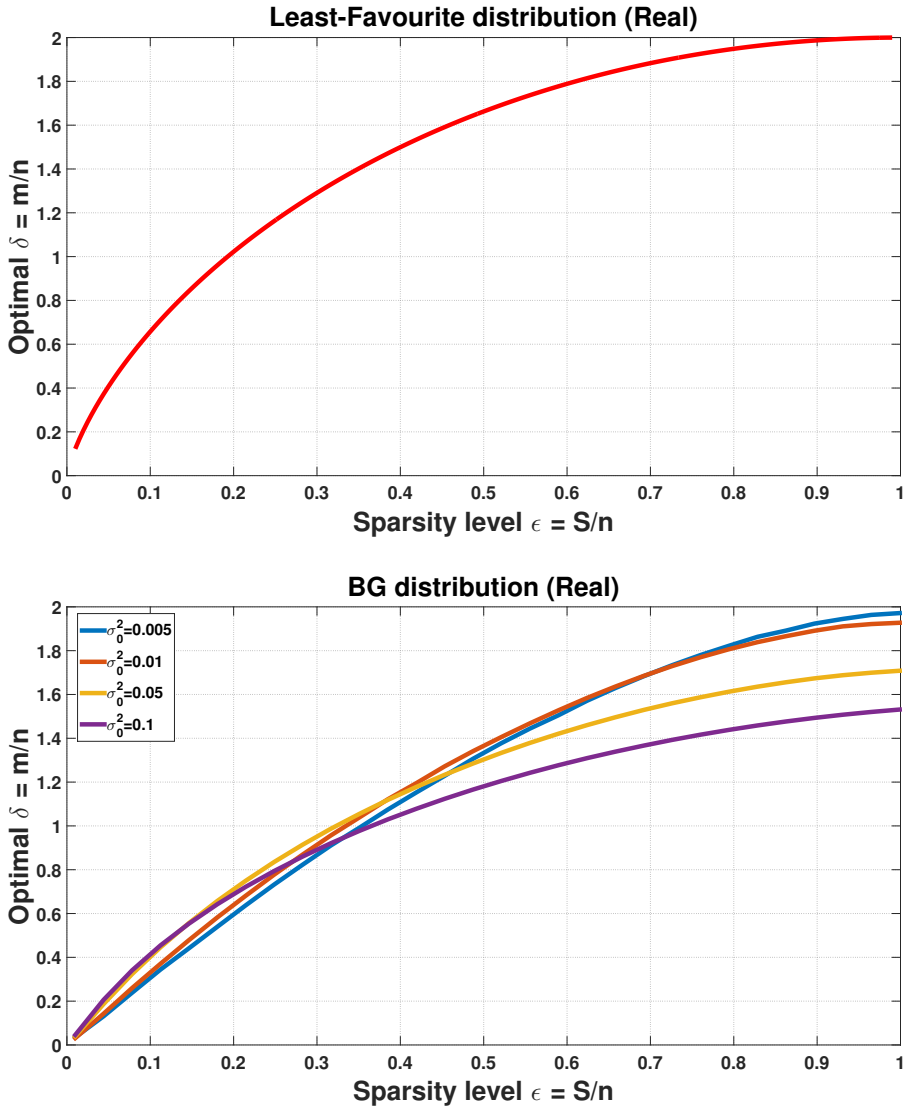


Figure 4.5.3: Optimal δ vs sparsity level for real case. For LF distribution, the curve is not related with noise; for BG distribution, the noise base level changes from 0.005 to 0.1. All curves are upper bounded by 2.

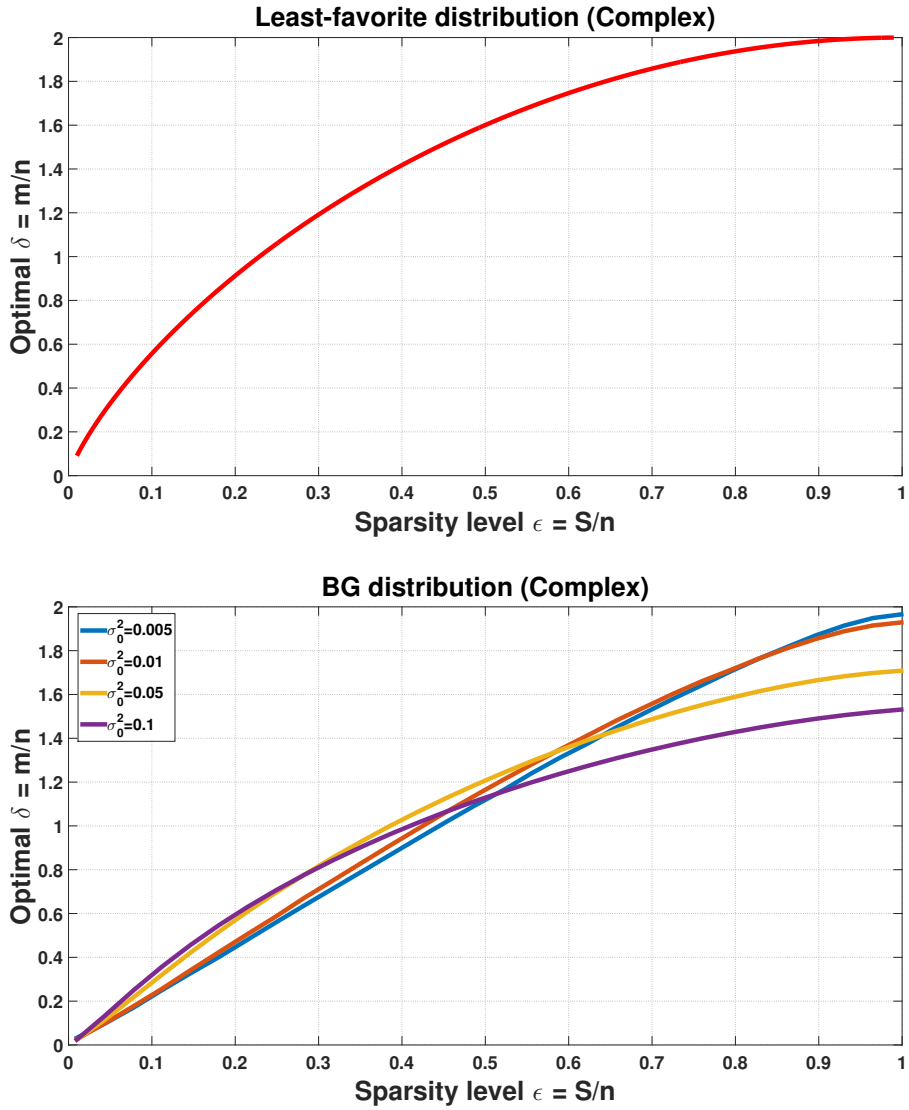


Figure 4.5.4: Optimal δ vs sparsity level for complex case. For LF distribution, the curve is not related with noise; for BG distribution, the noise base level changes from 0.005 to 0.1. All curves are upper bounded by 2.

4.6 Proof

4.6.1 Proof of $\eta(\cdot)$ and Err for Real and Complex Bernoulli-Gaussian Prior

For the real case, we directly apply the conclusion from Chapter 3. Based on (3.5.1) by setting $K = 1$, we achieve (4.3.5). Based on (3.6.23) by setting $K = 1$, we have

$$\text{Err} = \epsilon\sigma_x^2 - \epsilon R^2 \int \frac{p_G\left(\tilde{x};\, 0\,\sigma_x^2 + \sigma_e^2\right)}{\frac{(1-\epsilon)}{\epsilon}\sqrt{\frac{1}{1-R}}\exp\left(-\frac{(\tilde{x})^2}{2\sigma_e^2}R\right) + 1}\tilde{x}^2 d\tilde{x}. \tag{4.6.1}$$

Changing the variable by defining $\gamma \coloneqq \frac{\tilde{x}}{\sqrt{\sigma_e^2+\sigma_x^2}}$, then $d\tilde{x} = \sqrt{\sigma_e^2 + \sigma_x^2}d\gamma$. Substituting these into (4.6.1) will produce

$$\begin{aligned}
\text{Err} &= \epsilon\sigma_x^2 - \epsilon\sigma_x^2 R \int \frac{\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\gamma^2}{2}\right)}{\frac{(1-\epsilon)}{\epsilon}\sqrt{\frac{1}{1-R}}\exp\left(-\frac{\gamma^2\sigma_x^2}{2\sigma_e^2}\right) + 1}\gamma^2 d\gamma, \\
&= \epsilon\sigma_x^2\left(1 - RI\left(R,\epsilon\right)\right), \\
&= \frac{R\epsilon}{1-R}\left(1 - RI\left(R,\epsilon\right)\right)\sigma_e^2,
\end{aligned}$$

and (4.3.6) is proved.

