Compressive Sensing Algorithms with Applications to Massive MIMO Systems

by

Lixiang LIAN

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
in the Department of Electronic and Computer Engineering

Jan 2020, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

Lixiang LIAN

Jan 2020

Compressive Sensing Algorithms with Applications to Massive MIMO Systems

by

Lixiang LIAN

This is to certify that I have examined the above PhD thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

_____

Prof. Vincent K. N. LAU (Supervisor)

_____

Prof. Kevin CHEN (Acting Head of Department)

Thesis Examination Committee

1. Prof. Vincent K. N. LAU (Supervisor)	Department of Electronic and Computer Engineering
2. Prof. Ross MURCH	Department of Electronic and Computer Engineering
3. Prof. Danny H.K. TSANG	Department of Electronic and Computer Engineering
4. Prof. Ning CAI	Department of Industrial Engineering and Decision Analytics
5. Prof. Lin DAI (External Examiner)	Department of Electronic Engineering City University of Hong Kong

The Department of Electronic and Computer Engineering

Jan 2020

To My Family

Acknowledgements

These years of PhD study at the Hong Kong University of Science and Technology have been very fulfilling. First and foremost, I would like to express my appreciation and gratitude to my supervisor, Prof. Vincent LAU for his endless support and incessant encouragement throughout my PhD study. His high scientific standards, persistence, rigorous way of thinking and passion for academics have greatly influenced my attitude towards research. His outstanding academic insight, vision and deep knowledge have inspired this research. In addition, I appreciate the patience and freedom he gave to allow me to conduct the research I am interested in. It has been a true honor to be a PhD student of Prof. LAU.

I am also indebted to Prof. An LIU for his generous support and constant guidance during my PhD studies. The thoughtful discussion with him has always been inspirational to me. I learned so much from him about how to find out an interesting research topic, how to tackle a research problem, how to write a clean paper and how to deal with the challenges encountered during research. His wisdom, insight, encouragement and enthusiastic have greatly shaped my way of conducting research. This thesis would not be possible without the collaboration with him.

I have been fortunate to have enjoyed the guidance and support from Prof. Wei WANG during undergraduate study at Zhejiang University. He is the first person who led me into the wireless communication realm and gave me strong motivation to pursue the graduate-level research. I am tremendously grateful for his generous help, positive encouragement and his dedication to serving my best interests.

I would like to thank my Thesis Committee Members, Prof. Ross MURCH, Prof. Danny H.K. TSANG, Prof. Ning CAI, and Prof. Lin DAI. Their constructive comments and suggestions have helped to make this a better thesis. I thank Prof. Ling SHI and Prof. Wei CHEN for being my thesis proposal committee. I would also like to thank the faculty of the Department of Electronic and Computer Engineering for providing great courses and continued supports

for graduate study and living. In particular, I thank Prof. Vincent LAU, Prof. Ling SHI, Prof. Li QIU, Prof. Kani CHEN, Prof. Daniel P. PALOMAR and Prof. Qian ZHANG for their teaching.

I would like to thank my friends in Huawei-HKUST lab for their constant support, encouragement and company. I have many thanks to my group mates: Prof. Rui ZHAO, Prof. Yinglei TENG, Prof. Jisheng DAI, Fan ZHANG, Xiongbin RAO, Feibai ZHU, Naeimeh OMIDVAR, Jiachang LIU, Songfu CAI, Pak Him CHU, Huiyi GAO, Timothy LEE, Darren WONG, Borna KANANIAN, Victor BARSOUM, Shuqi CHAI, Yechao SHE, Manli YU, Biao HE, Bingpeng ZHOU, Huayan GUO, Xuanyu ZHENG, Ye XUE, Minjie Tang, Liqun SU and Fei HAN. I would also like to express my gratitude to a number of good friends around me at HKUST. I feel very lucky to have been able to spend a great deal of time with Di ZHAO, Wenchao DING, Lin YANG, Qianyi HUANG, Junxiao SONG, Ying SUN, Yiyong FENG, Tianyu QIU, Licheng ZHAO, Ziping ZHAO, Linlong WU, Junyan LIU, Xi PENG, Lu YANG, Rui WANG, Yuyi MAO, Xianghao YU, Liusha YANG, Shibo CHEN, Haobo LIANG, Xiaokang WANG, Runfa ZHOU, Ziyang GUO, Kemi DING, Zhaokang CHEN, Yanfang MO and Feng GAO. Thank you all!

Last, but the most, I can never forget the unconditional love and support from my parents, my sister, my brothers and my entire families. They have always believed in me, been fully supportive to all my decisions and provided me the required confidence and courage to face every challenge in my life. I would like to dedicate this thesis to them for their endless love, support and care. I would also like to express my deepest gratitude to my boyfriend for his self-giving accompany, support and encouragement during the whole time of my PhD journey.

Table of Contents

List of Figures

List of Tables

Compressive Sensing Algorithms with Applications to Massive MIMO Systems

by Lixiang LIAN

Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology

Abstract

Compressive sensing (CS) has attracted significant attention as a technique that undersamples high dimensional signals and accurately recovers them exploiting the sparsity of these signals. There are several ingredients of the CS algorithm. The first is the structure of the sparse signal. By exploiting additional signal structures in addition to the simple sparsity, additional performance gains can be obtained. How to choose a flexible yet tractable sparse prior to capture various sophisticated structured sparsity in specific application would be one of the challenges for the CS algorithm design. Another important ingredient that would affect the CS recovery performance is the measurement matrix. Different applications may result in measurement matrices with different features. How to handle a general measurement matrix would be another challenge for the CS algorithm design. In wireless communication system, due to the limited number of scatterers in the environment, the massive multi-input multi-output (MIMO) channel can be quite sparse under an appropriate spatial basis. Besides the channel sparsity, the massive MIMO channel further exhibits additional structures. In this thesis, we focus on the CS algorithm designs with applications to massive MIMO systems to exploit the possible structured sparsity and handle specific measurement requirement under different application contexts.

First, we consider channel support side information (CSSI) is available at base station, which can be exploited to enhance the channel estimation performance and reduce the pilot overhead. We propose a weighted LASSO algorithm to fully exploit the CSSI and propose an optimal weight policy to optimize the recovery performance. We also derive the closed-form accurate expression for the minimum asymptotic normalized squared error and characterize the minimum number of measurements required to achieve stable recovery.

Then, we consider a channel tracking problem in downlink frequency-division duplexing (FDD) massive MIMO system. We propose a two-dimensional Markov model to capture the two-dimensional (2D) dynamic sparsity of massive MIMO channels. We derive an effective message passing algorithm to recursively track the dynamic massive MIMO channels exploiting the 2D dynamic sparsity.

Besides the above works, we further propose a more general CS algorithm to solve the problem of recovering a structured sparse signal from a linear measurement model with uncertain measurement matrix. The proposed general framework can be utilized to provide

highly accurate user location tracking in massive MIMO systems. Specifically, a three-layer hierarchical structured sparse prior model is proposed to capture complicated structured sparsities. By combining the message passing and variational Bayesian inference (VBI) approaches via the turbo framework, the proposed Turbo-VBI algorithm is able to fully exploit the structured sparsity for robust recovery of structured sparse signals under an uncertain measurement matrix.

Abbreviations

2D	two-dimensional
2D-MM	two-dimensional Markov model
3LHS	three-layer hierarchical structured
ADC	analog to digital converter
AMP	approximate message passing
aNSE	asymptotic normalized squared error
AoA	angle of arrival
AoD	angle of departure
AO	alternating optimization
AS	angular spread
AWGN	Additive White Gaussian Noise
BCD	block coordinate descent
BS	base station
CDF	cumulative density function
CE	channel estimation
CGMT	convex Gaussian min-max theorem
CoSaMP	compressive sampling matching pursuit
C-RAN	cloud radio access network
CS	compressive sensing
CSI	channel state information
CSSI	channel support side information
DFT	discrete Fourier transformation
DWT	discrete wavelet transform
D-VBI	dynamic variational Bayesian inference
D-TOAMP	dynamic Turbo orthogonal approximate message passing
EM	expectation-maximization
FDD	frequency-division duplexing
GS	group-sparsity
GPS	Global Positioning System
i.i.d.	independent and identically distributed
LASSO	least absolute shrinkage and selection operator

LBSs	location-based services
LOS	line-of-sight
LS	least squares
LTE	Long Term Evolution
mmWave	millimeter-wave
mm-SSCM	millimeter-wave statistical spatial channel model
MAP	maximum a posteriori
MIMO	multi-input multi-output
ML	maximum likelihood
MM	majorization-minimization
MSE	mean square error
MMSE	minimum mean square error
MS	mobile station
NLOS	non-line-of-sight
OAMP	orthogonal approximate message passing
OFDM	orthogonal frequency-division multiplexing
OMP	orthogonal matching pursuit
PA	power amplifier
PDF	probability density function
PSI	prior support information
PTC	probabilistic temporal correlation
RIP	restricted isometry property
RMSE	root-mean-square error
RP	random permutation
RSS	received-signal-strength
SBL	sparse Bayesian learning
SCM	spatial channel model
SE	state evolution
SNR	signal noise ratio
SP	subspace pursuit
SPMP	sum-product message passing
TDD	time-division duplexing
TMGS	temporal Markov group-sparse
ToA	time-of-arrival
Tx	Transmitter
UCA	uniform circular array
ULA	uniform linear array
VBI	variational Bayesian learning
w.r.t.	with respect to

Notations

a, A	scalar
$\mathbf{x}$	vector
$\mathbf{X}$	matrix
$(\cdot)^T$	transpose
$(\cdot)^{-1}$	inverse
$(\cdot)^*$	conjugate
$(\cdot)^H$	conjugate transpose
$(\cdot)^\dagger$	Moore-Penrose pseudoinverse
$\text{rank}(\cdot)$	rank
$\lvert\cdot\rvert$	absolute value or cardinality of a set
$\mathbf{I}$	identity matrix
$\lVert\mathbf{x}\rVert$	ℓ_2-norm of vector $\mathbf{x}$
$\lVert\mathbf{x}\rVert_1$	ℓ_1-norm of vector $\mathbf{x}$
$\mathbf{x}[n]$	the n-th element of vector $\mathbf{x}$
$\boldsymbol{x}[\mathcal{I}] \in \mathbb{C}^{\lvert\mathcal{I}\rvert}$	subvector consisting of the elements of $\boldsymbol{x}$ indexed by the set $\mathcal{I}$
$\mathbf{X}[:, i]$	the i-th column of matrix $\mathbf{X}$
$vec(\mathbf{A})$	vectorization of matrix $\mathbf{A}$
$\otimes$	Kronecker product
$\mathcal{I} \backslash \mathcal{A}$	means the complement of set $\mathcal{A}$ with respect to set $\mathcal{I}$
$\mathcal{CN}(x; \mu, \nu)$	the PDF of a complex Gaussian random variable x with mean μ and variance ν
$\mathcal{CN}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	the PDF of a complex Gaussian vector $\mathbf{x}$ with expectation $\boldsymbol{\mu}$ and co-variance matrix $\boldsymbol{\Sigma}$
$\mathbf{A} = \text{Diag}(\mathbf{A}_1, \mathbf{A}_2)$	matrix $\mathbf{A}$ is a diagonal block matrix with $\mathbf{A}_1$ and $\mathbf{A}_2$ on the diagonal
$\mathbf{X} = \text{diag}(\mathbf{x})$	matrix $\mathbf{X}$ is a diagonal matrix with $\mathbf{x}$ on the diagonal
$[\mathbf{A}_1, \mathbf{A}_2] = \text{diagblock}(\mathbf{A})$	matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are extracted from the diagonal of matrix $\mathbf{A}$
$\text{tr}(\mathbf{X})$	trace of matrix $\mathbf{X}$
$a = \Theta(1)$	a is order 1 (i.e., $-\infty < a < \infty$)
$\delta(\cdot)$	Dirac delta function
$1(\cdot)$	indication function
$\mathbf{X} = [\mathbf{X}_1; \mathbf{X}_2]$	$\mathbf{X} = \left[\mathbf{X}_1^T, \mathbf{X}_2^T\right]^T$

$\mathbb{E}(\cdot)$	expectation
$\mathbb{R}$	real numbers
$\mathbb{C}$	complex numbers

Chapter 1

Introduction

1.1 Compressive Sensing

As the widely deployment of the sensors, data acquisition devices and the growth of the web and cloud infrastructures, tremendous volume of datasets have been generated, which poses a great challenge for the data acquisition and interpretation. The traditional acquisition of a signal is to sample it according to the Nyquist sampling theorem to guarantee no lose information and reconstruct it by the samples through a simple linear interpolation. However, in many applications, such as the imaging systems and video cameras, the Nyquist rate can end up with too many samples, and increasing the sampling rate can induce significant hardware cost. This motivates us to develop a new technique to reduce the number of measurements required to completely describe the signal without sacrificing the reconstruction fidelity.

For most of the natural signals and man-made signals, such as image, video and audio signals, the wavelet representation of these signals is approximately sparse, i.e., most of the coefficients are close to zero. In wireless communication system, due to the limited number of scatterers in the environment, the massive MIMO channel can be quite sparse under an appropriate spatial basis [2]. Therefore, compressive sensing (CS) is proposed as a new sampling paradigm that requires fewer measurements for a sparse signal. Mathematically, if $\mathbf{y} \in \mathbb{C}^M$ denotes a length M observation vector and $\mathbf{h} \in \mathbb{C}^N$ denotes a length N signal with $M \ll N$, the sampling process can be described as

$$\mathbf{y} = \mathbf{\Phi h} + \mathbf{n}, \tag{1.1.1}$$

where $\mathbf{\Phi} \in \mathbb{C}^{M\times N}$ represents a linear transform of signal $\mathbf{h}$, and $\mathbf{n} \in \mathbb{C}^M$ is the measurement noise. As mentioned earlier, most of the high-dimensional signals are compressible (sparse) in a suitable chosen basis, i.e.,

$$\mathbf{h} = \mathbf{A}\mathbf{x}, \tag{1.1.2}$$

where $\mathbf{A} \in \mathbb{C}^{N\times N}$ is the basis matrix, $\mathbf{x} \in \mathbb{C}^N$ is the sparse representation of $\mathbf{h}$ in the $\mathbf{A}$ domain. Therefore, compressible signal can be well approximated by its sparse representation. Denote the non-zero indices $\Omega = \{n : \mathbf{x}[n] \neq 0\}$ as the support of $\mathbf{x}$, we have $|\Omega| \ll N$. Substituting (1.1.2) into the linear measurement process (1.1.1), we can get

$$\mathbf{y} = \mathbf{\Phi}\mathbf{A}\mathbf{x} + \mathbf{n} = \mathbf{\Psi}\mathbf{x} + \mathbf{n}. \tag{1.1.3}$$

The goal in the CS algorithm is to recover a high dimensional sparse signal $\mathbf{x}$ from significantly fewer measurements $\mathbf{y}$ based on the signal model (1.1.3) with known measurement matrix $\mathbf{\Psi}$.

From the signal model (1.1.3), there are several ingredients of the CS algorithm. The first is the structure of the sparse signal. By exploiting additional signal structures in addition to the simple sparsity, additional performance gains can be obtained. The additional signal structures are presented in a wide range of applications, including magnetoencephalography (MEG) [3], dynamic magnetic resonance imaging (MRI) [4], underwater communication [5] and natural images [6]. In wireless communication system, due to the physical scattering structure in the environment, the angular domain channel support (i.e., the index set of non-zero elements of the channel vector) has a burst structure [7] and clustered structure [8] in the spatial domain. Moreover, due to the slowly changing environment, the channel support usually changes slowly compared to the instantaneous channel state information (CSI). As a result, the common support assumption for the channels over time has been made in [9] and [10]. Another important ingredient is the measurement matrix $\mathbf{\Psi}$. Different CS recovery algorithms have different requirements for the measurement matrix. For example, [11] shows that with i.i.d. Gaussian measurement matrix, which can be shown to have the restricted isometry property (RIP) with high probability, we can exactly reconstruct the sparse signal with overwhelmingly high probability via basis pursuit (or minimum ℓ_1 norm reconstruction) for noiseless case. On the other hand, different applications may result in measurement matrices with different features, which may even contain uncertain parameters. Therefore, it's

paramount to design efficient and robust CS algorithms based on different sparsity structures and measurement requirements under different applications.

There are several common methods to solve the CS problem.

1.1.1 Greedy Algorithms

Orthogonal matching pursuit (OMP) [12] and many variants of OMP such as the compressive sampling matching pursuit (CoSaMP) [13] and subspace pursuit (SP) [14] are greedy approaches, which iteratively improve their estimates by choosing the column of the measurement matrix that has the most correlation with the residual. The main difference between the other greedy algorithms and OMP is that instead of moving just one column to the active set at every iteration, they add more columns to the active set, and they allow removal of element from the active set as well. Later greedy algorithms have also considered structured sparse signal. For example, a joint OMP (JOMP) is proposed to recover partially joint sparse signal in [15], a simultaneous OMP (SOMP) is proposed to recover common sparse signals in [16]. These greedy algorithms have relatively low computational complexity. However, it's hard to extend the OMP-based algorithms to incorporate more complicated sparsity structures, such as Markov structure. Moreover, RIP used to analyze the performance for OMP-based algorithms is too loose to get any insightful information and greatly restricts the type of the measurement matrix that can be chosen.

1.1.2 Generalized LASSO

Another widely used method to recover the sparse signal $\mathbf{x}$ is the generalized LASSO [17], which solves the following problem:

$$\hat{\mathbf{x}} := \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{y} - \mathbf{\Psi}\mathbf{x}||_2^2 + \lambda f(\mathbf{x}), \tag{1.1.4}$$

where $f(\mathbf{x})$ is the regularization function, $\lambda \geq 0$ is the regularizer parameter. The regularization function could be used to promote the structure of the sparse signal $\mathbf{x}$. For example, if $\mathbf{x}$ is a group-sparse signal, i.e., $\mathbf{x}$ can be divided into blocks $\mathbf{x} = [\mathbf{x}_1; \cdots; \mathbf{x}_B]$ and only a few of them are non-zero, we can choose $f(\mathbf{x}) = \sum_{i=1}^{B} \|\mathbf{x}_i\|$ to impose the group-sparse structure of $\mathbf{x}$ [18]. For the standard sparse signal $\mathbf{x}$, $f(\mathbf{x}) = \|\mathbf{x}\|_1$ can be used to promote the i.i.d. sparsity of $\mathbf{x}$. There are several major drawbacks of generalized LASSO. First, the

performance is sensitive to the choice of the regularization function $f(\mathbf{x})$ and the value of regularizer parameter λ. Complicated structured sparsity cannot be well modeled by a simple function. Moreover, it's difficult to find the optimal regularizer λ, especially for complicated regularization function. Second, if we choose a non-convex regularization function, the complexity of solving the resulting problem is high and the algorithm can easily get stuck at a bad local optimum.

1.1.3 Approximate Message Passing

Approximate message passing (AMP) [19] applies Gaussian and quadratic approximations at the asymptotic region (i.e., $M, N \to \infty$) to loopy belief propagation, which is fast and highly accurate, and admits to rigorous analysis, i.e., state evolution (SE), when i.i.d. Gaussian measurement matrix and i.i.d. sparse priors are involved [20]. Various variations of AMP algorithm have been proposed to handle the non-i.i.d. Gaussian measurement matrix and more complicated sparse priors. For example, orthogonal approximate message passing (OAMP) has been proposed in [21–23] to achieve a better performance than AMP when the sensing matrix is a partial orthogonal matrix, such as a partial discrete Fourier transformation (DFT) matrix. Generalized AMP (GAMP) [20] has been proposed to handle non-Gaussian measurement noise. Moreover, AMP and turbo approach have been combined to design more advanced CS recovery algorithms such as the Turbo-AMP [24] and Turbo-CS [25], which can handle more complicated priors of $\mathbf{x}$ (e.g., Markov and Markov tree priors [26]). However, these AMP-based algorithms may diverge under a more general measurement matrix, especially when there is correlation between different columns of the measurement matrix or the mean of the measurement matrix is non-zero.

1.1.4 Sparse Bayesian Inference

Sparse-Bayesian-inference-based algorithms, such as sparse Bayesian learning (SBL) [27] and variational Bayesian inference (VBI) [28, 29] have been proposed to solve the CS problem, in which a two-layer hierarchical prior is used to model the i.i.d. sparsity or group-sparsity of unknown signal. SBL/VBI can deal with the unknown parameters involved in the CS model, such as the uncertain parameters induced by the priors, or the uncertain parameters involved in the measurement matrix $\mathbf{\Psi}$. The unknown parameters can be learned through expectation maximization (EM) [30] framework, then the sparse signal $\mathbf{x}$ is recovered using

maximum a posteriori (MAP) based on the learned parameters and the measurements $\mathbf{y}$. The limitations of the SBL/VBI are that the two-layer hierarchical prior [27–29] can not handle more complicated sparse priors that may occur in practice, such as the Markov tree priors [26] or the hidden Markov priors [31].

1.2 Thesis Contributions

Massive MIMO, which operates with a large number of antennas at the base station (BS), is a promising technology for future wireless systems [32]. It can provide large spatial multiplexing gain as well as array gain to enhance both the capacity and energy efficiency of wireless systems [33]. Accurate CSI estimation and tracking is essential to reap the benefit of massive MIMO for communication over a dynamic wireless channel [9, 10, 34–37]. In addition to the communication benefits, the massive MIMO technique could also be exploited to enable high-accuracy localization [38, 39]. In this thesis, we aim to design new CS algorithms based on the existing methods with applications to the massive MIMO systems, such as massive MIMO channel estimation (CE), massive MIMO channel tracking and user location tracking in massive MIMO systems.

1.2.1 Optimally-tuned Weighted LASSO for Massive MIMO Channel Estimation

To start with, we design an optimally-tuned weighted LASSO algorithm to fully exploit the statistical prior support information (PSI). In practice, it is possible to obtain some prior information of the support $\Omega = \{n : x[n] \neq 0\}$. For example, in many applications, we need to recover a sequence of sparse signals $\mathbf{x}(t)$ whose supports change slowly over time from a sequence of measurements $\mathbf{y}(t)$. In this case, the support estimated at the previous time can be used as the PSI to recover the current sparse signal. Usually, the PSI is imperfect due to the estimation error or the time varying statistics of the unknown signal, we cannot perfectly know the exact positions of the nonzero elements. However, it is possible to obtain a statistical prior about the support location. For example, a practitioner may have different level of confidence on the different parts of signal to be nonzero, i.e., the elements located in some parts are believed to be nonzero with higher probability compared to those located in other parts. In other settings, the probabilities on each entry being nonzero may be provided.

Therefore, it is important to incorporate such statistical PSI in an optimal way to enhance the recovery performance.

In massive MIMO system, it is shown [40] that channel statistics are changing slowly. Therefore, the channel support estimated previously provides some prior information for the current CE. Considering a BS with imperfect channel support side information (CSSI), the proposed optimally-tuned weighted LASSO can be employed to exploit the CSSI in massive MIMO system to enhance the CE performance/reduce the number of required pilots.

1.2.2 Dynamic Turbo-OAMP for Downlink FDD-Massive MIMO Channel Tracking

In LASSO algorithm, it is difficult to design a proper penalty function $f(\mathbf{x})$ to capture sophisticated sparsity structure of $\mathbf{x}$ with low computational cost and good performance. However, in some applications, the unknown signal exhibits quite complicated sparsity structures. On the other hand, OAMP is a variation of the well-known AMP [19, 41] and it is shown in [23] to achieve a better performance than AMP when the measurement matrix is a partial orthogonal matrix. However, the OAMP in [23] and the associated SE analysis only works for i.i.d. priors. We extend the OAMP to dynamic Turbo-OAMP (D-TOAMP), which works for arbitrary priors, and thus provides a systematic framework for the design of dynamic sparse sequence tracking algorithms.

In massive MIMO system, due to the clustered scattering, the support of the massive MIMO channels will have clustered structure in spatial domain, i.e., the non-zero elements of the angular domain channel tend to concentrate on a few clusters. Furthermore, due to slowly changing propagation environment, the dynamic scattering structures are temporally correlated, which will result in the probabilistic dependencies of channels across time. Therefore, it is paramount to design a new algorithm exploiting such 2D dynamic sparsity (i.e., structured sparsity in the spatial domain and probabilistic temporal dependency of channel in the temporal domain) of massive MIMO channels to improve the channel tracking performance. The proposed D-TOAMP algorithm can be employed to recursively track the time-varying channels with a 2D dynamic sparsity.

1.2.3 Turbo-VBI for User Location Tracking in Massive MIMO Systems

Even though the AMP-based algorithms, such as Turbo-AMP [24] and Turbo-CS [25] can exploit the sophisticated priors, they perform badly under a general measurement matrix, especially when the measurement matrix is ill-conditioned. The performance of SBL/VBI is insensitive to the measurement matrix. However, the two-layer hierarchical prior in SBL/VBI can not handle more complicated sparse priors. This motivates us to propose a novel Turbo-VBI framework to overcome the drawbacks of the existing methods and achieve robust recovery of structured sparse signals with more general uncertain measurement matrix. The proposed Turbo-VBI framework can exploit sophisticated structured sparsity to improve the recovery performance. It is robust w.r.t. the uncertain parameters in the measurement matrix and prior distribution and it works well for more general measurement matrices with possibly correlated columns.

Due to the high directivity and increased spectral efficiency, the massive MIMO technology employed in 5G networks can potentially provide accurate user localization. We focus on using massive MIMO systems for efficient tracking of user's location. Instead of performing individual localization at each time slot, we could exploit the user mobility and the temporal correlation of wireless channels to improve location tracking accuracy, which will lead to special structured sparsity for massive MIMO channels. Moreover, the measurement matrix in the location tracking problem is ill-conditioned with off-grid parameters. Therefore, Turbo-VBI algorithm can be employed to recursively track user's location in massive MIMO systems.

1.3 Thesis Organization

In this thesis, we focus on compressive sensing algorithms designs with applications to the massive MIMO systems. The remainer of this thesis is organized as follows.

In Chapter 2, we propose a weighted LASSO algorithm to fully exploit the statistical PSI and optimize the recovery performance. In the proposed algorithm, we exploit the most general statistical PSI model, a multi-level PSI, and incorporate it into the LASSO using a weighted l_1 norm penalty function. An optimal weight policy is proposed to minimize the asymptotic normalized squared error (aNSE). We also derive the closed-form accurate

expression for the minimum aNSE and characterize the minimum number of measurements required to achieve stable recovery. Based on this, we discuss the impact of PSI quality on the aNSE performance of the proposed algorithm. Theoretical analysis and simulations both show the performance advantages of our proposed solution over various baselines. Moreover, we utilize the proposed optimally-tuned weighted LASSO algorithm to exploit the CSSI in massive MIMO system to enhance the CE performance. The material in this chapter has been presented in part in [J1, C1].

In Chapter 3, we consider downlink FDD-massive MIMO system operating with limited scattering around the base station and flat fading channel is considered. We propose a two-dimensional Markov Model (2D-MM) to capture the 2D dynamic sparsity of massive MIMO channels. The 2D-MM has the flexibility to model different propagation environments in practice. We derive an effective message passing algorithm, i.e., D-TOAMP, to recursively track a dynamic massive MIMO channel with a 2D-MM prior. The proposed D-TOAMP algorithm does not require knowledge of the 2D-MM channel parameters, which could be automatically learned through the expectation maximization framework. Extensive simulations show that the proposed D-TOAMP can achieve significant gains over the existing algorithms under realistic channels. The material in this chapter has been presented in part in [J2].

In Chapter 4, we propose a novel Turbo-VBI algorithm framework, in which a three-layer hierarchical structured (3LHS) sparse prior model is proposed to capture various sophisticated structured sparsities that may occur in practice. By combining the message passing and VBI approaches via the turbo framework, the proposed Turbo-VBI algorithm is able to fully exploit the structured sparsity (as captured by the 3LHS sparse prior) for robust recovery of structured sparse signals under an uncertain measurement matrix. The material in this chapter has been presented in part in [J4, J6, J7].

In Chapter 5, we apply the Turbo-VBI framework to user location tracking problem in massive MIMO systems. Under this application scenario, we firstly propose a 3LHS sparse prior, i.e., temporal Markov group-sparse (TMGS) model, based on a grid reference to capture the probabilistic temporal correlation and group sparsity of the massive MIMO channels jointly. Then based on the Turbo-VBI framework, we propose a variant of Turbo-VBI algorithm, i.e., dynamic variational Bayesian inference (D-VBI) algorithm, to handle the TMGS priors under ill-conditioned measurement matrix in the location tracking problem. The D-VBI algorithm can jointly recover the user's coarse location in the grid reference and refine

the off-grid points to automatically learn the user's exact location to high accuracy. Moreover, the TMGS-based D-VBI algorithm can provide prior information about the user's next location and the possible arriving directions of the future channels to the consecutive time slot to improve the location tracking accuracy. Finally, we verify the superior performance of the proposed location tracking algorithm by extensive simulations. The material in this chapter has been presented in part in [J3].

In Chapter 6, we close this thesis with a brief summary and discussions of future research directions.

1.4 Publications

The publications during my PhD study are listed below.

Journal Papers

J1. **L. Lian**, A. Liu and V. K. N. Lau, "Weighted LASSO for Sparse Recovery With Statistical Prior Support Information," in IEEE Transactions on Signal Processing, vol. 66, no. 6, pp. 1607-1618, 15 March15, 2018.

J2. **L. Lian**, A. Liu and V. K. N. Lau, "Exploiting Dynamic Sparsity for Downlink FDD-Massive MIMO Channel Tracking, in IEEE Transactions on Signal Processing," vol. 67, no. 8, pp. 2007- 2021, 15 April15, 2019.

J3. **L. Lian**, A. Liu and V. K. N. Lau, "User Location Tracking in Massive MIMO Systems via Dynamic Variational Bayesian Inference, in IEEE Transactions on Signal Processing," vol. 67, no. 21, pp. 5628-5642, 1 Nov.1, 2019.

J4. **L. Lian** and V. K. N. Lau, "Configuration Optimization and Channel Estimation in Hybrid Beamforming mmWave Systems with Channel Support Side Information," submitted to IEEE Transactions on Signal Processing, 2019.

J5. A. Liu, **L. Lian** and V. K. N. Lau, "Downlink Channel Estimation in Multiuser Massive MIMO With Hidden Markovian Sparsity," in IEEE Transactions on Signal Processing, vol. 66, no. 18, pp. 4796-4810, 15 Sept.15, 2018.

J6. A. Liu, **L. Lian**, V. K. N. Lau and M. Zhao, "5G-based Cooperative Localization for Vehicle Platoons: A Turbo Approach," to appear in IEEE Transactions on Signal Processing, 2019.

J7. A. Liu, G. Liu, **L. Lian**, V. K. N. Lau and M. Zhao, "Robust Recovery of Structured Sparse Signals with Uncertain Sensing Matrix: A Turbo-VBI Approach," to appear in IEEE Transactions on Wireless Communications, 2019.

Conference Papers

C1. **L. Lian**, A. Liu and V. K. N. Lau, "Optimal-Tuned Weighted LASSO for Massive MIMO Channel Estimation with Limited RF Chains," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1-6.

C2. A. Liu, V. Lau, M. L. Honig and **L. Lian**, "Compressive RF training and channel estimation in massive MIMO with limited RF chains," 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-6.

C3. **L. Lian** and V. K. N. Lau, "Compressive Channel Estimation in mmWave Systems with Flexible Hybrid Beamforming Architecture," submitted to 2020 IEEE International Conference on Communications (ICC).

Chapter 2

Weighted LASSO for Sparse Recovery with Statistical Prior Support Information

2.1 Introduction

We consider recovering the unknown sparse signal $\mathbf{x}^\star \in \mathbb{R}^N$ (i.e., the number of non-zero entries of $\mathbf{x}^\star$ is much smaller than N) based on the measurement matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$ [1] and measurements $\mathbf{y} \in \mathbb{R}^M$ from the following model:

$$\mathbf{y} = \mathbf{A}\mathbf{x}^\star + \mathbf{n}, \tag{2.1.1}$$

where $M \ll N$ and $\mathbf{n} \in \mathbb{R}^M$ is the measurement noise vector. The entries of $\mathbf{n}$ are i.i.d. Gaussian with mean 0 and variance σ^2. A widely used method to recover the sparse signal $\mathbf{x}^\star$ is the l_2^2-LASSO, which solves the following problem:

$$\hat{\mathbf{x}} := \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 + \sigma\lambda \|\mathbf{x}\|_1 . \tag{2.1.2}$$

We denote the support of $\mathbf{x}$ as $\Omega \triangleq \{n : \mathbf{x}[n] \neq 0\}$ for brevity. Without any PSI, standard LASSO in (2.1.2) is a useful method for sparse recovery. In some settings, a statistical prior about the support of the sparse signal may be provided. It is critical to optimally incorporate such statistical PSI to enhance the recovery performance. One major approach to incorporate

[1] We use $\mathbf{A}$ as the notation for measurement matrix in this chapter.

the statistical PSI is weighted l_1 norm minimization, which has several variations. For example, [42–46] studied the following weighted l_1 norm minimization problem in a noiseless case:

$$\hat{\mathbf{x}} := \underset{\mathbf{x}}{\operatorname{argmin}} \ \|\mathbf{x}\|_{1,\mathbf{w}} \text{ subject to } \mathbf{A}\mathbf{x} = \mathbf{y}, \tag{2.1.3}$$

where $\|\mathbf{x}\|_{1,\mathbf{w}}$ denotes the weighted l_1 norm, given by

$$\|\mathbf{x}\|_{1,\mathbf{w}} = \sum_{i=1}^{N} \mathbf{w}[i] \, |\mathbf{x}[i]| \, , \tag{2.1.4}$$

and $\mathbf{w}$ is a weight depending on the statistical PSI, $\mathbf{w}[i]$ and $\mathbf{x}[i]$ are the i-th element of $\mathbf{w}$ and $\mathbf{x}$, respectively. In a noisy case, the following problems, (2.1.5) and (2.1.6), were studied in [47–50] and [51], respectively,

$$\hat{\mathbf{x}} := \underset{\mathbf{x}}{\operatorname{argmin}} \, ||\mathbf{x}||_{1,\mathbf{w}} \text{ subject to } ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2 \leq \epsilon, \tag{2.1.5}$$

$$\hat{\mathbf{x}} := \underset{\mathbf{x}}{\operatorname{argmin}} \, \frac{1}{2}||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \sigma||\mathbf{x}||_{1,\mathbf{w}}. \tag{2.1.6}$$

Problem (2.1.6) is also called a weighted LASSO[2] since it is a generalization of the standard LASSO in (2.1.2).

How to choose the weights $\mathbf{w}$ is a subtle process and is critical to guarantee the performance of PSI-aided compressive signal recovery algorithms. There are several existing methods to determine the weights $\mathbf{w}$ based on different forms of PSI available.

0-1 weights with estimated support: If a support estimation $\tilde{\Omega}$ is given, the modified-CS proposed in [42, 47] and modified basis pursuit denoising proposed in [51] incorporated this prior information using weighted l_1 norm minimization with zero weight on the known (but possibly inaccurate) support part. Namely, given the support estimation $\tilde{\Omega}$, weight $w_i = 0$ was set for the indices $i \in \tilde{\Omega}$, and $w_i = 1$ otherwise. However, with inaccurate support estimation, setting zero weight on the known support part is usually not optimal. Intuitively, the optimal choice of weight should depend on the PSI accuracy. If the PSI is perfect, the weight on the indices $\tilde{\Omega}$ should be zero to impose the sparsity outside of $\tilde{\Omega}$. If the PSI is very inaccurate, we should also penalize the high values $\mathbf{x}[i]$ in $\tilde{\Omega}$ by assigning nonzero weight.