For the complex case, we starts from (4.4.3). Following the same steps in section (3.6.8) , we have

$$\begin{aligned}
p_{CG}\left(x\right) &= \left(1-\epsilon\right)\delta_{x=0}^f + \epsilon p_{CG}\left(x; 0, \sigma_x^2\right), \\
p_{X,\tilde{x}}\left(x,\tilde{x}\right) &= \left(1-\epsilon\right)p_{CG}\left(\tilde{x}-x; 0, \sigma_e^2\right)\delta_{x=0}^f + \epsilon p_{CG}\left(x; R\tilde{x}, R\sigma_e^2\right)p_{CG}\left(\tilde{x}; 0, \sigma_e^2+\sigma_x^2\right) \\
p_C\left(\tilde{x}\right) &= \left(1-\epsilon\right)p_{CG}\left(\tilde{x}; 0, \sigma_e^2\right) + \epsilon p_{CG}\left(\tilde{x}; 0, \sigma_e^2+\sigma_x^2\right)
\end{aligned}$$

and

$$\begin{aligned}\hat{x} = \mathrm{E}\left[x|\tilde{x}\right] &= \int xp\left(x|\tilde{x}\right)dx, \\ &= \frac{p_{CG}\left(\tilde{x};0,\sigma_e^2+\sigma_x^2\right)}{p_C\left(\tilde{x}\right)}\epsilon R\tilde{x}.\end{aligned}$$

The MSE calculation is based on section (3.6.7) and in the complex case, it becomes

$$\mathrm{Err}_C = Ex^2 - E_{\tilde{x}}\left[\left|E_{x|\tilde{x}}\left[x|\tilde{x}\right]\right|^2\right].$$

where we still have $Ex^2 = \epsilon\sigma_x^2$. The second term provides

$$\begin{aligned}E_{\tilde{x}}\left[\left|E_{x|\tilde{x}}\left[x|\tilde{x}\right]\right|^2\right] &= \int\left|E_{x|\tilde{x}}\left[x|\tilde{x}\right]\right|^2 p_C\left(\tilde{x}\right)d\tilde{x} \\ &= \epsilon^2R^2\int\frac{p_{CG}^2\left(\tilde{x};0,\sigma_e^2+\sigma_x^2\right)}{p_C\left(\tilde{x}\right)}\left|\tilde{x}\right|^2 d\tilde{x} \\ &= \epsilon R^2\int_{\tilde{x}^R}\int_{\tilde{x}^I}\frac{p_{CG}\left(\tilde{x};0,\sigma_e^2+\sigma_x^2\right)}{\frac{(1-\epsilon)}{\epsilon}\frac{p_{CG}(\tilde{x};0,\sigma_w^2)}{p_{CG}(\tilde{x};0,\sigma_w^2+\sigma_x^2)}+1}\left|\tilde{x}\right|^2 d\tilde{x}^R d\tilde{x}^I \qquad (4.6.2)\end{aligned}$$

Recall that $\phi_c\left(x\right) = p_{CG}\left(x,0,1\right)$ which is the standard complex Gaussian distribution, by defining $x = \sigma_x y$ we have

$$\begin{aligned}p_{CG}\left(x;0,\sigma_x^2\right) &= \frac{1}{\sigma_x^2}\frac{1}{\pi}\exp\left(-\frac{\sigma_x^2\left|y\right|^2}{\sigma_x^2}\right) \\ &= \frac{1}{\sigma_x^2}\phi_c\left(y\right) \\ &= \frac{1}{\sigma_x^2}\phi_c\left(\frac{x}{\sigma_x}\right).\end{aligned}$$

This implies that (4.6.2) can be rewritten as

$$E_{\tilde{x}}\left[\left|E_{x|\tilde{x}}\left[x|\tilde{x}\right]\right|^2\right]=\epsilon R^2\int_{\tilde{x}^R}\int_{\tilde{x}^I}\frac{\frac{1}{\sigma_e^2+\sigma_x^2}\phi_c\left(\frac{\tilde{x}}{\sqrt{\sigma_e^2+\sigma_x^2}}\right)}{\frac{(1-\epsilon)}{\epsilon}\frac{p_{CG}(\tilde{x};0,\sigma_e^2)}{p_{CG}(\tilde{x};0,\sigma_e^2+\sigma_x^2)}+1}\left|\tilde{x}\right|^2 d\tilde{x}^R d\tilde{x}^I.$$

Define $\gamma \coloneqq \frac{\tilde{x}}{\sqrt{\sigma_w^2+\sigma_x^2}}$, we have $d\tilde{x}^R = \sqrt{\sigma_e^2+\sigma_x^2}d\gamma^R$, $d\tilde{x}^I = \sqrt{\sigma_e^2+\sigma_x^2}d\gamma^I$, $\left|\tilde{x}\right|^2=(\sigma_e^2+\sigma_x^2)\left|\gamma\right|^2$ and substituting into $E_{\tilde{x}}\left[\left|E_{x|\tilde{x}}\left[x|\tilde{x}\right]\right|^2\right]$, the MSE will be

$$\begin{aligned}\mathrm{Err}_C&=\epsilon\sigma_x^2-\sigma_x^2\epsilon R\int_{\gamma^R}\int_{\gamma^I}\frac{\phi_c(\gamma)}{\frac{(1-\epsilon)}{\epsilon}\frac{1}{1-R}\exp\left(-\frac{R}{1-R}\left|\gamma\right|^2\right)+1}\left|\gamma\right|^2 d\gamma^R d\gamma^I,\\&=\epsilon\sigma_x^2\left(1-RI_C\left(R,\epsilon\right)\right),\\&=\frac{R\epsilon}{1-R}\left(1-RI_C\left(R,\epsilon\right)\right)\sigma_e^2,\end{aligned}$$

(4.4.7) is proved.

4.6.2 Boundary Analysis for Gaussian Distribution

We have the MSE for Gaussian estimator with $\sigma_w^2=\delta\sigma_0^2$,

$$\lim_{n\to\infty}\frac{1}{n}\left\|\boldsymbol{x}-\hat{\boldsymbol{x}}\right\|^2=\frac{\delta}{2}\left[\left(-\delta\sigma_0^2+c\sigma_x^2\right)+\sqrt{\left(\delta\sigma_0^2+c\sigma_x^2\right)^2+4\delta\sigma_0^2\sigma_x^2}\right].$$

where $c=\frac{1-\delta}{\delta}$. Define

$$\begin{aligned}f(\delta)&=\left(-\delta^2\sigma_0^2+(1-\delta)\sigma_x^2\right)+\sqrt{\left(\delta^2\sigma_0^2+(1-\delta)\sigma_x^2\right)^2+4\delta^3\sigma_0^2\sigma_x^2}\\g(\delta)=\frac{f(\delta)}{\sigma_0^2}&=\left(-\delta^2+(1-\delta)\frac{\sigma_x^2}{\sigma_0^2}\right)+\sqrt{\left(\delta^2+(1-\delta)\frac{\sigma_x^2}{\sigma_0^2}\right)^2+4\delta^3\frac{\sigma_x^2}{\sigma_0^2}}\\&=\left(-\delta^2+(1-\delta)C_1\right)+\sqrt{\left(\delta^2+(1-\delta)C_1\right)^2+4\delta^3C_1}\end{aligned}$$

where $C_1 = \frac{\sigma_x^2}{\sigma_0^2}$. Let $\frac{\partial g(\delta)}{\partial \delta} = 0$ which provides

$$2C_1^2\delta^3 + C_1^3\delta^2 - 2C_1^3\delta = 0$$

as C_1 is a positive value, the only possible solution is the positive roots of

$$2\delta^2 + C_1\delta - 2C_1 = 0$$

which gives

$$\delta^\dagger = \frac{\sqrt{C_1^2 + 16C_1} - C_1}{4} =: L\left(C_1\right).$$

We have

$$\frac{\partial L(C_1)}{\partial C_1} = \frac{(C_1 + 8)}{(C_1^2 + 16C_1)^{\frac{1}{2}}} - 1 > 0$$

then for $C_1 > 0$, $L(C_1)$ is a monotonic increasing function and

$$\delta^\dagger = \frac{\sqrt{C_1^2 + 16C_1} - C_1}{4} < \frac{\sqrt{C_1^2 + 16C_1 + 64} - C_1}{4} = \frac{(C_1 + 8) - C_1}{4} = 2$$

4.6.3 Boundary Analysis for Least-Favourite Distribution

Firstly, we consider (2.2.7) for the real case analysis, which can be rewritten as

$$M\left(\epsilon, \alpha\right) = \epsilon T_1 + T_2,$$

where

$$T_1 = \left(1+\alpha^2\right) + 2\alpha\phi\left(\alpha\right) - 2\left(1+\alpha^2\right)\Phi\left(-\alpha\right),$$
$$T_2 = 2\left(1+\alpha^2\right)\Phi\left(-\alpha\right) - 2\alpha\phi\left(\alpha\right).$$

For any $\alpha \geq 0$, we have $\Phi\left(-\alpha\right) \leq \frac{1}{2}$, thus

$$\begin{aligned} T_1 &= \left(1+\alpha^2\right) + 2\alpha\phi\left(\alpha\right) - 2\left(1+\alpha^2\right)\Phi\left(-\alpha\right) \\ &\geq \left(1+\alpha^2\right) - 2\left(1+\alpha^2\right)\Phi\left(-\alpha\right) \\ &= \left(1+\alpha^2\right)\left(1-2\Phi\left(-\alpha\right)\right) \\ &\geq 0 \end{aligned}$$

which means for any fixed $\alpha \geq 0$, $M\left(\epsilon,\alpha\right)$ is a monotonic increasing function. Thus, for any $0 < \epsilon_1 < \epsilon_2 \leq 1$, we have $M\left(\epsilon_1,\alpha\right) < M\left(\epsilon_2,\alpha\right) \leq M\left(1,\alpha\right)$. Where

$$M\left(1,\alpha\right) = 1+\alpha^2$$

and based on

$$\alpha^\dagger = \arg\min_{\alpha\geq 0} M\left(1,\alpha\right)$$

we have $\alpha^\dagger = 0$ and $M\left(1,\alpha^\dagger\right) = 1$. Finally, we found that

$$0 < M\left(\epsilon,\alpha^\dagger\right) \leq 1$$

the lower bound is by definition. Based on Theorem 4.3.1, $\delta^\dagger$ is upper bounded by 2.

For the complex case, we also can rewritten (4.4.2) as $M_C(\epsilon, \alpha) = \epsilon T_1 + T_2$ where

$$T_1 = 1 + \alpha^2 - \sqrt{2\pi}\phi\left(\sqrt{2}\alpha\right) + 2\alpha\sqrt{\pi}\Phi\left(-\sqrt{2}\alpha\right),$$
$$T_2 = \sqrt{2\pi}\phi\left(\sqrt{2}\alpha\right) - 2\alpha\sqrt{\pi}\Phi\left(-\sqrt{2}\alpha\right).$$

and for any given $\alpha \geq 0$, we have

$$\begin{aligned} T_1 &= 1 + \alpha^2 - \sqrt{2\pi}\phi\left(\sqrt{2}\alpha\right) + 2\alpha\sqrt{\pi}\Phi\left(-\sqrt{2}\alpha\right) \\ &\geq 1 + \alpha^2 - \sqrt{2\pi}\phi\left(\sqrt{2}\alpha\right) \\ &= 1 + \alpha^2 - \exp\left(-\alpha^2\right) \\ &\geq 0 \end{aligned}$$

By following the same analysis in the real case, a same result can be achieved.

4.6.4 Boundary Analysis for Bernoulli-Gaussian Distribution

Firstly, we consider the real case analysis. Based on Theorem 4.3.3, we have

$$\delta = \frac{\left(\sigma_e^{\infty}\right)^2 \pm \sqrt{\left(\sigma_e^{\infty}\right)^4 - 4\sigma_0^2 \mathrm{Err}_{\infty}(\epsilon)}}{2\sigma_0^2}$$

and

$$\delta^{\dagger} = \frac{\left(\sigma_e^{\infty}\right)^2}{2\sigma_0^2}$$

where $\left(\sigma_e^{\infty}\right)^4 = 4\sigma_0^2 \mathrm{Err}_{\infty}(\epsilon)$ which means $\delta^{\dagger}$ will increase with the increasing of $\mathrm{Err}_{\infty}(\epsilon)$(here we use $\mathrm{Err}_{\infty}(\epsilon)$ instead of Err_{∞} to highlight that $\mathrm{Err}_{\infty}(\epsilon)$ is

a function of ϵ). Recall (4.3.6) which is

$$\begin{aligned}\mathrm{Err}(\epsilon) &\coloneqq \left[\frac{R\epsilon}{1-R}\left(1-RI(R,\epsilon)\right)\right]\sigma_e^2 \\ &= \sigma_x^2\epsilon - \sigma_x^2 RI(R,\epsilon)\,\epsilon\end{aligned}$$

where

$$I(R,\epsilon) \coloneqq \int \frac{\phi(x)}{1+\frac{1-\epsilon}{\epsilon}\frac{1}{\sqrt{1-R}}\exp\left(-\frac{R}{1-R}\frac{x^2}{2}\right)}x^2dx.$$

Then for any given $R=\frac{\sigma_x^2}{\sigma_x^2+\sigma_e^2}>0$, let $0<\epsilon_1<\epsilon_2\leq 1$, we have

$$0<I(R,\epsilon_1)<I(R,\epsilon_2)\leq 1.$$

Define

$$\begin{aligned}f_1(\epsilon) &\coloneqq \sigma_x^2\epsilon, \\ f_2(\epsilon) &\coloneqq \sigma_x^2 RI(R,\epsilon)\,\epsilon = \frac{\sigma_x^4}{\sigma_x^2+\sigma_e^2}I(R,\epsilon)\,\epsilon, \\ f_3(\epsilon) &\coloneqq \frac{\sigma_x^4}{\sigma_x^2+\sigma_e^2}\epsilon,\end{aligned}$$

and

$$\begin{aligned}f_{13}(\epsilon) &\coloneqq f_1(\epsilon) - f_3(\epsilon), \\ f_{32}(\epsilon) &\coloneqq f_3(\epsilon) - f_2(\epsilon).\end{aligned}$$

Because we have $f_1(\epsilon)/f_3(\epsilon)\geq 1$ and $f_3(\epsilon)/f_2(\epsilon)\geq 1$ for any $0<\epsilon\leq 1$, which means $f_{13}(\epsilon)$ and $f_{32}(\epsilon)$ are both monotonic increasing functions. In

addition

$$\mathrm{Err}\left(\epsilon\right) = f_{13}\left(\epsilon\right) + f_{32}\left(\epsilon\right) = f_1(\epsilon) - f_2\left(\epsilon\right),$$

which is also a monotonic increasing function. Thus for $0 < \epsilon_1 < \epsilon_2 \leq 1$, we have

$$0 < \mathrm{Err}(\epsilon_1) \leq \mathrm{Err}(\epsilon_2) \leq \mathrm{Err}(1)$$

and for $\epsilon = 1$, the BG distribution degenerates to the Gaussian signal and $\delta^\dagger$ is also upper bounded by 2.

For the complex case, we focus on (4.4.8) which has the same behaviour of $I\left(R, \epsilon\right)$, the analysis should be exactly the same as above.

Chapter 5

Improved AMP for Non I.I.D. Gaussian Random Matrices

This chapter studies the sparse recovery problem of AMP algorithm with non i.i.d. Gaussian random matrices. AMP enjoys low computational complexity and good performance guarantees. However, the algorithm and performance analysis rely heavily on the assumption that the measurement matrix is a standard Gaussian random matrix with i.i.d. components. The main contribution of this chapter is an improved AMP (IAMP) algorithm that works better for non i.i.d. Gaussian random matrices. The algorithm is equivalent to AMP for standard Gaussian random matrices but provides better recovery when the correlations between elements of the measurement matrix deviate from those of the standard Gaussian random matrices. The derivation is based on a modification of the message passing mechanism that removes the conditional independence assumption. Examples are provided to demonstrate the performance improvement of IAMP where both a particularly designed matrix and a matrix from real applications are used.

5.1 Introduction

The AMP algorithm has received wide attention due to its two nice properties: low computational complexity and good performance guarantees. It only involves matrix-vector products and scalar operations and therefore the complexity is $O(n^2)$. At the same time, if the measurement matrix is a standard Gaussian random matrix, it has been rigorously proved that AMP achieves the same phase transition curve as ℓ_1-minimisation does. Furthermore, AMP allows complicated statistical models for both the unknown sparse signal and the noise [1, 60, 30, 29, 90]. It has been proved in [60] that any signal model can be applied as long as the corresponding denoiser is Lipschitz continuous. This extends the applicability of AMP.

One drawback of AMP is that both the algorithm and the performance analysis rely heavily on the standard Gaussian random matrix. It has been numerically observed that the performance of AMP may severely deteriorate if the measurement matrix is significantly different from the standard Gaussian random matrix. A particularly designed example is given in Section 5.4 to highlight this phenomenon. This drawback limits the applicability of AMP. There have been methods proposed to address this issue, including damped GAMP [91] which linearly combines the results from two adjacent iterations, SwAMP [31] which updates components in $\boldsymbol{x}$ sequentially, ADMM-GAMP [92] which considers the inference problem of generalized linear models (GLM) as a large-system-limit approximation of the bethe free energy (LSL-BFE) minimization problem and uses alternating direction method of multipliers (ADMM) method to solve it, orthogonal AMP [93] which is based on de-correlated linear estimation and divergence-free non-linear estimation, and vector AMP [94] which

is derived using an approximation of belief propagation on a factor graph with vector-valued variable nodes.

The major contribution of this chapter is an improved AMP (IAMP) algorithm that works better for non i.i.d. Gaussian random matrices. The derivation of AMP is based on a factor graph representation of the system and Gaussian approximations of the passed messages on the factor graph. We observe that the conditional independence assumption used in the message computation is not valid any more when the elements of the Gaussian random matrix are not i.i.d.. It turns out that the correlation profile of the elements of the measurement matrix needs to be taken into consideration. Based on this observation, a new message passing mechanism is derived where all messages are computed at the variable nodes. This is quite different form previous approaches in [91, 31, 92]. The developed IAMP algorithm reduces to AMP when the measurement matrix is standard Gaussian; at the same time, substantial performance improvements of IAMP are demonstrated for non i.i.d. measurement matrices. It is noteworthy that IAMP involves extra computations. However the extra computations can be made offline so that the 'operational' complexity of IAMP is in the same order as that of AMP.

5.2 Message Passing of Approximate Message Passing

In [60], it has been shown that AMP can achieve the same phase transition curve as the famous LASSO/BPDN problem

$$\hat{\boldsymbol{x}} = \arg \min_{\boldsymbol{x}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 \tag{5.2.1}$$

for an appropriately chosen constant $\lambda \in \mathbb{R}^+$.