[2]In weighted LASSO formulation, the actual weight vector of the weighted ℓ_1 norm regularization term is $\sigma\mathbf{w}$, the weight vector that we optimize in this chapter is normalized (w.r.t. the noise standard deviation σ) weight vector $\mathbf{w}$, which is a scaling coefficient of the actual weight. Therefore, the actual weight in the weighted LASSO formulation contains the noise effect.

Therefore, it can be conjectured that the optimal weight decreases with the accuracy of the PSI.

Adaptive weights with estimated support in the noiseless case: In [43], an estimated support is also provided. They used the weighted nonuniform null space property to analyze the necessary and sufficient condition for successful recovery. They also proposed to choose the weight on the estimated support part as $1-\alpha$, where α is the support accuracy, if the weight on the remaining part is set to be 1.

PSI-aware weights with two-level PSI in the noiseless case: [44,45] studied the case in which the PSI is defined in terms of two disjoint sets with the probability of being nonzero in each set provided and two different weights are assigned to the two sets. [44] and [45] derived the sufficient recovery condition based on the Grassman angle approach and "escape through a mesh lemma", respectively. However, they did not directly provide any method to calculate the optimal weights.

Optimal weights with multi-level PSI in the noiseless case: Recently, [46] adopted the multi-level PSI model, which separated the complete index set $[1,\cdots,N]$ into T disjoint partitions, and different weights could be assigned to each partition. In [46], the authors used a notion of statistical dimensions to provide a simple analytical formula for optimal weights which minimizes the measurement threshold needed for exact recovery[3].

However, the adaptive weights in [43] and optimal weights in [46] are all discussed under a noiseless setup. Their technical tools are not applicable to a noisy scenario. To the best of our knowledge, the optimal choice of weights in a noisy case remains an unexplored problem.

In this chapter, we consider the general multi-level PSI model under a noisy setup, and propose an *optimally-tuned weighted LASSO* algorithm to exploit the statistical PSI to enhance the recovery performance and reduce the number of measurements required for stable recovery[4]. The main contributions are summarized below.

- **Accurate Recovery Error Bound:** To determine the optimal weights in the weighted LASSO for a noisy case, we need to first obtain an accurate recovery error bound. The reconstruction error bound of weighted l_1 norm minimization with PSI has been discussed in [47–50]. However, their results are all based on the RIP [53], which is

[3]Compared to [46], this chapter considers a more practical model with noise in the measurements. Due to the consideration of noise, the problem formulations, analytical approaches, performance metrics and optimal weights formulas are all different.

[4]Stable recovery means that the ratio between the estimation error and the noise variance is bounded as the noise variance goes to zero [52].

known to be too loose to obtain any insightful information. To overcome this challenge, we apply the CGMT approach in [52] and derive an accurate expression for the aNSE of the proposed algorithm, which is defined as the ratio between the estimation error and the noise variance as the noise variance goes to zero.

- **Weights Optimization and Closed-form Expression of the Minimum aNSE:** Based on the expression of the aNSE, we derive the optimal weights that minimize the aNSE. Moreover, we obtain a closed-form expression of the minimum aNSE under the optimal weights and characterize the minimum number of measurements required for stable recovery.

- **Impact of PSI Quality on the aNSE Performance:** We also analyze the impact of PSI quality on the performance of the optimally-tuned weighted LASSO, which theoretically shows the type of PSI the proposed optimally-tuned weighted LASSO can benefit from.

The rest of this chapter is organized as follows. In Section 2.2, we describe the weighted LASSO algorithm to exploit the multi-level PSI and apply the algorithm to channel estimation problem in massive MIMO systems. In Section 2.3, we derive an accurate aNSE, the optimal weights which minimize the aNSE, and the closed-form minimum aNSE. Based on this, we offer some discussions. In Section 2.4, we discuss the impact of PSI quality on the performance of the optimally-tuned weighted LASSO. The analytical results are verified numerically in Section 2.5, and summaries are given in Section 2.6.

2.2 Weighted LASSO with Multi-level Prior Support Information

2.2.1 Multi-level Prior Support Information

In this chapter, the unknown vector $\mathbf{x} \in \mathbb{R}^N$ is a K sparse signal, i.e., $|\Omega| = K$. Besides the prior information that the unknown vector is K sparse, we are also given some statistical prior information about where the support is situated. Specifically, we divide the complete index set $\mathbf{n} = [1, \cdots, N]^T \in \mathbb{R}^N$ into T disjoint partitions $\{S_t\}_{t=1}^T$; we call S_t the *t-th PSI component set* in the rest of the chapter. Two parameters α_t, β_t are associated with each set

S_t, which are defined as follows:

$$\alpha_t = \Pr\left(\mathbf{x}[i] \neq 0\right), \forall i \in S_t; \beta_t = \frac{|S_t|}{N}. \tag{2.2.1}$$

By this, we mean that for any $i \in S_t$, $\mathbf{x}[i]$ is nonzero with probability α_t, and is zero with probability $1 - \alpha_t$. When $\alpha_t = 1$, this means the set S_t is exactly part of the support of unknown vector $\mathbf{x}$. When $0 < \alpha_t < 1$, this means only part of the indices in set S_t belong to the support. When $\alpha_t = 0$, this means the corresponding indices of vector $\mathbf{x}$ in set S_t could be ignored in the recovery process. Therefore, we only consider the case when $\alpha_t > 0$ in the following discussion. We use α_t to denote the PSI accuracy in the t-th PSI component set S_t. β_t reflects the size of set S_t. Therefore, the average sparsity in set S_t (the average number of non-zero elements indexed in S_t) is $N\beta_t\alpha_t$.

The information set $\mathbb{I} = \left\{K, \{S_t, \alpha_t, \beta_t\}_{t=1}^{T}\right\}$ is our prior information. Since it is comprised of partial PSI with different accuracy levels (different parameters $\{S_t, \alpha_t, \beta_t\}$), we call such statistical PSI as multi-level PSI. Note that the same multi-level PSI is also considered in [46] for the noiseless case and it includes many PSI models in previous works as special cases. For example, if $T = 2$, multi-level PSI degrades to two-level PSI as in [44, 45]. If $T = 2$, $S_1 = \tilde{\Omega}$, and $\alpha_1 = |\tilde{\Omega} \cap \Omega|/|\tilde{\Omega}|$, multi-level PSI degrades to the case when estimated support is given as in [42,43,47,48,51,54]. If $T = m+1$, $\cup_{t=1,\cdots m} S_t = \tilde{\Omega}$,$\alpha_t = |S_t \cap \Omega|/|S_t|$ for $t = 1, \cdots, m$, the multi-level PSI degrades to the PSI model as in [49]. The standard compressive sensing model is a special case of multi-level PSI where these probabilities α_t are all equal to K/N.

2.2.2 Application Examples

In this section, we give two important application examples that involve sparse signal recovery with multi-level PSI.

Example 2.1 (Massive MIMO Channel Estimation Problem)**.** Consider a massive MIMO system with one BS serving single-antenna users, where the BS is equipped with a *half-wavelength space* ULA comprised of N antennas. To estimate the downlink channel $\mathbf{h}^H \in \mathbb{C}^{1\times N}$, the BS transmits M training sequences $\mathbf{p}_t \in \mathbb{C}^{N\times 1}, t = 1, ..., M$. Then the received

signal $\mathbf{y} \in \mathbb{C}^{M\times 1}$ of a user can be written as

$$\mathbf{y} = \mathbf{P}\mathbf{h} + \mathbf{n}, \tag{2.2.2}$$

where $\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_M]^H$ is a $M \times N$ pilot matrix which is known to the receiver and $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ is the Gaussian noise. Define the angular domain channel [2] as $\mathbf{x}^\star = \mathbf{F}\mathbf{h}$, where $\mathbf{F}$ is the DFT matrix. Then the received signal can be rewritten as

$$\mathbf{y} = \mathbf{P}\mathbf{F}^H\mathbf{x}^\star + \mathbf{n} = \mathbf{A}\mathbf{x}^\star + \mathbf{n}, \tag{2.2.3}$$

where $\mathbf{A} = \mathbf{P}\mathbf{F}^H$. Note that the angular domain channel $\mathbf{x}^\star$ is usually sparse due to limited scatters at the BS [7]. Therefore, (2.2.3) is the CS model in (2.1.1) with measurement matrix $\mathbf{A}$ and sparse signal $\mathbf{x}^\star$. Moreover, the channel support, denoted by $\Omega \triangleq \{n : \mathbf{x}^\star[n] \neq 0\}$, only depends on the AoD at the BS in the downlink pilot transmission, which is determined by the scattering environment around the BS. Since the scattering environment usually changes at a slow timescale, the channel support also changes very slowly. And thus the BS can obtain some knowledge of channel support from previously estimated channel supports using long term stochastic learning [55]. However, the BS has imperfect knowledge of Ω either due to the random nature of the channel or the estimation error in the channel support learning process. Specifically, the BS can obtain an estimated channel support $\hat{\Omega}$ with $|\hat{\Omega}| = \hat{L}$, a lower bound on the number of correct indices in the estimated support $|\Omega \cap \hat{\Omega}| \geq L_c$, and an upper bound on the actual support size $|\Omega| \leq L$ [36]. For example, $\hat{\Omega}$ could be the previously estimated channel support at the BS. The information set $\{\hat{\Omega}, L, L_c\}$ corresponds to a multi-level PSI model, in which $T = 2$, $S_1 = \hat{\Omega}$, $S_2 = \{1, ..., N\} \setminus \hat{\Omega}$, $\beta_1 = \hat{L}/N$, $\beta_2 = (N - \hat{L})/N$, $\alpha_1 \approx L_c/\hat{L}$, $\alpha_2 \approx (L - L_c)/(N - \hat{L})$.

Example 2.2 (Real-time dynamic magnetic resonance image reconstruction problem)**.** Consider real-time dynamic magnetic resonance (MR) image reconstruction using the 2-D DWT as the sparsifying basis. For an image sequence, let $(\mathbf{Z})_{n_1\times n_1}$ denote the image at the current time, and $N = n_1^2$ be its dimension. Let $\mathbf{X}$ denote the 2D DWT of $\mathbf{Z}$, i.e. $\mathbf{X} = \mathbf{W}\mathbf{Z}\mathbf{W}^T$, where $\mathbf{W}$ is the DWT matrix. Let $\mathbf{Y} = \mathbf{F}\mathbf{Z}\mathbf{F} = \mathbf{F}\mathbf{W}^T\mathbf{X}\mathbf{W}\mathbf{F}$ be the 2D DFT of $\mathbf{Z}$, and $\mathbf{F}$ is the DFT matrix. Using Kronecker product to convert the above to a 1-D problem, we can get

$$\mathbf{y}_{full} = \mathbf{F}_{1D}\mathbf{W}_{1D}^T\mathbf{x}^\star, \tag{2.2.4}$$

where $\mathbf{y}_{full} = vec(\mathbf{Y})$, $\mathbf{F}_{1D} = \mathbf{F} \otimes \mathbf{F}$, $\mathbf{W}_{1D}^T = \mathbf{W}^T \otimes \mathbf{W}^T$, and $\mathbf{x}^\star = vec(\mathbf{X})$. In MR, a smaller number, $M < N$, of Fourier coefficients of the image will be captured. This can be modeled by applying an $M \times N$ mask $\mathbf{G}$ to $\mathbf{y}_{full}$ to get the observation $\mathbf{y}$. The mask matrix $\mathbf{G}$ only contains a single 1 at a different location in each row and all other entries are zero. In certain situations, the MR measurements are noisy, then (2.2.4) can be rewritten as

$$\mathbf{y} = \mathbf{A}\mathbf{x}^\star + \mathbf{n}, \tag{2.2.5}$$

where $\mathbf{A} = \mathbf{G}\mathbf{F}_{1D}\mathbf{W}_{1D}^T$ and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. Note that natural images, e.g., cross-sectional images of human organs are usually piece-wise smooth, the wavelet transform domain of these images, $\mathbf{x}^\star$, is sparse [4]. Therefore, (2.2.5) is the CS model in (2.1.1) with measurement matrix $\mathbf{A}$ and sparse signal $\mathbf{x}^\star$. Let Ω denote the support of $\mathbf{x}^\star$. It's empirically observed that the support set of image's wavelet transform changes very slowly with time, due to the temporal dependencies [4]. Thus, we can obtain some prior information about current support Ω from previously estimated images. Specifically, we can obtain a support estimation $\hat{\Omega}$, the error in the known part of support $\Delta_e = \hat{\Omega} \cap \Omega_c$ and the unknown part of the support $\Delta = \Omega \cap \hat{\Omega}_c$ [4, 42], where $\Omega_c = \{1, ..., N\} \setminus \Omega$, $\hat{\Omega}_c = \{1, ..., N\} \setminus \hat{\Omega}$. For example, $\hat{\Omega}$ could be the estimated support of the wavelet coefficients' vector of the previous image. This prior information set $\{\hat{\Omega}, \Delta_e, \Delta\}$ corresponds to a multi-level PSI model, in which $T = 2$, $S_1 = \hat{\Omega}$, $S_2 = \hat{\Omega}_c$, $\beta_1 = |\hat{\Omega}|/N$, $\beta_2 = (N - |\hat{\Omega}|)/N$, $\alpha_1 = (|\hat{\Omega}| - |\Delta_e|)/|\hat{\Omega}|$, $\alpha_2 = |\Delta|/(N - |\hat{\Omega}|)$.

2.2.3 Weighted LASSO Algorithm

Based on the multi-level PSI $\mathbb{I}$, we consider the weighted LASSO algorithm with weighted l_1 norm as the regularization function. The weighted LASSO algorithm obtains the estimated sparse signal $\hat{\mathbf{x}}$ from the linear measurement model in (2.1.1) by solving the following weighted LASSO problem:

$$\hat{\mathbf{x}} := \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \sigma||\mathbf{x}||_{1,\mathbf{w}}, \tag{2.2.6}$$

where the multi-level PSI is incorporated in the weight vector $\mathbf{w}$. Specifically, for each PSI component set, we assign a weight $w_t \geq 0$, and choose the weight vector $\mathbf{w} \in \mathbb{R}^N$ according

to

$$\mathbf{w} = \sum_{t=1}^{T} w_t 1_{S_t}(\mathbf{n}), \tag{2.2.7}$$

where $1_{S_t}(\cdot)$ is the indicator function such that for each element $i \in \mathbf{n}$

$$1_{S_t}(i) = \begin{cases} 1 & \text{if } i \in S_t \\ 0 & \text{else} \end{cases}. \tag{2.2.8}$$

In the following section, we will analyze the closed-form performance of the weighted LASSO algorithm to investigate how the weights $\mathbf{w}$ and PSI $\mathbb{I}$ will affect the performance, based on which, the optimal weight vector $\mathbf{w}$ will be given.

2.3 Closed-form Performance Analysis of Weighted LASSO

In this section, we shall apply the CGMT approach [56] to derive the minimum aNSE with the optimal LASSO weight vector $\mathbf{w}^\star$ (minimized over the weight vector $\mathbf{w}$). Specifically, we first formally define the aNSE and another important concept called *Average Gaussian Distance*. Then we show that the aNSE for a given LASSO weight vector $\mathbf{w}$ can be expressed using the average Gaussian distance. After that, we derive the optimal LASSO weight vector and the associated minimum aNSE, which is expressed using the minimum average Gaussian distance. We also discuss how to calculate the minimum average Gaussian distance and obtain its accurate and closed-form approximations and the corresponding minimum aNSE. Finally, based on the above results, we obtain stable recovery condition of the proposed algorithm (i.e., the minimum number of measurements required to achieve stable recovery).

2.3.1 Definitions of aNSE and Average Gaussian Distance

In this section, we first give some definitions which will be used to measure the performance and express the result. Let $\hat{\mathbf{x}}$ denote the estimate of $\mathbf{x}^\star$ obtained by solving the weighted LASSO problem (2.2.6). The *normalized squared error* (NSE) of (2.2.6) with parameter $\mathbf{w}$ is defined as

$$\mathrm{NSE}(\mathbf{w}, \sigma) \triangleq \frac{||\hat{\mathbf{x}} - \mathbf{x}^\star||_2^2}{\sigma^2}, \tag{2.3.1}$$

and aNSE is defined as

$$\mathrm{aNSE}(\mathbf{w}) \triangleq \lim_{\sigma \to 0} \mathrm{NSE}. \tag{2.3.2}$$

aNSE has been widely used as an important performance metric for performance analysis of noisy CS problem [7, 52, 57–59]. aNSE can be interpreted as the coefficient of the first order Taylor expansion of the squared error (SE) as explained below. Let $SE\left(\mathbf{w}, \sigma^2\right) \triangleq \left\|\hat{\mathbf{x}} - \mathbf{x}\right\|^2$ denote the squared error of the weighted LASSO formulation. The first order Taylor expansion of $SE\left(\mathbf{w}, \sigma^2\right)$ w.r.t. σ^2 is given by

$$\begin{aligned} SE\left(\mathbf{w}, \sigma^2\right) &= SE\left(\mathbf{w}, 0\right) + \frac{\partial SE\left(\mathbf{w}, 0\right)}{\partial \sigma^2}\sigma^2 + o\left(\sigma^2\right) \\ &= \frac{\partial SE\left(\mathbf{w}, 0\right)}{\partial \sigma^2}\sigma^2 + o\left(\sigma^2\right). \end{aligned} \tag{2.3.3}$$

Note that in (2.3.3), we assume the number of measurements is sufficiently large for perfect recovery when there is no noise in the measurements, and thus $SE\left(\mathbf{w}, 0\right) = 0$. By the definition of aNSE, we have $aNSE\left(\mathbf{w}\right) = \frac{\partial SE(\mathbf{w},0)}{\partial \sigma^2}$. Therefore, although aNSE is defined in the asymptotic regime as $\sigma \to 0$ and itself does not contain the noise variance σ^2, it captures the first order effect of noise on the SE by characterizing how fast the first order term in the SE increases with the noise variance σ^2. It is conjectured in [52] that aNSE is the worst case NSE: $\mathrm{aNSE}(\mathbf{w}) = \sup_{\sigma > 0} \mathrm{NSE}(\mathbf{w}, \sigma)$, which highlights the significance of studying the aNSE. In practice, aNSE can be used to tune the algorithm parameters to minimize the worst case reconstruction error over unknown noise variance. Moreover, simulation results in [7, 52] show that

$$\mathrm{MSE} \triangleq \frac{1}{N}||\hat{\mathbf{x}} - \mathbf{x}^\star||^2 \approx \mathrm{aNSE}(\mathbf{w})\sigma^2/N \tag{2.3.4}$$

at moderate and high SNR. Therefore, it's very important to analyze the aNSE performance.