AMP is based on the well known belief propagation mechanism. Describe the probability model of the system using the factor graph in Fig. 5.2.1, where a variable node $i \in [n]$ contains information of x_i and a factor node $a \in [m]$ specifies the conditional probability $p(y_a|\boldsymbol{x})$. The message from a factor node a to a variable node i, denoted by $m_{a\to i}(x_i)$, can be considered as

$$m_{a\to i}(x_i) \coloneqq p(x_i|y_a) \propto \int p(y_a|x_i, \boldsymbol{x}_{\sim i})\, p(\boldsymbol{x}_{\sim i})\, d\boldsymbol{x}_{\sim i} \tag{5.2.2}$$

$$= \int p(y_a|x_i, z_{a\to i})\, p(z_{a\to i})\, dz_{a\to i}, \tag{5.2.3}$$

where $\boldsymbol{x}_{\sim i}$ denotes all the components in $\boldsymbol{x}$ except x_i, and $z_{a\to i} \coloneqq \sum_{j\neq i} A_{aj}x_j$. Note that generally speaking, $p(z_{a\to i})$ is complicated and it is computationally expensive to compute the integral involved in $m_{a\to i}(x_i)$. However, when $\boldsymbol{A}$ is a standard Gaussian random matrix, $z_{a\to i}$ is a *Gaussian* random variable [60, 95]. The message $m_{a\to i}(x_i)$ can be easily obtained and is also Gaussian. Now consider the message from a variable node i to a factor node a. Let $\hat{x}_{i\to a} \coloneqq \arg\max_{x_i} p(x_i|\boldsymbol{y}_{\sim a})$, where

$$p(x_i|\boldsymbol{y}_{\sim a}) \overset{(a)}{\propto} p(x_i) \prod_{b\neq a} p(y_b|x_i) \propto p(x_i) \prod_{b\neq a} m_{b\to i}(x_i), \tag{5.2.4}$$

where the relation (a) is based on the *conditional independence* assumption. When each $m_{b\to i}(x_i)$ is in Gaussian form, the computation of $p(x_i|\boldsymbol{y}_{\sim a})$ is highly simplified. In summary, the Gaussian approximation and conditional independence assumption are the two key elements in the derivation.

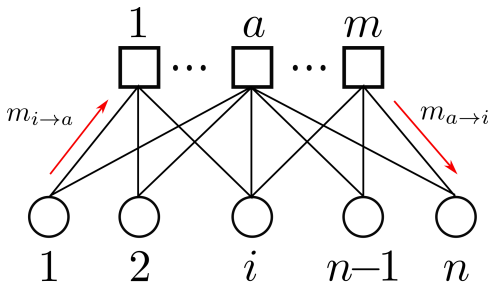


Figure 5.2.1: Factor graph and message passing: Squares represent factor nodes and circles represent variable nodes.

5.3 Improved Approximate Message Passing (IAMP)

5.3.1 Modification of Message Passing

The main difference between AMP and IAMP is the message passing mechanism to handle general measurement matrix $\boldsymbol{A}$. When the matrix $\boldsymbol{A}$ is sufficiently dense, $z_{a\to i}$ can be approximated by a Gaussian random variable so the Gaussian assumption for AMP is still valid. However, when the elements of $\boldsymbol{A}$ are highly correlated, the independence assumption (among $m_{b\to i}$'s, $b \in\sim a$) is not true any more and neither is (5.2.4). To address this issue, a new message passing mechanism has to be designed. In particular, due to the dependence between $m_{b\to i}$'s, the computation at the factor nodes becomes unnecessary. We focus on the message at the variable node

$$m_{\boldsymbol{y}\to i}(x_i) = p(x_i|\boldsymbol{y}) \propto \int p(\boldsymbol{y}|x_i, \boldsymbol{x}_{\sim i})\, p(\boldsymbol{x}_{\sim i})\, d\boldsymbol{x}_{\sim i}, \tag{5.3.1}$$

where we stick to the common assumption that $p(\boldsymbol{x}_{\sim i}) = \prod_{j\neq i} p(x_j)$. With the assumption that the measurement noise $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \sigma_w^2 \boldsymbol{I})$, the following Lemma suggests that $p(x_i|\boldsymbol{y})$ can be approximated by a simple Gaussian pdf.

Lemma 5.3.1. *Let $\boldsymbol{A}_i$ be the i^{th} column of $\boldsymbol{A}$ and $\boldsymbol{A}_i^T$ is the transpose of $\boldsymbol{A}_i$. Define $\tilde{y}_i := \boldsymbol{A}_i^T \boldsymbol{y}$, $z_i := \boldsymbol{A}_i^T \left(\sum_{j \neq i} \boldsymbol{A}_j x_j\right)$, $\mathrm{E}[x_j] = 0$, $\sigma_{x_j}^2 := \mathrm{E}\left[x_j^2\right]$, and*

$$\sigma_{z_i}^2 := \mathrm{var}(z_i) = \sum_{j \neq i} \left(\boldsymbol{A}_i^T \boldsymbol{A}_j\right)^2 \sigma_{x_j}^2 \tag{5.3.2}$$

Assume that $\|\boldsymbol{A}_i\|_2 = 1$, $\forall i \in [n]$, and $\boldsymbol{Ax}$ is jointly Gaussian. Then $p(x_i|\boldsymbol{y})$ can be approximated by $\mathcal{N}\left(\tilde{y}_i, \sigma_w^2 + \sigma_{z_i}^2\right)$.

Proof. See Section 5.6.1. □

5.3.2 Algorithm Description

At the variable nodes, the operation of IAMP is the same as that of AMP: each signal component is denoised individually from its noisy observation

$$\tilde{y}_i = x_i + \tilde{w}_i, \tag{5.3.5}$$

where $\tilde{w}_i$ is additive Gaussian noise with distribution $\mathcal{N}\left(0, \sigma_{\tilde{w}_i}^2\right)$. Based on Lemma 5.3.1, $\sigma_{\tilde{w}_i}^2 = \sigma_w^2 + \sigma_{z_i}^2$. To make the notation more intuitive, we also denote $\sigma_{\tilde{w}_i}^2$ by $\sigma_{\mathrm{in},i}^2$. Now consider the popular denoiser of the form [60]

$$\hat{x}_i = \eta(\tilde{y}_i; \theta_i) = \begin{cases} \tilde{y}_i - \theta_i & \tilde{y}_i > \theta_i \\ 0 & -\theta_i \leq \tilde{x}_i, \leq \theta_i \\ \tilde{y}_i + \theta_i & \tilde{y}_i < -\theta_i. \end{cases} \tag{5.3.6}$$

where θ_i is the corresponding threshold. In this chapter, (5.3.6) is different from (2.1.3), as we allow different threshold values θ_i's for different $\tilde{y}_i$'s. Define the mean squared error of this denoiser by $\sigma_{\mathrm{out},i}^2 := \mathrm{E}\left[(\hat{x}_i - x_i)^2\right]$. Consider

Algorithm 5.1 Pseudo code of IAMP algorithm

Import: $\boldsymbol{y}$: the observation vector. $\boldsymbol{A} = [\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3, ..., \boldsymbol{A}_n]$: the measurement matrix. σ_w^2: the noise variance. ϵ: the nonzero probability (defined as the ratio between the number of nonzero elements in $\boldsymbol{x}$ and n the dimension of $\boldsymbol{x}$).

Output: $\hat{\boldsymbol{x}}$: the estimated signal.

Initialization: Let $\boldsymbol{r}^0 = \boldsymbol{y}$, $\boldsymbol{x}^0 = 0$ and $t = 0$. Set $\sigma_{\text{out},i}^2 = \sigma_x^2 = \left(\|\boldsymbol{y}\|_2^2 - m\sigma_w^2\right) / \|\boldsymbol{A}\|_F^2$.

Iteration: In the t-th iteration, do

1. Based on (5.3.2), compute

$$\sigma_{\text{in},i}^2 = \sum_{j \neq i} \left(\boldsymbol{A}_i^T \boldsymbol{A}_j\right)^2 \sigma_{\text{out},j}^2 + \sigma_w^2, \ \forall i \in [n]\,. \tag{5.3.3}$$

2. Let $\tilde{y}_i^t = x_i^t + \sum_a A_{ai} r_a^t$. Update the estimated signal

$$x_i^{t+1} = \eta\left(\tilde{y}_i^t; \theta_i^t\right), \ \forall i \in [n]\,,$$

where the denoiser $\eta(\cdot)$ and the threshold θ_i^t are defined in (5.3.6) and (5.3.7) respectively.

3. Update the "residual" signal $\boldsymbol{r}^{t+1}$ by

$$r_a^{t+1} = y_a - \sum_{i=1}^n A_{ai} x_i^{t+1} + \sum_{i=1}^n A_{ai}^2 \eta'\left(\tilde{y}_i^t; \theta_i^t\right) r_a^t, \ \forall a \in [m]\,. \tag{5.3.4}$$

4. Compute $\sigma_{\text{out},i}^2$ via Equation (5.3.9).

5. [Optional] Adjust the "output" noise variance.
Let $\bar{\sigma}_r^2 = \frac{1}{m} \sum_{i=1}^n \|\boldsymbol{A}_i\|_2^2 \sigma_{\text{out},i}^2$. Set $\sigma_{\text{out},i}^2 = c\sigma_{\text{out},i}^2$, where

$$c = \left(\frac{1}{m}\left\|\boldsymbol{r}^{t+1}\right\|_2^2 - \sigma_w^2\right)_+ / \bar{\sigma}_r^2.$$

6. $t = t + 1$.

7. Go back to step 1 unless the stopping criteria are satisfied.

the worst case analysis as in Section 2.1, the optimal threshold θ_i (to minimise $\sigma^2_{\text{out},i}$) and the corresponding mean squared error $\sigma^2_{\text{out,i}}$ are given by

$$\theta_i = \alpha^\dagger \sigma_{\text{in},i}, \tag{5.3.7}$$

$$\alpha^\dagger := \arg\min_{\alpha} M(\epsilon, \alpha) \tag{5.3.8}$$

$$\sigma^2_{\text{out},i} = M\left(\epsilon, \alpha^\dagger\right) \sigma^2_{\text{in},i}. \tag{5.3.9}$$

where $M(\epsilon, \alpha)$ is given by (2.2.7).

With above notations, the IAMP algorithm is detailed in Algorithm 5.1. In the initialisation step, an estimation of the variance of σ_x^2 will be needed. From the model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$, it is approximately true that $\|\boldsymbol{A}\boldsymbol{x}\|_2^2 = \|\boldsymbol{y}\|_2^2 - m\sigma_w^2$. On the other hand, assume that x_i's are independent and $\sigma^2_{x_i} = \sigma^2_{x_j} = \sigma_x^2$, $\forall i \neq j$. Then $\|\boldsymbol{A}\boldsymbol{x}\|_2^2 \approx \sum_i \sigma^2_{x_i} \|\boldsymbol{A}_i\|_2^2 = \sigma_x^2 \|\boldsymbol{A}\|_F^2$. As a result, one can set $\sigma_x^2 = \left(\|\boldsymbol{y}\|_2^2 - m\sigma_w^2\right) / \|\boldsymbol{A}\|_F^2$.

In the IAMP algorithm, $\frac{1}{m}\|\boldsymbol{r}^{t+1}\|_2^2$ measures the uncertainty that still exists after current estimation which contains two parts, one part comes from the current estimation of x_i^{t+1}'s and another comes from the measurement noise. $\left(\frac{1}{m}\|\boldsymbol{r}^{t+1}\|_2^2 - \sigma_w^2\right)_+$ calculates the practical uncertainty from the current estimation and $\bar{\sigma}_r^2$ is the corresponding theoretical value. c (in step 5) can be treated as the practical-theoretical ratio and by the operation of $\sigma^2_{\text{out},i} = c\sigma^2_{\text{out},i}$, we transfer the theoretical value of $\sigma^2_{\text{out},i}$ into the 'practical' value. It is not guaranteed that with the optional step, the performance of IAMP algorithm will always be improved as it will affect each threshold value θ_i^{t+1} at the next iteration.

The major differences between AMP and IAMP are as follows. Assume that $\sigma^2_{\text{in},j} = \sigma^2_{\text{in},k}$ and $\sigma^2_{\text{out},j} = \sigma^2_{\text{out},k}$, $j \neq k$, and therefore $\sigma^2_{\text{in},i}$ and $\sigma^2_{\text{out},i}$

are replaced by σ_{in}^2 and σ_{out}^2 respectively. In AMP, Equation (5.3.3) becomes $\sigma_{\text{in}}^2 = \frac{1}{\delta}\sigma_{\text{out}}^2 + \sigma_w^2$ with $\delta := \frac{m}{n}$. The implementation of the denoising function η in (2.1.1) depends on this information. The second difference is that the last term in (5.3.4) becomes $\frac{1}{m}\sum_{i=1}^{n}\eta'\left(\tilde{y}_i^t;\theta_i^t\right)r_a^t = \frac{1}{\delta}\left\langle\eta'\left(\tilde{y}_i^t;\theta_i^t\right)\right\rangle r_a^t$.

The 'operational' complexity of IAMP is the same as that of AMP. The most computationally intensive step is the evaluation of (5.3.3), of which the complexity is $O\left(n^3\right)$. However, $\boldsymbol{A}_i^T\boldsymbol{A}_j$, $\forall i \neq j$, can be computed off-line. All other steps only involve at most $O\left(n^2\right)$ computations.

5.4 Performance Discussions

In this section, we will first show that if the measurement matrix $\boldsymbol{A}$ is a standard Gaussian random matrix, then IAMP reduces to AMP. Next, we construct a Gaussian random matrix such that the marginal distribution of each entry is still $\mathcal{N}\left(0,\frac{1}{m}\right)$ but the entries are dependent. For this scenario, we show the significant performance improvement of IAMP. Finally, we demonstrated the improvement of IAMP using synthetic data of a real application — radar imaging.

5.4.1 The Standard Gaussian Random Matrix

In this subsection, we consider the behaviour of IAMP for standard Gaussian random matrices, i.e., the entries are independently generated from $\mathcal{N}\left(0,\frac{1}{m}\right)$. Under this assumption and using the approximation techniques mentioned in [60] , the IAMP algorithm can be simplified when the sizes of the system m and n are sufficiently large. In particular, it can be shown that $\boldsymbol{A}_i^T\boldsymbol{A}_j = \frac{1}{m} + o\left(\frac{1}{m}\right)$ and hence Equation (5.3.3) becomes $\sigma_{\text{in}}^2 = \frac{1}{\delta}\sigma_{\text{out}}^2 + \sigma_w^2 + o\left(1\right)$. Furthermore, each

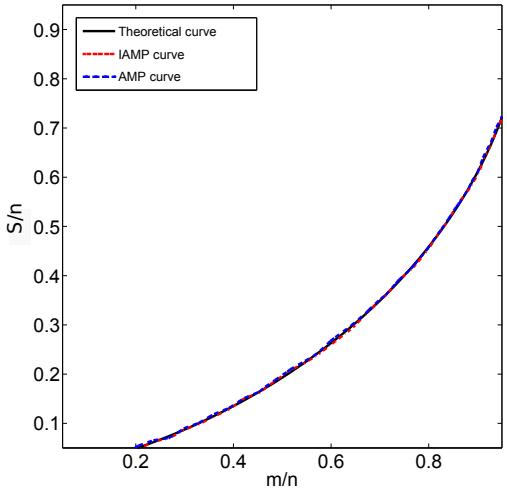


Figure 5.4.1: Phase transition for a standard Gaussian matrix. AMP and IAMP both achieve the same performance.

component of the matrix $A_{a,i} = O\left(\frac{1}{\sqrt{m}}\right)$. The last term in (5.3.4) becomes $\frac{1}{m}\sum_{i=1}^{n}\eta'\left(\tilde{y}_i^t;\theta_i^t\right)r_a^t = \frac{1}{\delta}\left\langle\eta'\left(\tilde{y}_i^t;\theta_i^t\right)\right\rangle r_a^t$. Hence, IAMP reduces to AMP.

Figure 5.4.1 provides the numerical comparison between AMP and IAMP (without the adjustment of the "output" noise variance). We consider the noise free case, i.e., $\sigma_w^2 = 0$. We are interested in the phase transition curve, that is, the exact reconstruction happens with dominant probability in the region below the curve while the recovery is not accurate with dominant probability in the region above the curve. (In empirical study, we use 50% probability to draw the phase transition curve.) The theoretic curve is obtained by asymptotic analysis presented in [60]. The empirical results are obtained via 100 independent trials. In the simulations, $n = 1000$, so that asymptotic analysis should be accurate. The simulation results suggest that the theoretical phase transition curve predicts the actual performance and the AMP and IAMP algorithms give the identical numerical performance.

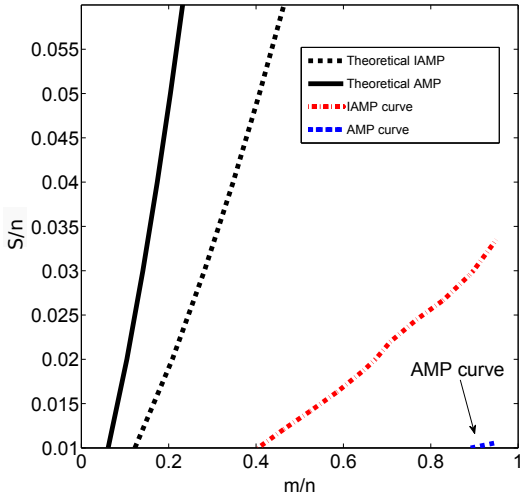


Figure 5.4.2: Phase transition for non i.i.d. Gaussian random matrices with $\rho = 0$. The gap between theoretical curve and practical curve of IAMP algorithm is smaller than AMP algorithm.

5.4.2 Non I.I.D. Gaussian Random Matrices

The more interesting results are obtained when the measurements are not the standard Gaussian matrix. Let $\boldsymbol{B} \in \mathbb{R}^{m\times n}$ be a standard Gaussian random matrix. Let $\boldsymbol{D} \in \mathbb{R}^{m\times m}$ be a diagonal matrix whose first $m/2$ diagonal entries (denoted by $d_k,\ k \in [m]$) are $\sqrt{\rho/2}$ and the rest $m/2$ diagonal entries are given by $\sqrt{(4-\rho)/2}$, where $\rho \in [0,4]$ is a given design constant. Let $\boldsymbol{H}$ be a normalised Hadamard matrix such that $\boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{H}\boldsymbol{H}^T = \boldsymbol{I}$. Define the measurement matrix as $\boldsymbol{A} = \boldsymbol{H}\boldsymbol{D}\boldsymbol{B}$.