In the following, we will give the definition of *Average Gaussian Distance*, which is the key step to analyze the aNSE using CGMT approach.

For any weight vector $\mathbf{z} \in \mathbb{R}^N$ satisfying $\mathbf{z} = \sum_{t=1}^{T} z_t 1_{S_t}(\mathbf{n})$ for some $z_t \geq 0, t = 1, ..., T$, let $\partial \left\|\mathbf{x}^\star\right\|_{1,\mathbf{z}}$ denote the subdifferential set of weighted l_1 norm $\left\|\mathbf{x}\right\|_{1,\mathbf{z}} = \sum_{i=1}^{N} \mathbf{z}[i] \left|\mathbf{x}[i]\right|$ at $\mathbf{x}^\star$. Then the *Average Gaussian Distance* to the $\partial \left\|\mathbf{x}^\star\right\|_{1,\mathbf{z}}$ with weight vector $\mathbf{z}$ is defined as

$$D(\mathbf{z}) \triangleq \frac{1}{N}\mathbb{E}_{\mathbf{h}}[\mathrm{dist}^2(\mathbf{h}, \partial \left\|\mathbf{x}^\star\right\|_{1,\mathbf{z}})], \tag{2.3.5}$$

where $\mathbf{h} \in \mathbb{R}^N$ has i.i.d. $\mathcal{N}(0, 1)$ entries, and $\mathrm{dist}(\mathbf{h}, \partial \left\|\mathbf{x}^\star\right\|_{1,\mathbf{z}})$ is the Gaussian distance for

fixed $\mathbf{h}$ defined as

$$\begin{aligned}\operatorname{dist}(\mathbf{h}, \partial \left\|\mathbf{x}^{\star}\right\|_{1,\mathbf{z}}) &= ||\mathbf{h} - \partial \left\|\mathbf{x}^{\star}\right\|_{1,\mathbf{z}}|| \\ &\triangleq \min_{\mathbf{s}\in\partial\|\mathbf{x}^{\star}\|_{1,\mathbf{z}}} ||\mathbf{h}-\mathbf{s}||. \end{aligned} \tag{2.3.6}$$

Note that we need to define the average Gaussian distance for any weight vector $\mathbf{z} \in \mathbb{R}^N$ satisfying $\mathbf{z} = \sum_{t=1}^{T} z_t 1_{S_t}(\mathbf{n})$, because the aNSE of the weighted LASSO with weight $\mathbf{w}$ is expressed using the average Gaussian distance for another weight vector $\mathbf{z}$ that depends on but not equal to $\mathbf{w}$.

The average Gaussian distance $D(\mathbf{z})$ for weighted l_1 norm can be calculated by the following Lemma.

Lemma 2.1 (Average Gaussian Distance of Weighted l_1 norm)**.** *Let $M, N \to \infty$ with $\frac{M}{N} \to \delta \in (0, \infty)$, and a partition $\{S_t\}_{t=1}^{T}$ is given with parameters $\frac{|\Omega \cap S_t|}{|S_t|} \to \alpha_t \in (0, 1]$, $\frac{|S_t|}{N} \to \beta_t \in (0, 1]$. The average Gaussian distance to the scaled subdifferential of the weighted l_1 norm defined in (2.3.5) can be calculated as follows:*

$$D(\mathbf{z}) = \sum_{t=1}^{T} \beta_t d(z_t), \tag{2.3.7}$$

where

$$d(z_t) = \alpha_t \left(1 + z_t^2\right) + (1 - \alpha_t) \left(2\left(1 + z_t^2\right) Q\left(z_t\right) - \sqrt{\frac{2}{\pi}} z_t e^{-\frac{z_t^2}{2}}\right), \tag{2.3.8}$$

and $Q(\cdot)$ is the Q function defined as $Q\left(x\right) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(-\frac{u^2}{2}\right) du$.

Please refer to Appendix 2.7.1 for the proof.

Remark 2.1. Even though in the definition of average Gaussian distance in (2.3.5), it is related to the actual sparse signal $\mathbf{x}^{\star}$. However, as shown in Appendix 2.7.1, after calculating the sub-differential of weighted ℓ_1 norm and the minimal distance for fixed vector in (2.3.6), and taking the expectation w.r.t. the i.i.d. Gaussian vector $\mathbf{h}$, the resulting average Gaussian distance only depends on the PSI parameters $\mathbb{I} = \left\{K, \{S_t, \alpha_t, \beta_t\}_{t=1}^{T}\right\}$ as shown in (2.3.7) and (2.3.8). For any sparse signal with same PSI parameters $\mathbb{I}$, the average Gaussian distance D shares the same form.

In the next section, we will show that the aNSE for a given LASSO weight vector $\mathbf{w}$ can

be expressed using the average Gaussian distance.

2.3.2 aNSE for Given LASSO Weight Vector w

For the statement of our result, we define a scalar function of vector $\mathbf{z}$, $G(\mathbf{z}) \triangleq \sum_{t=1}^{T} \beta_t g(z_t)$, where $g(z_t) = -(z_t/2)\partial d(z_t)/\partial z_t$ is a scaled product of z_t and $\partial d(z_t)/\partial z_t$, and $d(z_t)$ is defined in (2.3.8). We define the map as follows:

Definition 2.1 (map)**.** Let $\mathcal{R}_+^N \coloneqq \left\{\mathbf{z} \in \mathbb{R}_+^N | \delta - D(\mathbf{z}) > \max\{0, G(\mathbf{z})\}\right\}$ and define the map Λ: $\mathcal{R}_+^N \to \mathbb{R}_+^N$ as

$$\Lambda(\mathbf{z}) \coloneqq \mathbf{z}\frac{\sqrt{N}\left(\delta - D(\mathbf{z}) - G(\mathbf{z})\right)}{\sqrt{\delta - D(\mathbf{z})}}. \tag{2.3.9}$$

Similar to Lemma 2.1 in [52], if $\delta > \min_{\mathbf{z}} D(\mathbf{z})$, then the inverse mapping $\Lambda^{-1} : \mathbb{R}_+^N \to \mathcal{R}_+^N$ is well defined. Then we have the following theorem to characterize the behavior of the aNSE in the large system setup.

Theorem 2.1 (aNSE of weighted LASSO)**.** *Suppose the measurement matrix* $\mathbf{A}$ *has i.i.d. Gaussian entries with zero mean and unit variance. Fix any weight vector* $\mathbf{w}$ *in (2.2.6), let* $M, N \to \infty$ *with* $\frac{M}{N} \to \delta \in (0, \infty)$*, the following equality holds with probability 1:*

$$aNSE(\mathbf{w}) = \frac{D\left(\Lambda^{-1}(\mathbf{w})\right)}{\delta - D\left(\Lambda^{-1}(\mathbf{w})\right)}. \tag{2.3.10}$$

Please refer to Appendix 2.7.2 for the proof.

2.3.3 Optimal Tuning of LASSO Weights and Minimum aNSE

In the following Corollary, we give the optimal choice of LASSO weights which minimize the aNSE in Theorem 2.1, and the corresponding minimum aNSE.

Corollary 2.1 (Optimal LASSO weights and minimum aNSE)**.** *Suppose the measurement matrix* $\mathbf{A}$ *has i.i.d. Gaussian entries with zero mean and unit variance. Let* $M, N \to \infty$ *with* $\frac{M}{N} \to \delta \in (0, \infty)$*,* $D(\mathbf{z}^\star) \to D^\star \in (0, \delta)$ *where* $\mathbf{z}^\star = \arg\min_{\mathbf{z}} D(\mathbf{z})$*, and a partition* $\{S_t\}_{t=1}^T$ *is given with parameters* $\frac{|\Omega \cap S_t|}{|S_t|} \to \alpha_t \in (0, 1]$*,* $\frac{|S_t|}{N} \to \beta_t \in (0, 1]$*. Then the optimal LASSO weight vector that minimizes the aNSE is given by*

$$\mathbf{w}^\star = \mathbf{z}^\star \sqrt{N(\delta - D^\star)}. \tag{2.3.11}$$

The corresponding minimum aNSE is given by

$$aNSE^{\star}(\mathbf{w}^{\star}) = \frac{D^{\star}}{\delta - D^{\star}}, \tag{2.3.12}$$

Proof. According to (2.3.10), we have $\mathbf{z}^{\star} = \Lambda^{-1}(\mathbf{w}^{\star})$. Because $G(\mathbf{z}^{\star}) = 0$, we have $\mathbf{w}^{\star} = \Lambda(\mathbf{z}^{\star}) = \mathbf{z}^{\star}\sqrt{N(\delta - D(\mathbf{z}^{\star}))}$. Combining this with Theorem 2.1, we can conclude the minimum aNSE as in (2.3.12). □

Remark 2.2. To make sure $\delta > D^{\star}$ and the denominator of (2.3.12) is positive, the number of measurements should be larger than a critical point $ND^{\star}$. In this case, the robust recovery can be guaranteed in the noisy case.

In the following, we will calculate the minimum average Gaussian distance and obtain its accurate and closed-form approximation and the corresponding minimum aNSE.

Due to the separable property of $D(\mathbf{z})$, we have

$$D(\mathbf{z}^{\star}) = \sum_{t=1}^{T} \beta_t d(z_t^{\star}), \tag{2.3.13}$$

where $z_t^{\star} = \arg\min_{z_t} d(z_t)$, which depends on the parameter α_t. The minimum average Gaussian distance in (2.3.13) is expressed as a complicate function of $z_t^{\star}$, which is the optimal solution of an optimization problem. In the following, we derive an accurate and closed-form approximation for the minimum average Gaussian distance to simplify the calculation. The closed-form approximation also provides more useful insight about how the key PSI/system parameters affect the performance of the proposed optimally-tuned weighted LASSO algorithm. In particular, it facilitates the PSI quality analysis given in Section 2.4.

We first give a closed-form upper bound of $D(\mathbf{z}^{\star})$ in the following lemma.

Lemma 2.2. *For given $\{\alpha_t\}$ and $\{\beta_t\}$, the minimum average Gaussian distance $D(\mathbf{z}^{\star})$ can be upper bounded as*

$$D(\mathbf{z}^{\star}) \leq \sum_{t=1}^{T} \beta_t \alpha_t \left(2\log\left(\frac{1}{\alpha_t}\right) + 3\right). \tag{2.3.14}$$

Please refer to Appendix 2.7.3 for the proof. Although the upper bound in (2.3.14) is not tight, it characterizes the key features of the minimum average Gaussian distance, as shown in Fig. 2.1. Motivated by the upper bound in (2.3.14), we propose an accurate closed-form

approximation of the $d(z_t^\star)$ as

$$d(z_t^\star) \approx \widetilde{d^\star} = \alpha_t \left(c_1 \log\left(\frac{1}{\alpha_t}\right) + c_2 \right), \tag{2.3.15}$$

where c_1 and c_2 are obtained as follows. Note that the $d(z_t^\star)$ only depends on α_t. Therefore, we can express $d(z_t^\star)$ as a function of α_t as $d(z_t^\star) = f(\alpha_t) = \min_{z_t} \alpha_t (1 + z_t^2) + (1 - \alpha_t) \left(2(1 + z_t^2) Q(z_t) - \sqrt{\frac{2}{\pi}} z_t e^{-\frac{z_t^2}{2}} \right)$. Since $\alpha_t \in (0, 1]$, we can choose c_1 and c_2 to minimize the overall approximation error over the range $\alpha_t \in (0, 1]$ as

$$(c_1, c_2) = \underset{c_1', c_2'}{\text{argmin}} \sum_{i=1}^{N} \left| f\left(\frac{i}{N}\right) - \left(c_1' \log\left(\frac{N}{i}\right) + c_2' \right) \frac{i}{N} \right|^2.$$

The results are given by $c_1 = 0.9554$, $c_2 = 1.0033$. With these optimal coefficients, the approximation in (2.3.15) is very close to $d(z_t^\star)$ for all possible α_t values[5], as shown in Fig.2.1. Therefore, the minimum average Gaussian distance of the weighted l_1 norm can be accurately approximated by the following closed-form formula:

$$D(\mathbf{z}^\star) \approx \widetilde{D^\star} = \sum_{t=1}^{T} \beta_t \alpha_t \left(0.9554 \times \log\left(\frac{1}{\alpha_t}\right) + 1.0033 \right). \tag{2.3.16}$$

In the rest of the chapter, we will fix $c_1 = 0.9554$, $c_2 = 1.0033$.

From Corollary 2.1 and the closed-form approximation of $D(\mathbf{z}^\star)$ in (2.3.16), the minimum aNSE can be accurately approximated by the following closed-form:

$$\widetilde{aNSE}^\star = \frac{\widetilde{D^\star}}{\delta - \widetilde{D^\star}} = \frac{\sum_{t=1}^{T} \beta_t \alpha_t \left(c_1 \log\left(\frac{1}{\alpha_t}\right) + c_2 \right)}{\delta - \sum_{t=1}^{T} \beta_t \alpha_t \left(c_1 \log\left(\frac{1}{\alpha_t}\right) + c_2 \right)}. \tag{2.3.17}$$

2.3.4 Discussions

Fig. 2.2 illustrates the performance gain of the optimally-tuned weighted LASSO compared to the standard LASSO [53]. For illustration purposes, we consider two-level PSI with two PSI component sets S_1 and S_2 whose size parameters are $\beta_1 = 0.1$ and $\beta_2 = 0.9$, respectively. We can see that the theoretical aNSE prediction matches the simulated aNSE quite well ($\circ$

[5]Even though the approximation of the minimum average Gaussian distance inspired by the form of upper bound is very accurate, how to obtain its accurate closed form theoretically would be a challenging problem and will be left as part of future work.

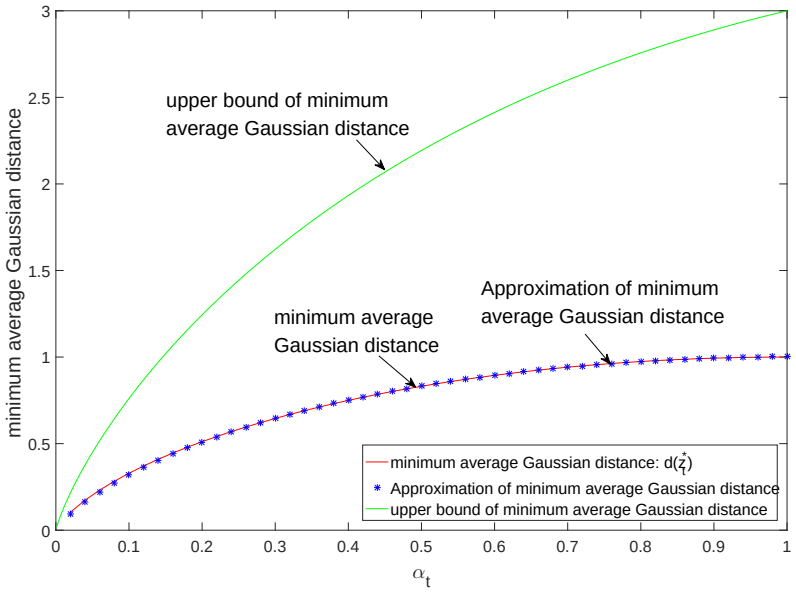

Figure 2.1: Comparison of minimal average Gaussian distance and its closed-form approximation in (2.3.15).

for standard LASSO, $*$ for optimally-tuned weighted LASSO) under moderate SNR and system dimensions, even though the conclusions in Corollary 2.1 are derived under asymptotic regime. Moreover, from Corollary 2.1 and Fig. 2.2, we have the following observations.

2.3.4.1 Phase Transition and Critical Measurements Number

As shown in (2.3.12), the aNSE of the optimally-tuned weighted LASSO exhibits a phase transition behavior. Specifically, when the measurements number M is larger than the critical point $\hat{M} \triangleq ND^\star$, the aNSE is finite and stable recovery is guaranteed. When the measurements number is approaching the critical point $\hat{M}$, the aNSE grows to infinity and it is impossible to achieve stable recovery. Also, from the separability of the minimum Gaussian distance in (2.3.13), the critical measurements number $\hat{M}$ of the optimally-tuned weighted LASSO can be expressed as $\hat{M} = \sum_{t=1}^{T} \hat{M}_t$, where

$$\hat{M}_t = N\beta_t d(z_t^\star) \approx N\beta_t\alpha_t \left(c_1 \log\left(\frac{1}{\alpha_t}\right) + c_2 \right)$$

is the critical measurements number of standard LASSO to recover a $N\beta_t\alpha_t$ sparse signal with dimensions $N\beta_t$.

Compared to the standard LASSO, the critical measurements number of the optimally-tuned weighted LASSO with PSI is much smaller, and it decreases as the PSI becomes more

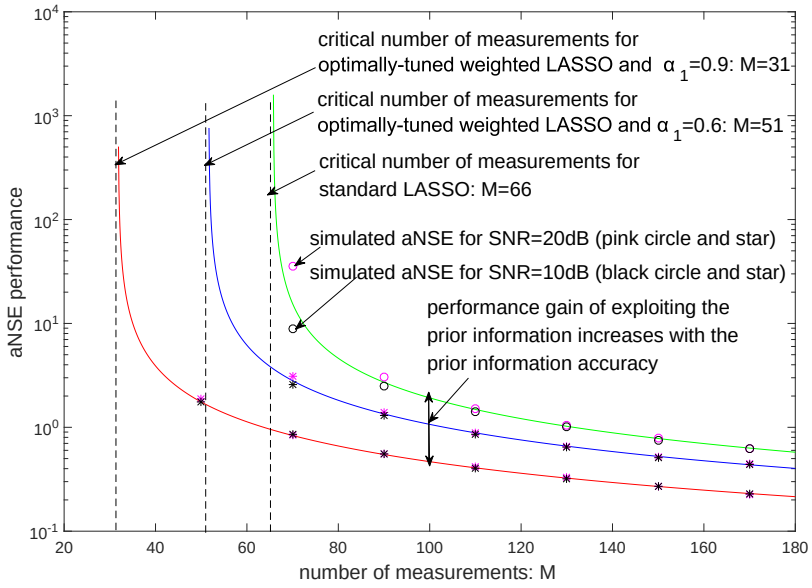

Figure 2.2: aNSE performance for the optimally-tuned weighted LASSO and standard LASSO. Consider $N = 200$, the sparsity level $K = 20$, and there are two partitions S_1 and S_2 with parameters $\beta_1 = 0.1$ and $\beta_2 = 0.9$, respectively. For the proposed algorithm, we consider two cases when $\alpha_1 = 0.9$ and $\alpha_1 = 0.6$, respectively. The simulated aNSE is given by simulated NSE with SNR=20 dB for pink circle and star, with SNR=10dB for black circle and star.

accurate. For example, it can be seen from Fig. 2.2 that the critical measurements number of the standard LASSO is 66, and it is much larger than that of the optimally-tuned weighted LASSO, which is about 51 when $\alpha_1 = 0.6$. When the PSI is more accurate, e.g., $\alpha_1 = 0.9$, the critical measurements number decreases to 31. Therefore, the PSI-aided optimally-tuned weighted LASSO significantly alleviates the number of measurements required.

2.3.4.2 Optimal LASSO Weights

The optimal weights $\mathbf{w}^\star$ described in (2.3.11) depend on $\mathbf{z}^\star$, whose elements can be calculated through the following equation:

$$(1-\alpha_t)\left(\sqrt{\frac{2}{\pi}}e^{-\frac{z_t^{\star 2}}{2}} - 2z_t^\star Q(z_t^\star)\right) = \alpha_t z_t^\star, \tag{2.3.18}$$

which is obtained by setting the differential of $d(z_t)$ with respect to z_t to be 0. Given the partitions S_1 and S_2, Fig. 2.3 shows the value of optimal weights w_1 and w_2 when PSI accuracy α_1 increases from 0 to 1. We can see that as α_1 increases, optimal weight w_1 decreases and w_2 increases. When PSI component set S_1 has perfect prior support information, i.e., $\alpha_1 = 1$,

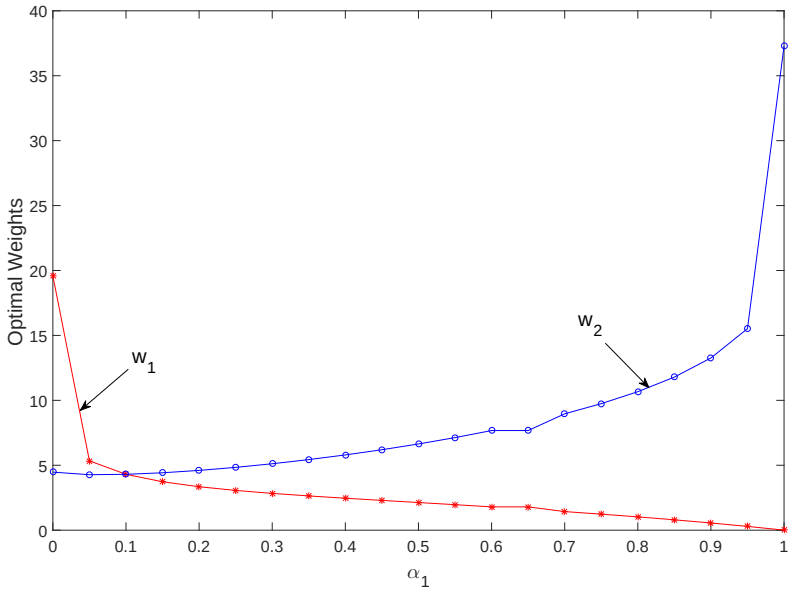

Figure 2.3: Optimal weights versus PSI accuracy α_1 for the optimally-tuned weighted LASSO. Consider $N = 200$, the sparsity level $K = 20$. And there are two partitions S_1 and S_2 with parameters $\beta_1 = 0.1$, $\beta_2 = 0.9$, respectively. The number of measurements is 80. The simulated aNSE is given by simulated NSE with SNR=20 dB.

the optimal weight $w_1 = 0$. But when $0 < \alpha_1 < 1$, the weight vector $\mathbf{w}$ should be optimally tuned to minimize the recovery error bound aNSE. Intuitively, we should assign a smaller weight w_t to the t-th PSI component set S_t when the PSI for this set is more accurate (i.e., α_t is larger), as verified in Fig, 2.3. When $\alpha_1 = \alpha_2 = K/N$, the optimal weights $w_1 = w_2$ and the optimally-tuned weighted LASSO reduces to the standard LASSO.

As shown in (2.3.11) and (2.3.17), the optimal weights and minimum aNSE all depend on the PSI parameters $\{\alpha_t, \beta_t\}_{t=1}^{T}$. In the following section, we will discuss how the performance of the optimally-tuned weighted LASSO will be affected by the PSI qualities.

2.3.4.3 The Noise Effect in the Proposed Algorithm

The noise effect is taken into account in the proposed algorithm as explained in the following aspects:

- **Problem formulation:** The weighted LASSO problem in (2.2.6) considers the noise effect and can be used to recover the sparse signal from noisy measurement.

- **Performance analysis**: The aNSE analyzed in our work captures the first order effect of noise on the squared error and it is widely used as a key performance metric

in the noisy case. Although aNSE is defined in the asymptotic regime as $\sigma \to 0$ and itself does not contain the noise variance σ^2, it captures the first order effect of noise on the squared error by characterizing how fast the first order term in the squared error increases with the noise variance σ^2, i.e. $SE(\mathbf{w}, \sigma^2) = aNSE(\mathbf{w})\sigma^2 + o(\sigma^2)$. Therefore, analyzing the asymptotic performance doesn't mean that the noise effect is ignored. On the contrary, we have captured the first order term of $SE(\mathbf{w}, \sigma^2)$ with respect to σ^2.

- **Determination of the optimal weight:** In our work, the optimal normalized weight $\mathbf{w}^\star$ obtained by minimizing the aNSE already contains the effect of noise in an implicit way, and the optimal actual weight $\sigma\mathbf{w}^\star$ explicitly contains the standard deviation of the noise σ. Specifically, minimizing aNSE is equivalent to minimizing the first order Taylor expansion of $SE(\mathbf{w}, \sigma^2)$ for any finite $\sigma > 0$, and first order Taylor expansion of SE is an upper bound of SE [17]. Therefore, minimizing aNSE can be viewed as minimizing the worst case SE over unknown noise variance. Although the optimal normalized weight $\mathbf{w}^\star$ derived by minimizing the aNSE performance does not contain the noise variance, it still incorporates the first order effect of noise by characterizing how fast the actual weight $\sigma\mathbf{w}$ should increase with the standard deviation of the noise σ in order to achieve the best aNSE performance for the weighted LASSO algorithm.

In summary, the optimal normalized weight vector derived in our paper is a result of incorporating the noise effect into our algorithm design and performance analysis by choosing proper problem formulation for the sparse signal recovery (i.e., the weighted LASSO formulation) and performance metric for optimizing the normalized weight vector (i.e., the aNSE performance). Simulations show that the proposed weighted LASSO algorithm with the optimal normalized weights achieves better performance than the OptWeight-ℓ_1-min algorithm in [46], which does not consider the effect of noise. This verifies that the proposed algorithm can indeed incorporate the noise effect and reduce the reconstruction error for the noisy case. Therefore, the noise effect plays a crucial role in the optimal weight vector in our work.

2.4 Impact of PSI Quality On the Performance

In this section, we analyze the impact of PSI quality on the performance of the optimally-tuned weighted LASSO algorithm. The PSI quality depends on how much information it can

provide about the position of the support. Specifically, when there are more partitions (larger T) and α_t's deviate more from the mean K/N (e.g., in each partition, α_t is either closer to 1 or much smaller than $\frac{K}{N}$), the PSI quality is higher. But when all α_t's are equal to the mean value K/N, we cannot obtain more prior information other than the sparsity level in the large system limit. Note that when α_t is smaller than K/N and close to 0, we can still have some prior information that the support is most likely outside the set S_t. The above intuition is formally proved in the following theorems.

2.4.1 Performance Comparison with Different Prior Information

We analyze the approximated minimum aNSE performance given in (2.3.17) in the following theorem which has been shown to be accurate by Fig. 2.1.

Theorem 2.2 (Performance of optimally-tuned weighted LASSO with more prior information)**.** *Given a partition $\mathcal{S} = \{S_t\}_{t=1}^{T}$ of the complete index set with parameters $\{\alpha_t, \beta_t\}_{t=1}^{T}$ defined in (2.2.1), we further divide the set S_t for $t \in \mathcal{T}$ into J_t partitions $\{S_{t,j}\}_{j=1}^{J_t}$, where $\mathcal{T} \subseteq [1, \cdots, T]$, and denote the set of parameters induced by the further partition as $\{\alpha_{t,j}, \beta_{t,j}\}_{j=1}^{J_t}$ for $t \in \mathcal{T}$ with the same definition as in (2.2.1). We denote the approximated minimum aNSE of the optimally-tuned weighted LASSO algorithm corresponding to the new partitions $\overline{\mathcal{S}} = \left\{ \left\{ \{S_{t,j}\}_{j=1}^{J_t} \right\}_{t\in\mathcal{T}}, \{S_t\}_{t\in\mathcal{T}_c} \right\}$ and the original partitions $\mathcal{S}$ as $\widetilde{aNSE}^{\star}(\overline{\mathcal{S}})$ and $\widetilde{aNSE}^{\star}(\mathcal{S})$, respectively, then we have the following result:*

$$\widetilde{aNSE}^{\star}(\overline{\mathcal{S}}) \leq \widetilde{aNSE}^{\star}(\mathcal{S}), \tag{2.4.1}$$

where the equality is obtained if and only if $\alpha_{t,j} = \alpha_t$, for all $j \in \{1, \cdots, J_t\}$ and $t \in \mathcal{T}$.

Please refer to Appendix 2.7.4 for the proof.

This theorem shows that when we have more prior information about where the support is located, the optimally-tuned weighted LASSO algorithm can perform better. Based on Theorem 2.2, we compare the approximated minimum aNSE performance of the optimally-tuned weighted LASSO and standard LASSO in the following corollary. The approximated minimum aNSE of standard LASSO is given by (2.3.17) with $T = 1, \beta_t = 1, \alpha_t = K/N$.

Corollary 2.2 (Comparison of the optimally-tuned weighted and standard LASSO)**.** *Given the PSI $\mathbb{I} = \left\{K, \{S_t, \alpha_t, \beta_t\}_{t=1}^{T}\right\}$, we denote the approximated minimum aNSE of the optimally-tuned weighted LASSO which exploits the PSI $\mathbb{I}$ as $\widetilde{aNSE}^{\star}(\mathbb{I})$, and denote the approximated*

minimum aNSE of the standard LASSO which does not exploit the PSI as $\widetilde{aNSE}_0^\star$*. Then we have the following result:*

$$\widetilde{aNSE}^\star(\mathbb{I}) \leq \widetilde{aNSE}_0^\star. \tag{2.4.2}$$

where the equality is obtained if and only if $\alpha_t = K/N$*, for all* $t \in \{1, \cdots, T\}$.

The proof of Corollary 2.2 is similar to the proof of Theorem 2.2. Please refer to Appendix 2.7.4 for the details.

Corollary 2.2 shows that the aNSE performance of the PSI-aided optimally-tuned weighted LASSO algorithm is always better than the standard LASSO except for the case when $\alpha_t = K/N$ for all t and the two algorithms achieve the same aNSE.

2.4.2 Examples

In the following, we use several examples to illustrate the above conclusions. Consider the case when $N = 200$, $M = 150$ and $K = 60$. We compare the minimum aNSE performance when different PSI is provided.

Firstly, we compare the aNSE performance of the optimally-tuned weighted LASSO with the standard LASSO when there are two partitions: $\mathcal{S} = \{S_1, S_2\}$ whose sizes are $\beta_1 = 0.3$ and $\beta_2 = 0.7$, respectively. Fig. 2.4 shows that the approximated minimum aNSE is very close to the theoretical minimum aNSE. Therefore, in the following examples, we only present the results for the theoretical minimum aNSE for compactness. Also, Fig. 2.4 shows that the minimum aNSE of the optimally-tuned weighted LASSO is not worse than the standard LASSO, and when $\alpha_1 = \alpha_2 = K/N$, they have same aNSE performance. This verifies the result in Corollary 2.2.

Next, we divide the first PSI component set S_1 in the partition $\mathcal{S} = \{S_1, S_2\}$ into two subsets and get a new partition $\overline{\mathcal{S}} = \{S_{1,1}, S_{1,2}, S_2\}$ with the parameters $\beta_{1,1} = 0.1, \beta_{1,2} = 0.2$ and $\beta_2 = 0.7$, respectively. Further partition induces two new parameters $\alpha_{1,1}$ and $\alpha_{1,2}$. We compare the minimum aNSE performance of the optimally-tuned weighted LASSO under two different partitions $\overline{\mathcal{S}}$ and $\mathcal{S}$, when $\alpha_{1,1}$ and $\alpha_{1,2}$ are changing, α_2 are fixed. It is shown that the performance of the optimally-tuned weighted LASSO with partition $\overline{\mathcal{S}}$ is always better than that with $\mathcal{S}$, and only when $\alpha_{1,1} = \alpha_{1,2} = \alpha_1$, they have the same performance. For example, the minimum aNSE of $\overline{\mathcal{S}}$ versus $\alpha_{1,1}$ and $\alpha_{1,2}$ when α_2 is fixed to $3/14$ is shown in Fig. 2.5 by the green curve. The minimum aNSE of $\mathcal{S}$ when $\alpha_1 = 0.5$ and $\alpha_2 = 3/14$ is

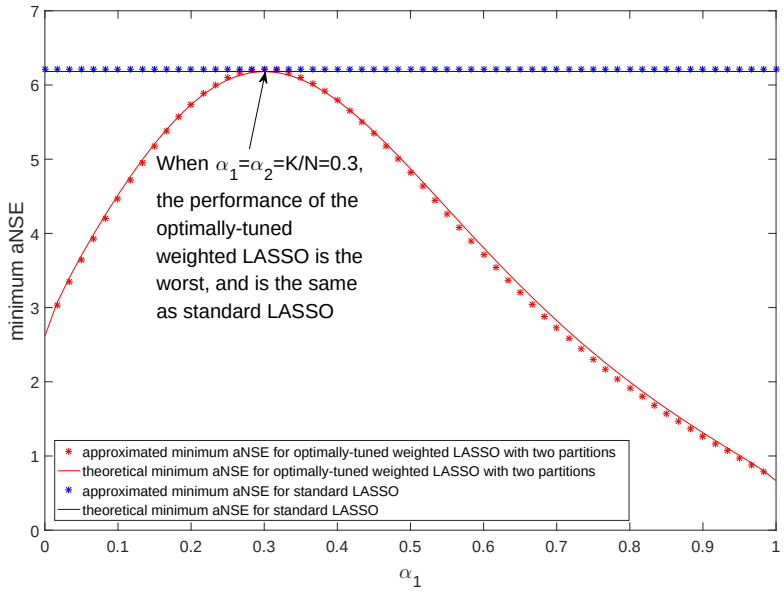

Figure 2.4: Minimum aNSE performance versus α_1 for the optimally-tuned weighted LASSO and standard LASSO. Consider $N = 200$, $K = 60$, $M = 150$, and 2 partitions with parameters $\beta_1 = 0.3$ and $\beta_2 = 0.7$.

shown in Fig. 2.5 by the red star, which is the intersection point of the green curve and the red star curve. It shows that when $\alpha_{1,1} \neq \alpha_{1,2}$, the minimum aNSE for partition $\overline{\mathcal{S}}$ is smaller than that for partition $\mathcal{S}$. When $\alpha_{1,1} = \alpha_{1,2} = 0.5$, the minimum aNSE for partition $\overline{\mathcal{S}}$ achieves the largest value, which equals the minimum aNSE for partition $\mathcal{S}$. Fig. 2.5 also shows the cases when α_2 is fixed to be other values. They all come to the same conclusion which is consistent with Theorem 2.2.

2.5 Simulation Results

In this section, we present the results of some numerical experiments to verify the practical performance of the proposed optimally-tuned weighted LASSO algorithm. Assume that $N = 200$, and the sparsity level $K = 20$. We consider the following baselines:

- Baseline 1 (Std. LASSO): This is a standard LASSO algorithm [**?**], which does not exploit the PSI, and l_1 norm is used as the penalty function.