This definition is motivated by equation (5.3.2). It is clear that $\boldsymbol{A}$ is a Gaussian random matrix. The marginal distribution of an entry $A_{a,i}$ is given by $A_{a,i} \sim \mathcal{N}\left(0, \frac{1}{m}\right)$. Furthermore, it can be shown that the cross-correlation

between two columns has variance given by

$$\begin{aligned}
&\mathrm{E}\left[\left(\boldsymbol{A}_i^T\boldsymbol{A}_j\right)^2\right] = \mathrm{E}\left[\left(\boldsymbol{B}_i^T\boldsymbol{D}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{D}\boldsymbol{B}_j\right)^2\right] \\
&= \mathrm{E}\left[\left(\boldsymbol{B}_i^T\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{B}_j\right)^2\right] = \mathrm{E}\left[\left(\sum_k d_k^2 B_{k,i}B_{k,j}\right)^2\right] \\
&= \frac{1}{m^2}\sum_{k=1}^{m} d_k^4 = \frac{\rho^2+(4-\rho)^2}{8m} =: \frac{1}{m}\sigma_c^2(\rho),
\end{aligned}$$

which is not $1/m$ (the value for standard Gaussian random matrix) unless $\rho = 2$. The resulted IAMP behaves quite different from AMP. Equation (5.3.3) can be approximated by $\sigma_{\text{in}}^2 = \frac{1}{\delta}\sigma_c^2(\rho)\,\sigma_{\text{out}}^2 + \sigma_w^2$.

Figure 5.4.2 compares AMP and IAMP (with the adjustment of the "output" noise variance). IAMP gives much better performance than AMP, and theoretic prediction of IAMP is also better than that of AMP. Unfortunately, in this case, neither of the theoretical predictions is accurate. The main reason is that the Onsager terms in AMP and IAMP will no longer completely cancel the correlations across iterations, as the matrix $\boldsymbol{A}$ is not a standard Gaussian random matrix. Thus, after first iteration, $\boldsymbol{A}$ and $\boldsymbol{x}^t$ become dependent which makes the calculation of (5.3.2) no longer precise. In this case, the threshold values θ_i^t's for the $\eta(\cdot)$ functions in both algorithms are not optimally tuned. The practical performances of the non-optimally tuned AMP and IAMP cannot match the theoretical curves (see the behaviour of the non-optimally tuned IST algorithm in Fig 1.3.1 as an example).

5.4.3 Radar Imaging

For simplicity, we consider the 1-D radar imaging. (The 2-D image in Figure 5.4.3 is obtained by scan the picture line by line. The size of the image is

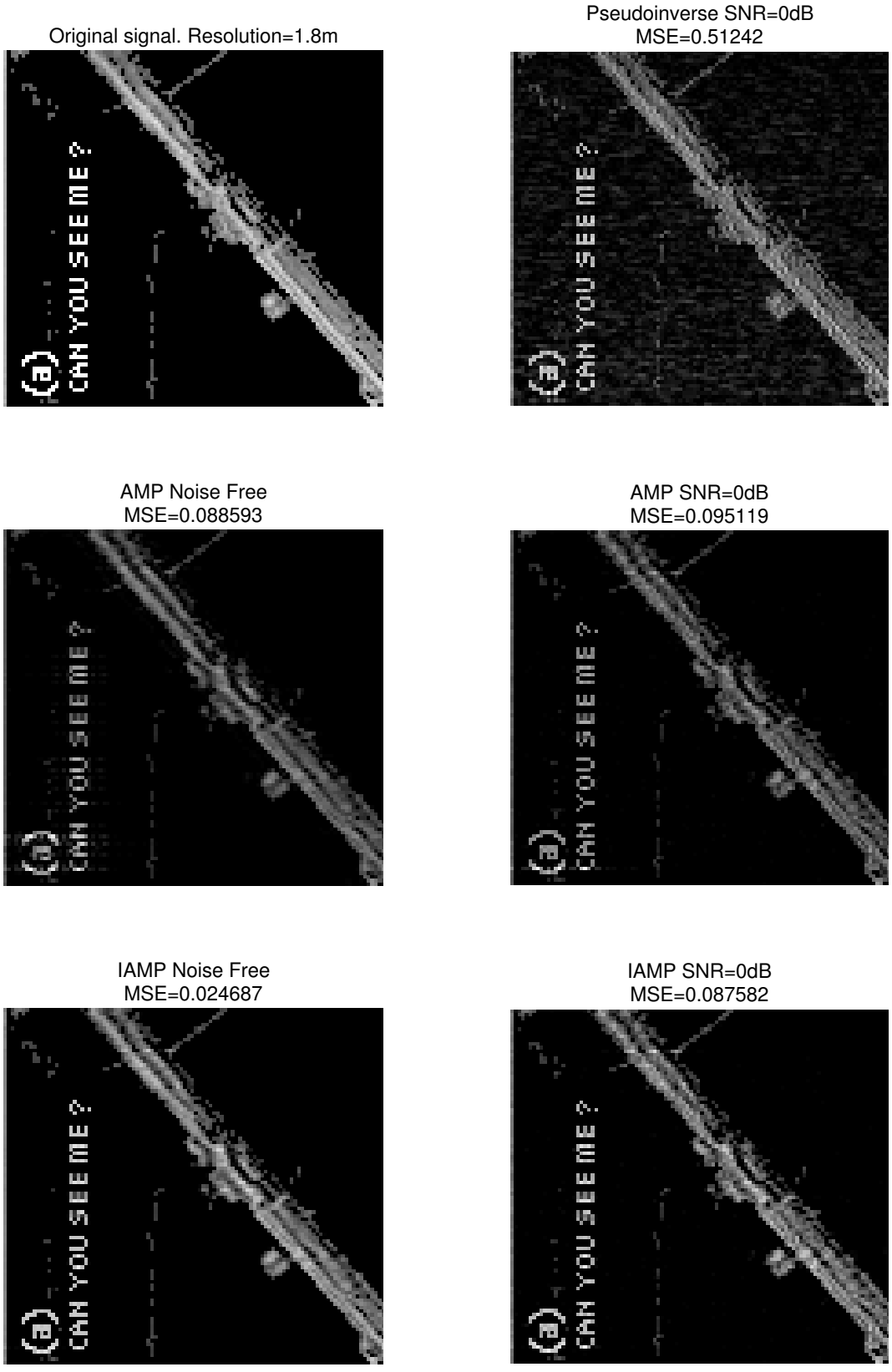


Figure 5.4.3: Radar imaging. IAMP algorithm is robust against noise.

207×194). A linear frequency modulated signal is transmitted and reflected by the existing targets in the scene. The received signal is then the superposition of the reflected signals. When the number of existing targets is small, this superposition is sparse. Depending on the distances between the radar system and the targets, the reflected signals are scaled versions of the transmitted signal with different delays. Mathematically, the received signal is given by $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, where columns of $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ are the transmitted signal with different delays, and $\boldsymbol{x} \in \mathbb{C}^n$ denotes the reflection coefficient vector and is sparse. The matrix $\boldsymbol{A}$ has two interesting structures. First, it is deterministic and Toeplitz. Second, it is tall rather than flat. Here we do not consider the compressed sensing scenario, i.e., no sub-sampling is performed. In practice, the sampling rate can be very high resulting $m > n$.

The simulated results are given in Figure 5.4.3. Besides AMP and IAMP (with the adjustment of the "output" noise variance), the least squares approach is also included. This is motivated by the fact that least squares approach can perfectly recover the signal $\boldsymbol{x}$ for the noise free case. However, least squares approach cannot incorporate the sparse prior information and therefore does not give a sparse solution for the noisy case. Figure of pseudo inverse demonstrates this point at SNR = 0dB. By contrast, due to accommodating sparse prior information, both AMP and IAMP perform well consistently for both high SNR and low SNR. As of the comparison between AMP and IAMP, it can be observed that IAMP results in less artifacts (see bottom left corner) and sharper images.[1] In summary, among the tested algorithms, IAMP is the most robust one against the noise.

[1]It is interesting to observe that the visual performance of AMP improves when SNR decreases. We don't fully understand the reason but suspect that it may be because the biased estimation of σ_{in}^2 (5.3.3) is neutralised by the large noise variance σ_w^2.

5.5 Conclusions

An improved AMP algorithm has been derived for non i.i.d. Gaussian measurement matrices. The performance improvement has been demonstrated by using a particularly constructed Gaussian matrix and a matrix from real applications. It turns out that the improvement is obtained by considering the correlation profile of the elements of the measurement matrix.

5.6 Proof

5.6.1 Proof of Lemma 5.3.1

We first calculate $p(\boldsymbol{y}|\, x_i,\, \boldsymbol{x}_{\sim i})$ by treating $\boldsymbol{y}$ and $\boldsymbol{x}_{\sim i}$ as constant vectors:

$$\ln p\left(\boldsymbol{y}|\, x_i,\, \boldsymbol{x}_{\sim i}\right) \tag{5.6.1}$$

$$= -\frac{1}{2\sigma_w^2}\left(\sum_a \left(y_a - A_{ai}x_i - \sum_{j\neq i} A_{aj}x_j\right)^2\right) + c \tag{5.6.2}$$

$$= -\frac{1}{2\sigma_w^2}\left(x_i^2 - 2\tilde{y}_i x_i + 2z_i x_i\right) + c + c' \tag{5.6.3}$$

$$= -\frac{1}{2\sigma_w^2}\left(\tilde{y}_i - z_i - x_i\right)^2 + c + c' + c'', \tag{5.6.4}$$

where c is a constant, $\tilde{y}_i = \boldsymbol{A}_i^T\boldsymbol{y}$, $z_i = \boldsymbol{A}_i^T\left(\sum_{j\neq i}\boldsymbol{A}_j x_j\right)$, and c' and c'' are two constants and their sum is given by

$$c' + c'' = -\frac{1}{2\sigma_w^2}\Big(\|\boldsymbol{y}\|_2^2 + \left\|\boldsymbol{A}_i^{\perp T}\boldsymbol{A}_{\sim i}\boldsymbol{x}_{\sim i}\right\|_2^2 - \tilde{y}_i^2 + 2\tilde{y}_i z_i - 2\left(\boldsymbol{A}_{\sim i}\boldsymbol{x}_{\sim i}\right)^T\boldsymbol{y}\Big), \tag{5.6.5}$$

where $\boldsymbol{A}_i^{\perp}$ is the orthogonal complement of $\boldsymbol{A}_i$. As a result, the integral in (5.3.1) becomes

$$\int p\left(\boldsymbol{y}|x_i, \boldsymbol{x}_{\sim i}\right) p\left(\boldsymbol{x}_{\sim i}\right) d\boldsymbol{x}_{\sim i} \tag{5.6.6}$$

$$\stackrel{a}{=} \int p\left(\boldsymbol{y}|x_i, z_i\right) p\left(z_i\right) dz_i \tag{5.6.7}$$

$$\stackrel{b}{=} \int c_1 \exp\left(-\frac{1}{2\sigma_w^2}\left(\tilde{y}_i - z_i - x_i\right)^2 - \frac{z_i^2}{2\sigma_{z_i}^2}\right) dz_i \tag{5.6.8}$$

where $\stackrel{a}{=}$ holds as $z_i = \boldsymbol{A}_i^T\left(\sum_{j\neq i}\boldsymbol{A}_j x_j\right)$ is a function of $\boldsymbol{x}_{\sim i}$ and we only care about the conditional probability $p\left(\boldsymbol{y}|x_i\right)$, $\stackrel{b}{=}$ holds as we treat z_i as a Gaussian variable (based on the joint Gaussian assumption of $\boldsymbol{A}\boldsymbol{x}$), thus, we have

$$p\left(x_i|\boldsymbol{y}\right) \propto c_2 \exp\left(-\frac{1}{2\left(\sigma_w^2 + \sigma_{z_i}^2\right)}\left(\tilde{y}_i - x_i\right)^2\right),$$

where c_1 and c_2 are two constants. This lemma is therefore proved.

Chapter 6

Conclusion and Future Research

In this chapter, we summarise the contents provided in the thesis and consider some potential problems for future research.

In Chapter 3, we discussed about the AMP-C-DCS algorithm proposed for solving the correlated DCS model. This model assumed that correlations existed in both measurement matrices and unknown signals across different measurement instances. In order to mathematically describe the correlation effects, we grouped the elements at the same location from different measurement instances to form the super components which were i.i.d drawn from some distributions p_A and p_x respectively. State evolution technique was used to analyse the asymptotic performance of the new algorithm in the asymptotic region. Correctness justification of the state evolution was provided for the two special cases in which the measurement matrices were assumed to be independent (DCS model) and identical (MMV model), respectively. While for the cases between them, due to the complexity and difficulty of applying the Gaussian condition lemma, we haven't found an efficient way to complete the justification. We consider this unsolved task is worth for future research, of

course, new techniques should be introduced to simplify the proof or to avoid the complexity issue that appeared during our analysis. Another potential work is to consider the measurement matrices and signals that do not have the same dimensions by considering that different kinds of sensors may be applied in practice. Thus how to describe the correlation effects among these super components reasonably becomes an urgent subtask.

In Chapter 4, we proposed a quadratically decreasing SNR model according to a practical signal transmission/receiving system with fixed energy budget. Under this condition, we were able to find an optimal number of measurements to minimize the MSE of estimation by applying the state evolution technique of the AMP algorithm. We considered the Gaussian, Bernoulli-Gaussian and Least-Favourite distributions for signal models in both real and complex domains. The analysis showed that for these distributions, the normalized optimal number of measurements ($\delta^\dagger$) was upper bounded by 2. Although based on the simulation results, we could always find an optimal $\delta^\dagger$ for different kinds of distributions listed above and different noise base level σ_0^2, the uniqueness of the optimal $\delta^\dagger$ was not rigorously proved. In order to make our analysis and model more solid, the justification of the uniqueness of $\delta^\dagger$ should be necessary and physical system should be built to check the correctness of our proposed model.

In Chapter 5, we considered the drawback of the AMP algorithm which was the assumption that the measurement matrix was a standard Gaussian random matrix. This kind of assumption is difficult to achieve in practice. It had been numerically observed that the performance of AMP might severely deteriorate if the measurement matrix was significantly different from the assumption. Under this circumstance, we proposed an improved AMP algorithm based

on a new message passing mechanism where all messages were computed at the variable nodes. Simulations showed that the improved AMP algorithm outperformed the AMP algorithm for the non-ideal measurement matrices but achieved same performance for the ideal case. The main problem was that for both algorithms, when the measurement matrices were not ideal Gaussian, there were gaps between the practical performances and theoretical curves achieved by state evolution, although, the gap of our proposed algorithm was smaller. We consider this phenomenon may be caused by the Onsager term which is not optimally designed based on the structure or information provided by the measurement matrix. Future research is required in this direction.

Bibliography

[1] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[2] M. M. Abo-Zahhad, A. I. Hussein, and A. M. Mohamed, "Compressive sensing algorithms for signal processing applications: A survey," *International Journal of Communications, Network and System Sciences*, vol. 8, no. 06, p. 197, 2015.

[3] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec 2005.

[4] D. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[5] S. Qaisar, R. Bilal, W. Iqbal, M. Naureen, and S. Lee, "Compressive sensing: From theory to applications, a survey," *Journal of Communications and networks*, vol. 15, no. 5, pp. 443–456, 2013.

[6] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar, 2008.

[7] M. Davenport, P. Boufounos, M. Wakin, and R. Baraniuk, "Signal processing with compressive measurements," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 445–460, 2010.

[8] D. Taubman, "Image compression fundamentals, standards and practice," *JPEG-2000*, 2002.

[9] M. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. Davies, "Sparse representations in audio and music: From coding to source separation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, June 2010.

[10] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *Multiscale Modeling & Simulation*, vol. 7, no. 1, pp. 214–241, 2008.

[11] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[12] E. Candes and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, p. 717, 2009.

[13] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[14] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[15] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[16] E. Candes and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *The Annals of Statistics*, pp. 2313–2351, 2007.

[17] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Asilomar Conference on Signals, Systems and Computers*, pp. 40–44 vol.1, Nov, 1993.

[18] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of computational mathematics*, vol. 9, no. 3, pp. 317–334, 2009.

[19] D. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[20] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, May, 2009.

[21] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis (ACHA)*, vol. 26, no. 3, pp. 301–321, 2009.

[22] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse prob-

lems," *IEEE Journal of selected topics in signal processing*, vol. 1, no. 4, pp. 586–597, 2007.

[23] D. Donoho, "De-noising by soft-thresholding," *IEEE transactions on information theory*, vol. 41, no. 3, pp. 613–627, 1995.

[24] T. Blumensath and M. Davies, "Iterative hard thresholding for compressed sensing," *Applied and computational harmonic analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[25] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[26] Y. Nesterov, "Gradient methods for minimizing composite objective function," 2007.

[27] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

[28] D. L. Donoho, A. Maleki, and A. Montanari, "Supporting information to: Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci. USA*, 2009.

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

[30] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," *IEEE International Symposium on Information Theory Proceedings (ISIT)*, pp. 2168–2172, July 2011.

[31] A. Manoel, F. Krzakala, E. Tramel, and L. Zdeborová, "Sparse estimation with the swept approximated message-passing algorithm," *arXiv preprint arXiv:1406.4311*, 2014.

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

[33] A. Maleki, L. Anitori, Z. Yang, and R. G. Baraniuk, "Asymptotic analysis of complex lasso via complex approximate message passing (camp)," *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4290–4308, July 2013.

[34] P. Schniter, S. Rangan, and A. Fletcher, "Vector approximate message passing for the generalized linear model," *2016 50th Asilomar Conference on Signals, Systems and Computers*, pp. 1525–1529, Nov 2016.

[35] C. Rush and R. Venkataramanan, "Finite sample analysis of approximate message passing algorithms," *arXiv preprint arXiv:1606.01800v2*, 2017.

[36] C. Rush, A. Greig, and R. Venkataramanan, "Capacity-achieving sparse superposition codes via approximate message passing decoding," *IEEE Transactions on Information Theory*, vol. 63, no. 3, pp. 1476–1500, March 2017.