- Baseline 2 (Modified-CS): The modified CS proposed in [47] incorporates the estimated PSI and minimizes the weighted l_1 norm with zero weight on the estimated support part.

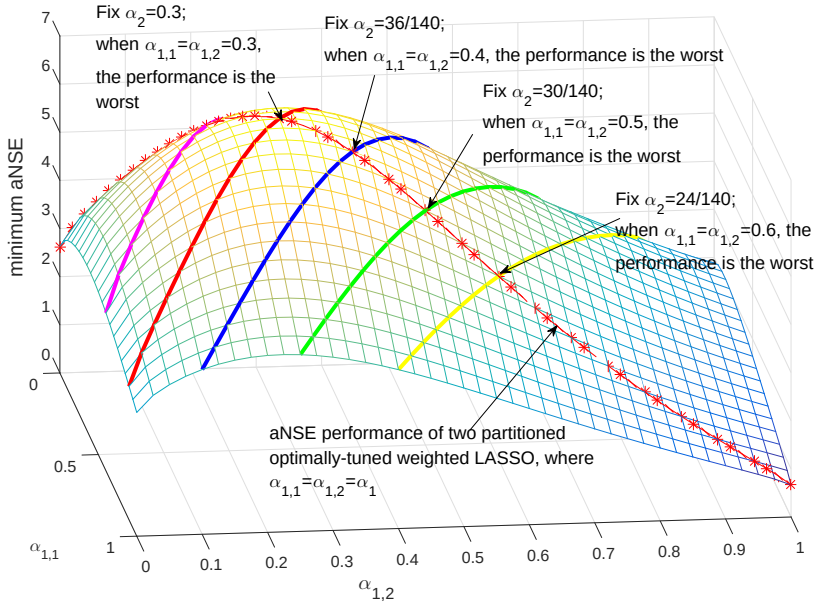

Figure 2.5: Minimum aNSE performance versus $\alpha_{1,1}$ and $\alpha_{1,2}$ for the optimally-tuned weighted LASSO. Consider $N = 200$, $K = 60$, $M = 150$, and 3 partitions with parameters $\beta_{1,1} = 0.1, \beta_{1,2} = 0.2$ and $\beta_2 = 0.7$.

- Baseline 3 (OptWeight-l_1-min): This is a naive extension of the optimally-tuned weighted l_1 norm minimization proposed in [46], where the weights are tuned to optimize the performance for the noiseless case. For the extension to the noisy case, we use the same weight strategy as in [46], but modify the linear constraint $\mathbf{y} = \mathbf{Ax}$ to be $||\mathbf{y} - \mathbf{Ax}|| \leq \epsilon$, where $\epsilon = \sqrt{M(1 + 0.1)\sigma^2}$ is determined by the noise variance σ^2.

- Baseline 4 (AdaptiveWeight-ℓ_1-min): This is an extension of the weighted ℓ_1 norm minimization proposed in [43] from the noiseless case to noisy case by changing the linear constraint $\mathbf{y} = \mathbf{Ax}$ to be $||\mathbf{y} - \mathbf{Ax}|| \leq \epsilon$, where $\epsilon = \sqrt{M(1 + 0.1)\sigma^2}$ is determined by the noise variance σ^2. We adopt the same weight policy proposed in [43].

2.5.1 Impact of Prior Information Accuracy On the Performance

We consider the case of two PSI component sets S_1 and S_2 with $\beta_1 = 0.1$ and $\beta_2 = 0.9$. We set M to be100, and the simulation SNR to be 20 dB. In Fig. 2.6, we compare the MSE performance of different algorithms versus the PSI accuracy α_1. For Modified-CS and AdaptiveWeight-ℓ_1-min baseline, set S_1 is considered as the given estimated support. It can be observed that the MSE of the proposed algorithm is increasing when α_1 is smaller than $K/N = 0.1$, and decreasing when α_1 is larger than $K/N = 0.1$. And when $\alpha_1 = K/N$,

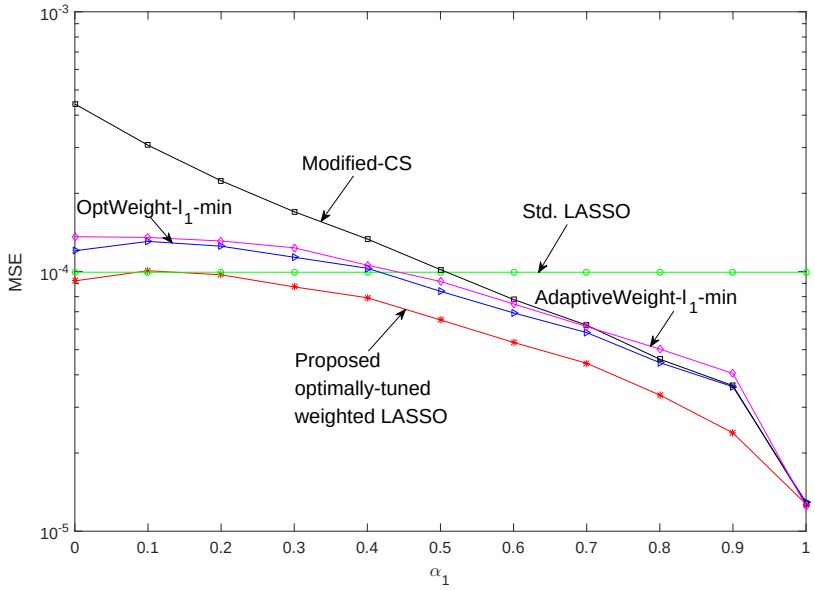

Figure 2.6: MSE versus PSI accuracy α_1 when there are two PSI component sets S_1 and S_2 with $\beta_1 = 0.1$ and $\beta_2 = 0.9$. Consider $N = 200$, $K = 20$, $M = 100$, and the simulation SNR is 20 dB.

the proposed optimally-tuned weighted LASSO performs the worst, and is the same as standard LASSO. This is consistent with the conclusion in Corollary 2.2. It also can be seen that the PSI-aided baselines outperform the standard LASSO when the PSI accuracy α_1 is larger than a critical point, which are 0.4, 0.43 and 0.5, respectively for the OptWeight-l_1-min, AdaptiveWeight-ℓ_1-min and Modified-CS baselines. When PSI accuracy is below these critical values, their respective performances are even worse than the standard LASSO. However, proposed algorithm never performs worse than the standard LASSO, and when α_1 is deviated from 0.1, the performance gain increases. When $\alpha_1 = 1$, the PSI-aided algorithms have same performance, the 0-1 weight strategy is optimal. Overall speaking, proposed optimally-tuned weighted LASSO is more robust to the PSI accuracy due to the optimal tuning of the LASSO weights. Also, our proposed algorithm achieves better performance compared to various baselines under any PSI accuracy.

2.5.2 Impact of Measurements Number on the Performance

Fig. 2.7 illustrates the MSE performance of different algorithms when the number of measurements changes. The simulation SNR is 20 dB. In order to see the impact of PSI quality on the performance, we consider two types of PSI:

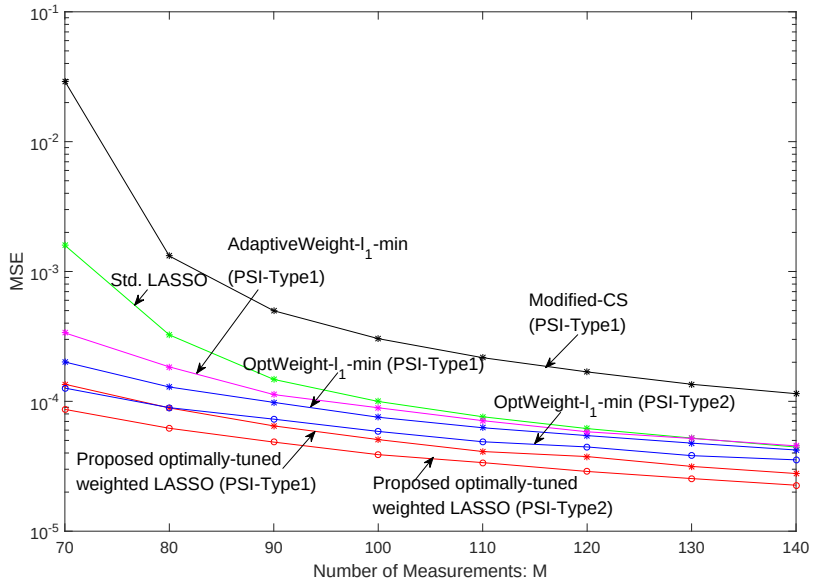

Figure 2.7: MSE versus measurements number. Consider $N = 200$, $K = 20$ and the simulation SNR is 20 dB and two types of PSI.

PSI-Type1: There are two partitions $\{S_1, S_2\}$ whose sizes are $\beta_1 = 0.3$ and $\beta_2 = 0.7$, respectively. The PSI accuracies in each PSI component set are $\alpha_1 = 0.3$ and $\alpha_2 = 1/70$, respectively.

PSI-Type2: There are four partitions $\{S_1, S_2, S_3, S_4\}$ whose sizes are $\beta_1 = 0.05, \beta_2 = 0.1, \beta_3 = 0.15$ and $\beta_4 = 0.7$, respectively. The PSI accuracies in each PSI component set are $\alpha_1 = 0.8, \alpha_2 = 0.4, \alpha_3 = 1/15$ and $\alpha_4 = 1/70$, respectively.

For the Modified-CS and AdaptiveWeight-ℓ_1-min baselines, set S_1 is considered as the given estimated support in PSI-Type1. It can be observed that the MSE decreases as the number of measurements increases. The proposed algorithm achieves the best performance among all of the schemes. Comparing the different types of PSI, the proposed optimally-tuned weighted LASSO and OptWeight-ℓ_1-min both can benefit from the high quality PSI PSI-Type2. This verifies the conclusion in Theorem 2.2. However, the proposed algorithm achieves better performance than OptWeight-ℓ_1-min for each PSI type.

2.5.3 Impact of SNR on the Performance

In Fig. 2.8, we compare the MSE performance of different schemes versus simulation SNR when $M = 80$. We also consider two types of PSI, as above. From this figure, we can see that

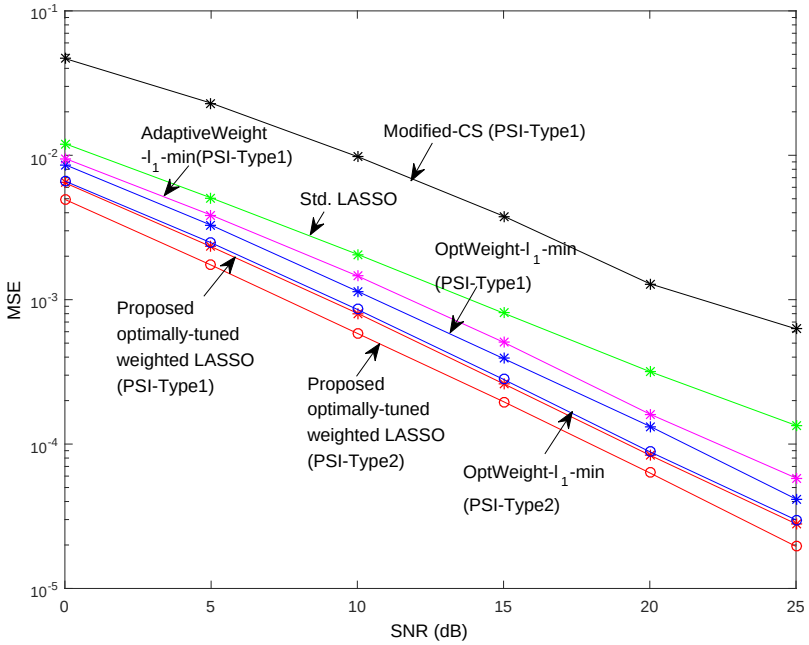

Figure 2.8: MSE versus simulation SNR. Consider $N = 200$, $K = 20$, $M = 80$, and two types of PSI.

the proposed algorithm achieves substantial performance gain compared to various baselines for each type of PSI. Also, when we have more prior information about where the support is situated, the proposed algorithm can perform better. This verifies the result in Theorem 2.2.

Remark 2.3 (MSE performance comparison with OptWeight-ℓ_1-min in [46]). In [46], the optimal weight is derived by minimizing the measurement threshold, i.e., the critical number of measurements required for success recovery of weighted ℓ_1 minimization problem with noiseless measurements, which may not be the optimal for the MSE performance. However, the proposed weight policy is optimal in the sense that it minimizes the aNSE performance, which provides a good approximation of the (scaled) MSE. As a result, even at high SNR regime (close to noiseless case), the proposed optimally-tuned weighted LASSO algorithm still performs better than the OptWeight-ℓ_1-min baseline.

2.5.4 Simulation Results for Massive MIMO Channel Estimation

In this subsection, we will present the numerical simulation results when the proposed optimally-tuned weighted LASSO is applied to the massive MIMO channel estimation problem, as shown by the Example 1 in Section 2.2.2. We assume that the BS has $N = 256$ antennas and transmits M training sequences for downlink channel estimation. The pilot matrix $\mathbf{P}$

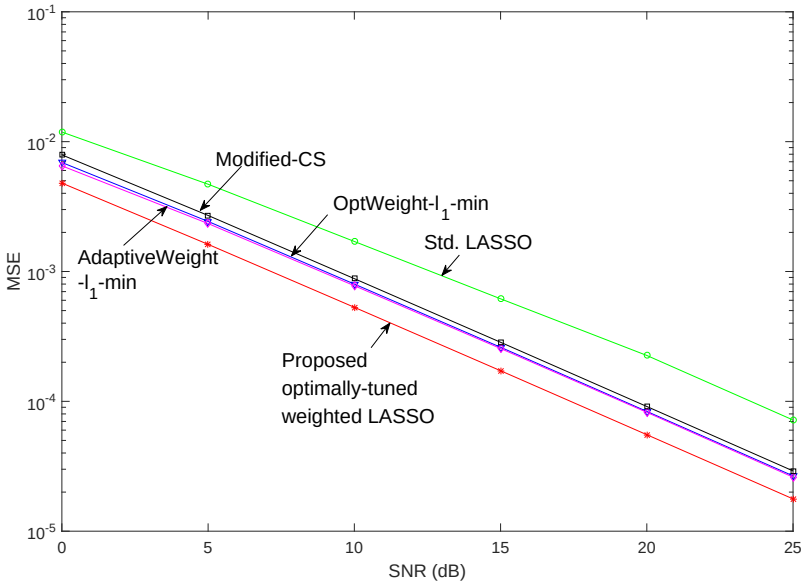

Figure 2.9: MSE versus simulation SNR. Consider $N = 256$ transmit antennas, $M = 100$ training sequences, $L = \hat{L} = 32$, $L_c = 24$.

is chosen to be i.i.d. Gaussian matrix. The resulting measurement matrix $\mathbf{A}$ is also i.i.d. Gaussian matrix. We set the channel support size equal to estimated channel support size, $L = \hat{L} = 32$, the common support size $L_c = 24$. This corresponds to the two-level PSI model, where $\alpha_1 \approx 0.75, \alpha_2 \approx 1/28, \beta_1 = 0.125, \beta_2 = 0.875$.

In Fig. 2.9 and Fig. 2.10, we compare the MSE performance of different schemes versus the simulation SNR and number of training sequences M. It can be seen that the proposed optimally-tuned weighted LASSO achieves better performance over various baselines under any SNR value and training sequences number.

2.6 Summary

We propose an optimally-tuned weighted LASSO algorithm to recover the compressed signal from a linear noisy model with statistical multi-level PSI and apply the proposed algorithm to massive MIMO channel estimation problem. By optimally tuning the LASSO weights according to the PSI accuracy in each PSI component set, the proposed algorithm can fully exploit the multi-level PSI to significantly improve the recovery performance and alleviate the requirement of the measurements number. We also analyze the performance of the proposed algorithm, and derive a closed-form, accurate bound on the recovery error, which helps

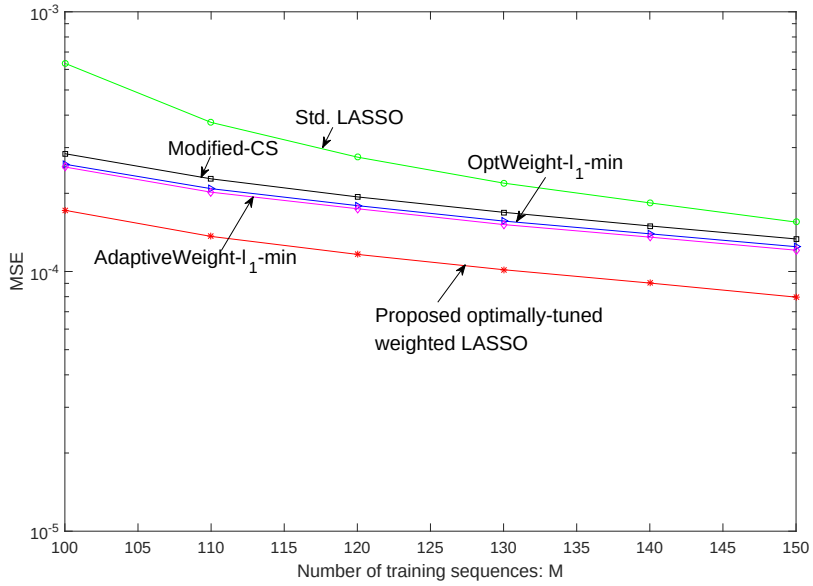

Figure 2.10: MSE versus number of training sequences M. Consider $N = 256$ transmit antennas, $L = \hat{L} = 32$, $L_c = 24$. Simulation SNR is15 dB.

to dimension the minimum number of measurements needed for stable recovery. Then we analyze the impact of PSI quality on the performance, which characterizes the type of PSI the proposed algorithm can benefit from. To the best of our knowledge, this is the first work to obtain the optimal LASSO weights and the associated closed-form performance analysis for PSI-aided LASSO algorithm in the presence of noise. Both the analysis and simulations show that the proposed optimally-tuned weighted LASSO algorithm outperforms the various baselines.