[37] A. Javanmard and A. Montanari, "State evolution for general approximate message passing algorithms, with applications to spatial coupling," *Information and Inference*, p. iat004, 2013.

[38] D. Donoho, A. Javanmard, and A. Montanari, "Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing," *arXiv preprint arXiv:1112.0708v2*, 2013.

[39] Y. Zhang, "Theory of compressive sensing via l1-minimization: A non-RIP analysis and extensions," *Journal of the Operations Research Society of China*, vol. 1, no. 1, pp. 79–105, 2013.

[40] A. Tillmann and M. Pfetsch, "The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1248–1259, 2014.

[41] A. Natarajan and Y. Wu, "Computational complexity of certifying restricted isometry property," *arXiv preprint arXiv:1406.5791*, 2014.

[42] F. Yang, S. Wang, and C. Deng, "Compressive sensing of image reconstruction using multi-wavelet transforms," *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems*, vol. 1, pp. 702–705, Oct 2010.

[43] X. Zhao and W. Dai, "On joint recovery of sparse signals with common supports," *IEEE International Symposium on Information Theory (ISIT)*, pp. 541–545, June 2015.

[44] R. Baraniuk and P. Steeghs, "Compressive radar imaging," *IEEE Radar Conference*, pp. 128–133, Mar, 2007.

[45] C. R. Berger, B. Demissie, J. Heckenbach, P. Willett, and S. Zhou, "Signal processing for passive radar using ofdm waveforms," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 1, pp. 226–238, Feb 2010.

[46] S. Som, L. C. Potter, and P. Schniter, "Compressive imaging using approximate message passing and a markov-tree prior," pp. 243–247, Nov 2010.

[47] U. S. Kamilov, S. Rangan, A. K. Fletcher, and M. Unser, "Approximate message passing with consistent parameter estimation and applications to sparse learning," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2969–2985, May 2014.

[48] J. Tan, Y. Ma, and D. Baron, "Compressive imaging via approximate message passing with image denoising," *IEEE Transactions on Signal Processing*, vol. 63, no. 8, pp. 2085–2092, April 2015.

[49] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, 2011.

[50] M. Lustig, D. Donoho, J. Santos, and J. M. Pauly, "Compressed sensing mri," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 72–82, March 2008.

[51] S. Vasanawala, M. Murphy, M. Alley, P. Lai, K. Keutzer, J. Pauly, and M. Lustig, "Practical parallel imaging compressed sensing MRI: Summary of two years of experience in accelerating body MRI of pediatric patients,"

IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI), pp. 1039–1043, Mar, 2011.

[52] W. Hou and C. Zhang, "A compressed sensing approach to low-radiation ct reconstruction," *2014 9th International Symposium on Communication Systems, Networks Digital Sign (CSNDSP)*, pp. 793–797, July 2014.

[53] "NHS choices: MRI scan," http://www.nhs.uk/conditions/mri-scan/Pages/Introduction.aspx, accessed@ 2017.07.12.

[54] S. Li, L. Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and internet of things," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2177–2186, Nov 2013.

[55] G. Quer, R. Masiero, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, "On the interplay between routing and signal representation for compressive sensing in wireless sensor networks," *2009 Information Theory and Applications Workshop*, pp. 206–215, Feb 2009.

[56] A. O'Donnell, J. Wilson, D. Koltenuk, and R. Burkholder, "Compressed sensing for radar signature analysis," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2631–2639, OCTOBER 2013.

[57] M. Herman and T. Strohmer, "High-resolution radar via compressed sensing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2275–2284, June 2009.

[58] D. Hsu, S. Kakade, J. Langford, and T. Zhang, "Multi-label prediction via compressed sensing," *Advances in neural information processing systems*, pp. 772–780, 2009.

[59] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, Feb 2009.

[60] A. Montanari, "Graphical models concepts in compressed sensing," *arXiv preprint arXiv:1011.4328*, 2010. [Online]. Available: http://arxiv.org/abs/1011.4328

[61] D. L. Donoho, A. Maleki, and A. Montanari, "The noise-sensitivity phase transition in compressed sensing," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6920–6941, 2011.

[62] G. Tzagkarakis, D. Milioris, and P. Tsakalides, "Multiple-measurement Bayesian compressed sensing using GSM priors for DOA estimation," *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 2610–2613, Mar, 2010.

[63] A. Yang, M. Gastpar, R. Bajcsy, and S. Sastry, "Distributed sensor perception via sparse representation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1077–1088, Jun, 2010.

[64] O. Lee, J. Kim, Y. Bresler, and J. Ye, "Compressive diffuse optical tomography: Noniterative exact reconstruction using joint sparsity," *IEEE Transactions on Medical Imaging*, vol. 30, no. 5, pp. 1129–1142, May 2011.

[65] M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk, "Distributed compressed sensing of jointly sparse signals," *Asilomar Conference on Signals, Systems and Computers*, pp. 1537–1541, Oct, 2005.

[66] J. Ziniel and P. Schniter, "Efficient high-dimensional inference in the multiple measurement vector problem," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 340–354, Jan, 2013.

[67] D. Sundman, S. Chatterjee, and M. Skoglund, "Distributed greedy pursuit algorithms," *Signal Processing*, vol. 105, pp. 298–315, 2014.

[68] T. Wimalajeewa and P. Varshney, "Cooperative sparsity pattern recovery in distributed networks via distributed-omp," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5288–5292, May 2013.

[69] ——, "Omp based joint sparsity pattern recovery under communication constraints," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 5059–5072, Oct 2014.

[70] K. Lee and Y. Bresler, "Subspace-augmented MUSIC for joint sparse recovery with any rank," *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 205–208, Oct, 2010.

[71] B. Rao, K. Engan, and S. Cotter, "Diversity measure minimization based method for computing sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, pp. ii–369–72 vol.2, May, 2004.

[72] Y. Lu and W. Dai, "Independent versus repeated measurements: a performance quantification via state evolution," *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Mar, 2016, in press.

[73] M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk, "Distributed compressed sensing of jointly sparse signals," *Asilomar Conference on Signals, Systems and Computers*, pp. 1537–1541, Oct, 2005.

[74] J. Kim, W. Chang, B. Jung, D. Baron, and J. Ye, "Belief propagation for joint sparse recovery," *arXiv preprint arXiv:1102.3289*, 2011.

[75] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, Aug 2005.

[76] Z. Zeinalkhani and A. H. Haghighatpanah, N.and Banihashemi, "On weighting/reweighting schemes for approximate message passing algorithms," pp. 1–5, May 2016.

[77] S. Rangan, A. K. Fletcher, V. K. Goyal, E. Byrne, and P. Schniter, "Hybrid approximate message passing," *IEEE Transactions on Signal Processing*, vol. 65, no. 17, pp. 4577–4592, Sept 2017.

[78] M. Mayer and N. Goertz, "Bayesian optimal approximate message passing to recover structured sparse signals," *arXiv preprint arXiv:1508.01104*, 2015.

[79] P. Schniter, "Turbo reconstruction of structured sparse signals," *44th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, March 2010.

[80] T. Schön and F. Lindsten, "Manipulating the multivariate gaussian density," *Division of Automatic Control, Linköping University, Sweden, Tech. Rep*, 2011.

[81] S. Verdu and S. Shamai, "Spectral efficiency of CDMA with random spreading," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 622–640, Mar, 1999.

[82] D. Tse and S. Hanly, "Linear multiuser receivers: effective interference, effective bandwidth and user capacity," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 641–657, Mar, 1999.

[83] G. Reeves and H. D. Pfister, "The replica-symmetric prediction for compressed sensing with gaussian matrices is exact," pp. 665–669, July 2016.

[84] M. Woodbury, "Inverting modified matrices," *Memorandum report*, vol. 42, no. 106, p. 336, 1950.

[85] J. R. Pierce, "Physical sources of noise," *Proceedings of the IRE*, vol. 44, no. 5, pp. 601–608, May 1956.

[86] D. Zhang, A. K. Bhide, and A. Alvandpour, "Design of cmos sampling switch for ultra-low power adcs in biomedical applications," pp. 1–4, 2010.

[87] B. Razavi, *Design of analog CMOS integrated circuits*. McGraw-Hill, 2000.

[88] A. Department, "Electronic warfare and radar systems engineering handbook," *A Comprehensive Handbook for Electronic Warfare and Radar Systems Engineers*, October 2013.

[89] Y. Lu and W. Dai, "Extended amp algorithm for correlated distributed compressed sensing model," *IEEE International Conference on Digital Signal Processing(DSP)*, Oct, 2016, in press.

[90] S. Rangan, "Estimation with random linear mixing, belief propagation and compressed sensing," pp. 1–6, 2010.

[91] S. Rangan, P. Schniter, and A. Fletcher, "On the convergence of approximate message passing with arbitrary matrices," pp. 236–240, June 2014.

[92] S. Rangan, A. Fletcher, P. Schniter, and U. Kamilov, "Inference for generalized linear models via alternating directions and bethe free energy minimization," *IEEE Transactions on Information Theory*, vol. 63, no. 1, pp. 676–697, Jan 2017.

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

[94] S. Rangan, P. Schniter, and A. Fletcher, "Vector approximate message passing," *arXiv preprint arXiv:1610.03082*, 2016.

[95] D. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," pp. 1–5, Jan 2010.