2.7 Appendix

2.7.1 Proof of Lemma 2.1

According to the definition in (2.3.5), for $i \in S_t, \forall t \in \{1, \cdots, T\}$,

$$\text{dist}\left(\mathbf{h}, \partial \left\|\mathbf{x}^\star\right\|_{1,\mathbf{z}}\right)_i = \begin{cases} \mathbf{h}[i] - z_t \cdot \text{sign}\left(\mathbf{x}^\star[i]\right) & \text{if } \mathbf{x}^\star[i] \neq 0 \\ \text{shrink}_{\mathbf{z_t}}\left(\mathbf{h}[i]\right) & \text{if } \mathbf{x}^\star[i] = 0 \end{cases}$$

where shrink function is defined as following:

$$\mathrm{shrink}_{\mathrm{z_t}}(h) = \begin{cases} h - z_t & \text{if } h > z_t \\ 0 & \text{if } |h| \leq z_t \\ h + z_t & \text{if } h < -z_t \end{cases}.$$

Consequently, after taking expectation, $\mathbb{E}\left[\left(\mathbf{h}[i] - z_t \cdot \mathrm{sign}\left(\mathbf{x}^\star[i]\right)\right)^2\right] = 1 + z_t^2$, and

$$\begin{aligned}\mathbb{E}\left[\left(\mathrm{shrink}_{\mathrm{z_t}}\left(\mathbf{h}[i]\right)\right)^2\right] &= \int_{z_t}^{\infty} \sqrt{\frac{2}{\pi}} \left(h - z_t\right)^2 e^{-\frac{h^2}{2}} dh \\ &= 2\left(1 + z_t^2\right) Q\left(z_t\right) - \sqrt{\frac{2}{\pi}} z_t e^{-\frac{z_t^2}{2}},\end{aligned}$$

where $Q(z_t) \triangleq \frac{1}{\sqrt{2\pi}} \int_{z_t}^{\infty} \exp\left(-\frac{u}{2}\right) du$. Therefore, the average Gaussian distance of weighted ℓ_1 norm is

$$\begin{aligned}D\left(\mathbf{z}\right) &= \frac{1}{N} \mathbb{E}_{\mathbf{h}}\left[\mathrm{dist}^2\left(\mathbf{h}, \partial \left\|\mathbf{x}^\star\right\|_{1,\mathbf{z}}\right)\right] \\ &= \sum_{t=1}^{T} \beta_t \alpha_t \left(1 + z_t^2\right) + \beta_t \left(1 - \alpha_t\right) \left(2\left(1 + z_t^2\right) Q\left(z_t\right) - \sqrt{\frac{2}{\pi}} z_t e^{-\frac{z_t^2}{2}}\right).\end{aligned}$$

2.7.2 Proof of Theorem 2.1

We rewrite the linear model (2.1.1) as $\mathbf{y} = \mathbf{A}\mathbf{x}^\star + \sigma\mathbf{v}$, where the entries of $\mathbf{v}$ are i.i.d. $\mathcal{N}(0, 1)$. Then we rewrite Problem (2.2.6) in a more convenient form for analysis,

$$\hat{\mathbf{e}} \coloneqq \underset{\mathbf{e}}{\mathrm{argmin}} \frac{1}{2}||\mathbf{A}\mathbf{e} - \mathbf{v}||^2 + \frac{1}{\sigma}||\mathbf{x}^\star + \sigma\mathbf{e}||_{1,\mathbf{w}}, \tag{2.7.1}$$

by changing the decision variable to be the normalized error vector $\mathbf{e} = (1/\sigma)(\mathbf{x} - \mathbf{x}^\star)$, substituting $\mathbf{y} = \mathbf{A}\mathbf{x}^\star + \sigma\mathbf{v}$ and rescaling the objective value by a factor of $1/\sigma^2$. Note that $||\hat{\mathbf{e}}||^2 = \mathrm{NSE}(\mathbf{w}, \sigma)$. We can handle Problem (2.7.1) by analyzing a different and simpler optimization problem based on the CGMT [56]. We summarize the result of the CGMT in the following lemma by following the similar proof to that of Lemma 1.2 in [17].

Lemma 2.3 (CGMT for Weighted LASSO)**.** *Suppose the measurement matrix* $\mathbf{A}$ *has i.i.d.*

Gaussian entries with zero mean and unit variance. Let

$$\tilde{f}(a,b) \coloneqq \sqrt{a^2+1}b - ab\sqrt{\frac{N}{M}D\left(\frac{1}{\sqrt{M}b}\mathbf{w}\right)} - \frac{b^2}{2}, \tag{2.7.2}$$

and $(a^\star, b^\star) \coloneqq \arg\min_{0\le a\le A}\max_{0\le b\le B}\ \tilde{f}(a,b)$*, where* A *and* B *are sufficiently large constants independent of* M, N*. Let* $\hat{\mathbf{e}}$ *be the minimizer of (2.7.1). As* $M, N \to \infty$ *with* $\frac{M}{N} \to \delta \in (0,\infty)$*, we have*

$$\lim_{\sigma\to 0}||\hat{\mathbf{e}}|| \xrightarrow{P} a^\star.$$

Based on Lemma 2.3, in a large system limit, the aNSE converges to minimizer $a^\star$ of the deterministic minimization in (2.7.2). To compute $a^\star$, we make use of the duality (the objective function is convex in a, and concave in b). We first fix b, and differentiate the objective $\tilde{f}(a,b)$ w.r.t. a and set it to 0 to find the minimizer $a^\star$, which is given by

$$a^\star = \sqrt{\frac{D\left(\frac{1}{\sqrt{M}b}\mathbf{w}\right)}{\delta - D\left(\frac{1}{\sqrt{M}b}\mathbf{w}\right)}}. \tag{2.7.3}$$

Substituting (2.7.3) back to (2.7.2), and setting the differential w.r.t. b to be 0, yield that

$$b^\star = \frac{\delta - D\left(\frac{1}{\sqrt{M}b^\star}\mathbf{w}\right) - G\left(\frac{1}{\sqrt{M}b^\star}\mathbf{w}\right)}{\sqrt{\delta\left(\delta - D\left(\frac{1}{\sqrt{M}b^\star}\mathbf{w}\right)\right)}}. \tag{2.7.4}$$

According to the definition of map Λ, the optimal $b^\star$ satisfies the following equation:

$$\Lambda^{-1}(\mathbf{w}) = \frac{1}{\sqrt{M}b^\star}\mathbf{w}.$$

Substituting the optimal $b^\star$ to the formula of $a^\star$, and based on Lemma 2.3, the aNSE can be expressed as (2.3.10) in a large system limit.

2.7.3 Proof of Lemma 2.2

Based on the Lemma H.1 in [57], we have

$$2\left(1+z_t^2\right)Q(z_t) - \sqrt{\frac{2}{\pi}}z_t e^{-\frac{z_t^2}{2}} \le \frac{2}{z_t^2+1}e^{-\frac{z_t^2}{2}}.$$

Therefore,

$$d(z_t) \leq \alpha_t(1+z_t^2) + (1-\alpha_t)\frac{2e^{-\frac{z_t^2}{2}}}{z_t^2+1} \leq \alpha_t(1+z_t^2) + 2e^{-\frac{z_t^2}{2}}.$$

Then

$$d(z_t^\star) \leq d\left(\sqrt{2\log\left(\frac{1}{\alpha_t}\right)}\right) \leq \alpha_t\left(2\log\left(\frac{1}{\alpha_t}\right)+3\right).$$

2.7.4 Proof of Theorem 2.2

The $\widetilde{aNSE}^\star$ in (2.3.17) is an increasing function of $\widetilde{D}^\star$ in (2.3.16), we can compare $\widetilde{D}^\star$for different prior information. We denote the approximated minimum Gaussian distance corresponding to partition $\overline{\mathcal{S}}$ and $\mathcal{S}$ as $\widetilde{D}^\star\left(\overline{\mathcal{S}}\right)$ and $\widetilde{D}^\star\left(\mathcal{S}\right)$, respectively. We denote $\widetilde{d}^\star$ in (2.3.15) as $\widetilde{d}^\star(\alpha_t)$, which is a function of α_t. Then

$$\widetilde{D}^\star\left(\mathcal{S}\right) = \sum_{t\in\mathcal{T}} A_t(\alpha_t) + \sum_{t\in\mathcal{T}_c} \beta_t \widetilde{d}^\star(\alpha_t), \tag{2.7.5}$$

$$\widetilde{D}^\star\left(\overline{\mathcal{S}}\right) = \sum_{t\in\mathcal{T}} \overline{A}_t(\alpha_{t,1},\cdots,\alpha_{t,J_t-1}) + \sum_{t\in\mathcal{T}_c} \beta_t \widetilde{d}^\star(\alpha_t), \tag{2.7.6}$$

where A_t and $\overline{A}_t$ are given as follows:

$$A_t(\alpha_t) = \beta_t \widetilde{d}^\star(\alpha_t) \tag{2.7.7}$$

$$\overline{A}_t(\alpha_{t,1},\cdots,\alpha_{t,J_t-1}) = \sum_{j=1}^{J_t-1}\beta_{t,j}\widetilde{d}^\star(\alpha_{t,j}) + \left(\beta_t - \sum_{j=1}^{J_t-1}\beta_{t,j}\right)\widetilde{d}^\star\left(\frac{\beta_t\alpha_t - \sum_{j=1}^{J_t-1}\beta_{t,j}\alpha_{t,j}}{\beta_t - \sum_{j=1}^{J_t-1}\beta_{t,j}}\right). \tag{2.7.8}$$

For each $t\in\mathcal{T}$, we compare the items A_t and $\overline{A}_t$. Firstly, we take the partial derivative of $\overline{A}_t$ w.r.t. $\alpha_{t,1},\cdots,\alpha_{t,J_t-1}$, respectively, and we can obtain

$$\frac{\partial\overline{A}_t}{\partial\alpha_{t,j_0}} = \beta_{t,j_0}c_1 log\left(\frac{\beta_t\alpha_t - \sum_{j=1}^{J_t-1}\beta_{t,j}\alpha_{t,j}}{\alpha_{t,j_0}\left(\beta_t - \sum_{j=1}^{J_t-1}\beta_{t,j}\right)}\right),$$

for $j_0 \in \mathcal{J}_t$, where $\mathcal{J}_t = \{1, \cdots, J_t - 1\}$. By setting $\partial \overline{A}_t / \partial \alpha_{t,j_0} = 0$, we conclude that if $\alpha_{t,j_0} < B$, $\overline{A}_t$ is an increasing function of α_{t,j_0}; if $\alpha_{t,j_0} > B$, $\overline{A}_t$ is a decreasing function of α_{t,j_0}. When $\alpha_{t,j_0} = B$, $\overline{A}_t$ achieves the largest value w.r.t. argument α_{t,j_0}. B is a function of variables $\{\alpha_{t,j}, j \in \mathcal{J}_t \backslash j_0\}$ and is given as follows:

$$B\left(\alpha_{t,j}, j \in \mathcal{J}_t \backslash j_0\right) = \frac{\beta_t \alpha_t - \sum_{j \in \mathcal{J}_t \backslash j_0} \beta_{t,j} \alpha_{t,j}}{\beta_t - \sum_{j \in \mathcal{J}_t \backslash j_0} \beta_{t,j}}.$$

Then the critical values which maximize the $\overline{A}_t$ should satisfy the following equations:

$$\begin{cases} \alpha_{t,1}^{\star} = B\left(\alpha_{t,j}^{\star}, j \in \mathcal{J}_t \backslash 1\right) & (1) \\ \vdots & \vdots \\ \alpha_{t,J_t-1}^{\star} = B\left(\alpha_{t,j}^{\star}, j \in \mathcal{J}_t \backslash J_t - 1\right) & (J_t - 1) \end{cases}$$

By substitution method, we can solve the equation set, and get that

$$\alpha_{t,1}^{\star} = \alpha_{t,2}^{\star} = \cdots = \alpha_{t,J_t-1}^{\star} = \alpha_t.$$

Then we have $\alpha_{t,J_t}^{\star} = \alpha_t$. Therefore, when $\alpha_{t,j} = \alpha_t$, for $j \in \{1, \cdots, J_t\}$, $\overline{A}_t$ is maximized, and the maximum value is

$$\overline{A}_t(\alpha_{t,1}^{\star}, \cdots, \alpha_{t,J_t-1}^{\star}) = \beta_t \widetilde{d}^{\star}(\alpha_t) = A_t(\alpha_t).$$

Therefore, $\overline{A}_t \leq A_t$, for all $t \in \mathcal{T}$, and the equality is obtained if and only if $\alpha_{t,j} = \alpha_t$, for all $j \in \{1, \cdots, J_t\}$ and $t \in \mathcal{T}$.

Chapter 3

Dynamic Turbo-OAMP for Downlink FDD-Massive MIMO Channel Tracking

3.1 Introduction

In an FDD system, if we use conventional CE methods, such as least squares (LS) and minimum mean square error (MMSE) [60] to estimate the downlink channel at the user side, the number of pilot symbols should be at least the same as the number of antennas M at the BS, which would be prohibitively large for massive MIMO system. Various CS-based downlink CE schemes have been developed to exploit the inherit sparsity of massive MIMO channels so as to reduce the pilot training overhead, such as OMP and CoSaMP [61]. Several channel estimation algorithms exploit the spatial and/or temporal correlation of the channel to further reduce the pilot overhead and improve the real-time channel tracking performance, as summarized below.

- **Algorithms exploiting the spatial correlation[1] of the channel support:** In [7], it is shown that the angular domain channel support has a burst structure due to the physical scattering structure in the environment, and this burst sparsity structure has been exploited to design a burst-LASSO algorithm in [7]. Recently, a structured Turbo-CS algorithm with Markov channel prior was proposed in [25] to exploit the clustered sparse structure in the spatial domain for massive MIMO channel estimation.

[1] In this chapter, spatial correlation refers to the structured spatial sparsity of a massive MIMO channel in which the non-zero elements of the angular domain channel tend to concentrate on a few clusters.

- **Algorithms exploiting the temporal correlation of the channel:** Since the propagation environment is dynamically changing with the temporal correlation, the previously estimated channel can provide some prior information about the current channel. In [62], the channel is modeled by a Markov process and the classical Kalman filter is used to track the channel. However, the method in [62] cannot exploit the sparsity in massive MIMO channels. In [34–36] and [37], both the prior information obtained from the temporal correlation and the channel sparsity have been exploited to design the prior-aided sparse channel tracking schemes.

- **Algorithms exploiting the frequency (delay)-temporal correlation of the channel support for OFDM systems:** In [9] and [10], adaptive channel estimation schemes are designed for broad band systems where orthogonal frequency-division multiplexing (OFDM) is used. They assume that the sparsity structure is shared by subchannels of different subcarriers (or delay domain channels between the user and different transmit antennas at the BS), and such sparsity is almost unchanged in multiple time blocks.

However, the existing algorithms have the following drawbacks. First, the algorithms exploiting the spatial and/or temporal correlation of the channel support rely on some restrictive assumptions. For example, the burst LASSO algorithm in [7] only works when all bursts of the channel vector have similar sizes, which is not satisfied by practical massive MIMO channels with random burst sizes. [9] and [10] assume that the channel supports across multiple time slots are approximately the same. However, in practice, channel support is time-varying and may undergo a sudden change within two adjacent time slots. Second, many existing algorithms [9, 35] are batch algorithms which require the collection of measurements across multiple time slots to recover a batch of unknown signals simultaneously. Such batch algorithms are offline, slow, and require linearly increasing memory with the sequence length. In practice, it is desirable to design recursive algorithms, which only use the previous signal estimate and current measurements to estimate the current signal, and thus have lower computation complexity and storage requirements. Third, in this chapter, we model the fading channel as a dynamic process, and at the same time, maintain spatial sparsity structures as channel evolves over time. Spatial sparsity structure means that the non-zero elements of the angular domain channel tend to concentrate on a few clusters; a dynamic process means the probabilistic dependency of the channels over time. Because these characteristics will cause the sparsity pattern of the angular domain channel spatially and temporally correlated,

we call it as *two-dimensional (2D) dynamic sparsity* in the rest of the chapter. The existing channel tracking works also exploited the dynamic sparsity. However, [9, 10, 36, 63] failed to model the dynamic evolution of the channel, and just assumed the common sparsity across time or the prior information quality bound; [34, 35, 37, 62, 64–66] considered the dynamic evolution of the channel, but they failed to exploit the spatial sparsity structure while the channel evolves. Therefore, there is no existing work which exploited the structured spatial sparsity and the probabilistic temporal dependency of the channels jointly to track the dynamic channels.

In this chapter, we consider downlink FDD-massive MIMO system and propose a systematic framework called *Dynamic Turbo Orthogonal Approximate Message Passing* (D-TOAMP) to recursively track time-varying channels with low pilot overhead and improved performance. The main contributions are summarized below.

- **Realistic Two-dimensional Markov Channel Model:** We propose a new statistical model called the 2D-MM to capture the 2D dynamic sparsity of massive MIMO channels. The 2D-MM has the flexibility to model different propagation environments that occur in practice. Moreover, we verify the validity of the 2D-MM using realistic channel models. To the best of our knowledge, this is the first work which uses a 2D-MM to model the 2D dynamic sparsity of sparse massive MIMO channels.

- **Design of Dynamic Turbo-OAMP Algorithm:** By combining the turbo approach and the OAMP [21–23], we propose an efficient SPMP algorithm to recursively track the dynamic channels with a 2D-MM prior. The OAMP in [21–23] only works for i.i.d. priors. We extend the OAMP to D-TOAMP, which works for the 2D-MM prior.

The rest of this chapter is organized as follows. In Section 3.2, we present the system model. In Section 3.3, we introduce the 2D-MM channel prior. In Section 3.4, we present the sparse channel tracking formulation. In Section 3.5, we present the proposed D-TOAMP algorithm and give a complexity analysis. Simulation results and summaries are given in Section 3.6 and 3.8, respectively.

3.2 System Model

3.2.1 Downlink Training

Consider a massive MIMO system with one BS serving a single antenna user[2]. The BS is equipped with $M \gg 1$ antennas. To estimate the downlink channel $\mathbf{h}_t^H \in \mathbb{C}^{1\times M}$ at time slot t, the BS transmits P pilot sequences $\mathbf{u}_{p,t} \in \mathbb{C}^{M\times 1}, p = 1, \cdots, P$. The received signal $\mathbf{y}_t^H \in \mathbb{C}^{1\times P}$ at time slot t is

$$\mathbf{y}_t^H = \mathbf{h}_t^H \mathbf{U}_t + \mathbf{n}_t^H, \tag{3.2.1}$$

where $\mathbf{U}_t = [\mathbf{u}_{1,t}, \cdots, \mathbf{u}_{P,t}] \in \mathbb{C}^{M\times P}$ is the pilot matrix, and $\mathbf{n}_t \sim \mathcal{CN}(0, \sigma_e^2 \boldsymbol{I})$ is the additive complex Gaussian noise.

In this chapter, we consider massive MIMO at the BS side. For large number of antennas, the spatial resolution of the angular basis increases. Hence, under limited scattering environment, the channel will be sparse under the angular basis.

3.2.2 Massive MIMO Channel Model

We consider flat fading massive MIMO channel with limited scattering around the BS. For clarity, we focus on the case when the BS is equipped with a *half-wavelength space* ULA. In this case, the downlink channel vector $\mathbf{h}_t \in \mathbb{C}^M$ can be modeled as [2]

$$\mathbf{h}_t = \sum_{c=1}^{N_c} \sum_{b=1}^{N_b} a_{t,c,b} \mathbf{a}\left(\vartheta_{t,c,b}\right), \tag{3.2.2}$$

where N_c stands for the number of scattering clusters, N_b stands for the number of sub-paths per cluster, $a_{t,c,b}$ and $\vartheta_{t,c,b}$ stand for the complex channel gain and the azimuth AoD corresponding to the b-th sub-path in the c-th scattering cluster at time slot t. The steering vector $\mathbf{a}\left(\vartheta\right) \in \mathbb{C}^M$ for ULA is

$$\mathbf{a}\left(\vartheta\right) = \left[1, e^{-j\pi\sin(\vartheta)}, \cdots, e^{-j\pi(M-1)\sin(\vartheta)}\right]^T.$$

[2]Note that we focus on downlink channel estimation, where the base station sends some common pilots for all users to estimate their own channel. In a multi-user downlink massive MIMO system, each user can independently perform channel tracking using the proposed D-TOAMP based on the received signal from the common pilots. Therefore, without loss of generality, we can focus on the algorithm design for a reference user, and the proposed D-TOAMP can be directly applied to a multi-user massive MIMO system.

3.2.3 Off-Grid Basis for Massive MIMO Channels

In this subsection, we describe the angular domain channel representation at time slot t. For ease of notation, we drop the time index t. The true AoDs could be denoted as $\{\vartheta_1, \cdots, \vartheta_L\}$ where $L = N_c N_b$. Let $\left\{\hat{\vartheta}_1, \cdots, \hat{\vartheta}_M\right\}$ be a uniform sampling grid, which uniformly covers the angular domain $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. In practice, the true AoDs usually do not lie exactly on the grid points. To handle the direction mismatch, we adopt an off-grid model. Specifically, if $\vartheta_l \notin \left\{\hat{\vartheta}_1, \cdots, \hat{\vartheta}_M\right\}$, and $\hat{\vartheta}_{m_l}$, $m_l \in \{1, \cdots, M\}$ is the nearest grid point to ϑ_l, we write ϑ_l as

$$\vartheta_l = \hat{\vartheta}_{m_l} + \beta_{m_l}, \tag{3.2.3}$$

where β_{m_l} is the off-grid gap. Then we have $\mathbf{a}(\vartheta_l) = \mathbf{a}\left(\hat{\vartheta}_{m_l} + \beta_{m_l}\right)$. The downlink channel $\mathbf{h}$ in (3.2.2) has a sparse representation with off-grid basis as given by

$$\mathbf{h} = \mathbf{A}(\boldsymbol{\beta})\,\mathbf{x}, \tag{3.2.4}$$

where $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_M]^T$, $\mathbf{A}(\boldsymbol{\beta}) = \left[\mathbf{a}\left(\hat{\vartheta}_1 + \beta_1\right), \cdots, \mathbf{a}\left(\hat{\vartheta}_M + \beta_M\right)\right]$, $\mathbf{x} \in \mathbb{C}^M$ is the sparse angular domain channel, and

$$\beta_{m_l} = \begin{cases} \vartheta_l - \hat{\vartheta}_{m_l}, & l = 1, \cdots L \\ 0, & \text{otherwise} \end{cases}. \tag{3.2.5}$$

Note that with the off-grid basis, the model could significantly alleviate the direction mismatch because there always exists some β_{m_l} making (3.2.3) hold exactly.

We can also obtain similar sparse representation with off-grid basis for more general 2D antenna arrays. In this case, the steering vector $\mathbf{a}(\vartheta, \varphi)$ can be expressed as a function of the azimuth angle ϑ and elevation angle φ. Please refer to [27] for the detailed expression of $\mathbf{a}(\vartheta, \varphi)$. In this case, the downlink channel vector $\mathbf{h}_t \in \mathbb{C}^M$ can be modeled as [27]

$$\mathbf{h}_t = \sum_{l=1}^{L} a_l \mathbf{a}(\vartheta_l, \varphi_l), \tag{3.2.6}$$

where L stands for the total number of sub-paths, a_l, ϑ_l, φ_l stand for the complex channel gain, the azimuth AoD, and elevation AoD corresponding to the l-th sub-path. Then the downlink channel $\mathbf{h}$ in (3.2.6) also has a sparse representation with off-grid basis as given by

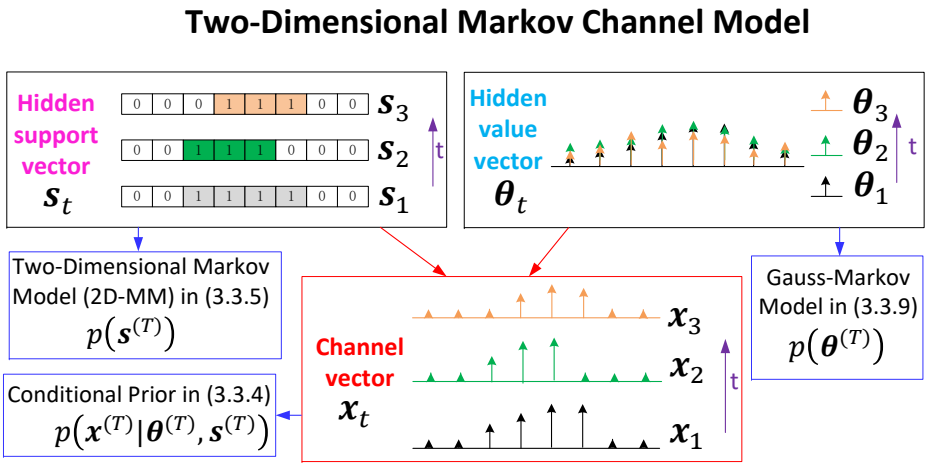

Figure 3.1: Two-dimensional Markov channel model

$\mathbf{h} = \mathbf{A}\left(\boldsymbol{\beta}, \hat{\boldsymbol{\varphi}}\right)\mathbf{x}$, where $\mathbf{A}\left(\boldsymbol{\beta}, \hat{\boldsymbol{\varphi}}\right) = \left[\mathbf{a}\left(\hat{\vartheta}_1 + \beta_1, \hat{\varphi}_1\right), \cdots, \mathbf{a}\left(\hat{\vartheta}_M + \beta_M, \hat{\varphi}_M\right)\right]$,

$$\hat{\varphi}_{m_l} = \begin{cases} \varphi_l, & l = 1, \cdots L \\ 0, & \text{otherwise} \end{cases},$$

the definition of m_l can be found in (3.2.3). The parameters $(\boldsymbol{\beta}, \hat{\boldsymbol{\varphi}})$ could be learned through the EM framework in the proposed D-TOAMP algorithm.

The proposed algorithm in this chapter can be applied to general array geometry at the BS with an invertible array response matrix $\mathbf{A}\left(\mathbf{0}, \mathbf{0}\right)$. For a ULA with half-wavelength inter antenna spacing, the array response matrix $\mathbf{A}$ with $\boldsymbol{\beta} = \mathbf{0}$ is a DFT matrix [9, 36], which is invertible.

3.3 Two-Dimensional Markov Channel Model

Using the angular domain channel representation, the downlink channel at time slot t can be expressed as

$$\mathbf{h}_t = \mathbf{A}\left(\boldsymbol{\beta}_t\right)\mathbf{x}_t, \tag{3.3.1}$$

where $\mathbf{x}_t$ is the angular domain channel vector at time slot t.

The channel model in (3.3.1) lacks a probability model for $\mathbf{x}_t$. Such a probability model provides the foundation for exploiting the 2D dynamic sparsity of massive MIMO channels. In existing work, there are some attempts to exploit the sparsity for massive MIMO CE under a very simple assumption for i.i.d. sparsity [61]. However, in practice, due to the clustered

scattering, the support of the massive MIMO channels will not be i.i.d. distributed. In [7], a burst sparsity is introduced to account for the clustered scattering. However, this model is deterministic and cannot capture a more complicated clustered scattering structure with random cluster sizes and locations. Furthermore, due to slowly changing propagation environment, the dynamic scattering structures are temporally correlated. A previous estimated channel can provide prior information to enhance the current CE efficiency [9, 10, 34–37]; however, the assumptions of the deterministic support structure shared by consecutive time slots in [9, 10] and the prior information quality bound in [36] are too restrictive.

Challenge 1: Propose a probabilistic channel model to capture a more realistic 2D dynamic sparsity of the massive MIMO channels.

In this section, we introduce a 2D Markov model to capture a more realistic 2D dynamic sparsity of the massive MIMO channels. Fig. 3.1 illustrates the high level structure of the 2D-MM for the massive MIMO channels $\mathbf{x}_1, \cdots, \mathbf{x}_T$.

Let Ω_t denote the index set of non-zero elements of $\mathbf{x}_t$, which is called the *channel support* at time slot t. In order to characterize the 2D dynamic sparsity of a dynamic channel $\mathbf{x}^{(T)} = \{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$, we adopt a probabilistic signal model with two hidden random processes $\mathbf{s}^{(T)} = \{\mathbf{s}_1, \cdots, \mathbf{s}_T\}$ and $\boldsymbol{\theta}^{(T)} = \{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_T\}$. The binary vector $\mathbf{s}_t = [s_{t,1}, \cdots, s_{t,M}]^T \in \{0,1\}^M$, with $s_{t,m} = 1$ if $m \in \Omega_t$, and $s_{t,m} = 0$ otherwise, represents the *hidden support vector* at time slot t, which describes the 2D dynamic sparsity of the channel sparsity pattern. The complex-valued vector $\boldsymbol{\theta}_t = [\theta_{t,1}, \cdots, \theta_{t,M}]^T \in \mathbb{C}^M$ with $\theta_{t,m} = x_{t,m}$ if $s_{t,m} = 1$ represents the *hidden value vector* at time slot t, which characterizes the temporal correlation of the channel coefficients. The dynamic channel can be modeled as

$$x_{t,m} = s_{t,m} \cdot \theta_{t,m}, \ t = 1, \cdots, T, 1 \leq m \leq M, \tag{3.3.2}$$

where $s_{t,m}$ denotes whether there is an active path from the m-th AoD direction in the t-th time slot at the BS, and $\theta_{t,m}$ denotes the corresponding complex path gain. Then the 2D-MM channel prior distribution (joint distribution of $\mathbf{x}^{(T)}, \mathbf{s}^{(T)}, \boldsymbol{\theta}^{(T)}$) is given by

$$p\left(\mathbf{x}^{(T)}, \mathbf{s}^{(T)}, \boldsymbol{\theta}^{(T)}\right) = \underbrace{p\left(\mathbf{x}^{(T)} \middle| \mathbf{s}^{(T)}, \boldsymbol{\theta}^{(T)}\right)}_{\text{channel vector}} \underbrace{p\left(\mathbf{s}^{(T)}\right)}_{\text{hidden support}} \underbrace{p\left(\boldsymbol{\theta}^{(T)}\right)}_{\text{hidden value}}, \tag{3.3.3}$$

where the probability model for channel vectors $\mathbf{x}^{(T)}$ is conditioned on the hidden support

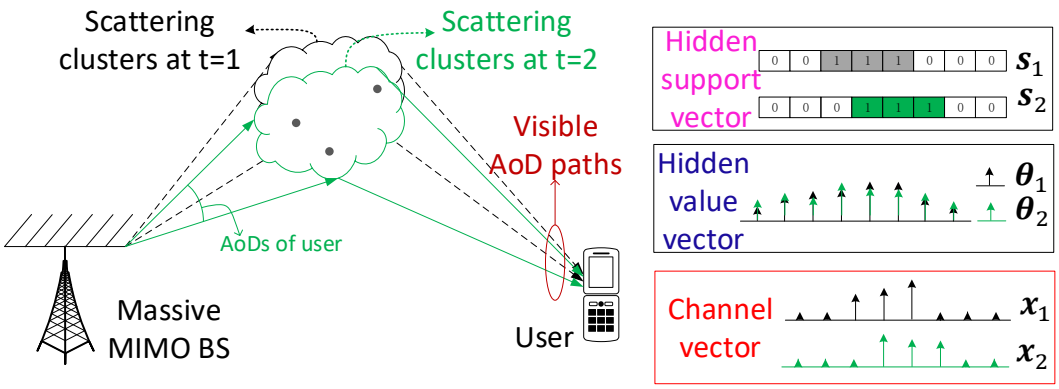

Figure 3.2: Illustration of the 2D dynamic sparsity of the massive MIMO channel for $T = 2$. Due to limited and clustered scattering at the BS, the hidden support vector $\mathbf{s}_t$ will be sparse with clustered non-zero elements. Due to the slowly changing scattering environment, the hidden support vector $\mathbf{s}_t$ and hidden value vector $\boldsymbol{\theta}_t$ will be temporally correlated.

vectors and hidden value vectors, the hidden support vectors $\mathbf{s}^{(T)}$ form a 2D-MM, and the hidden value vectors $\boldsymbol{\theta}^{(T)}$ form a Gauss-Markov process, as detailed below.

3.3.1 Probability Model for Channel Vector

The conditional prior $p\left(\mathbf{x}^{(T)} \middle| \mathbf{s}^{(T)}, \boldsymbol{\theta}^{(T)}\right)$ is given by

$$p\left(\mathbf{x}^{(T)} \middle| \mathbf{s}^{(T)}, \boldsymbol{\theta}^{(T)}\right) = \prod_{t=1}^{T} \prod_{m=1}^{M} \underbrace{p\left(x_{t,m} | s_{t,m}, \theta_{t,m}\right)}_{f_{t,m}(x_{t,m}, s_{t,m}, \theta_{t,m})} = \prod_{t=1}^{T} \prod_{m=1}^{M} \delta\left(x_{t,m} - \theta_{t,m} s_{t,m}\right), \tag{3.3.4}$$

where $\delta(\cdot)$ is the Dirac delta function. Conditioned on $\mathbf{s}^{(T)}, \boldsymbol{\theta}^{(T)}$, $x_{t,m}$'s are independent. By definition, $s_{t,m} = 0$ sets $x_{t,m} = 0$, while $s_{t,m} = 1$ sets $x_{t,m} = \theta_{t,m}$.

The 2D-MM channel is motivated by the physical propagation mechanism of radio waves. Physically, each Tx scatterer at the BS side corresponds to an AoD, as shown in Fig. 3.2. If the m-th AoD path is visible to the user (i.e., the signal of the m-th AoD path reflected/scattered from the Tx scatterer can reach the user), the m-th element of angular domain channel vector $\mathbf{x}_t$ will be non-zero. In practice, there are limited Tx scatterers at the BS, i.e., the BS is elevated high or the carrier frequency is high, so only part of the AoD paths can reach the user. The hidden support vector $\mathbf{s}_t$ represents the AoD paths that can be seen by the user. Moreover, since we consider a channel tracking problem where an invisible AoD path at the current time slot may become visible at the next time slot (and vice-versa), it is necessary to use a hidden value vector $\boldsymbol{\theta}_t$ to model the path gains of all the M potential AoD paths, as shown by [34] and [35]. Therefore, it is natural to model the actual channel vector $\mathbf{x}_t$ as the

product of $\mathbf{s}_t$ and $\boldsymbol{\theta}_t$.

3.3.2 Two-Dimensional Markov Model of Hidden Support Vector

Due to the clustered structure of the scatterers at the BS side, the non-zero elements in $\mathbf{s}_t$ will concentrate on a few clusters [7], where each cluster corresponds to a transmit scattering cluster. This spatially clustered structure implies that $s_{t,m}$ depends on $s_{t,m-1}$, e.g., if $s_{t,m-1} = 1$, then there is a higher probability that $s_{t,m}$ is also 1. Moreover, it has been verified in [34] and [35] that the channel supports often change slowly over time, which implies that $s_{t,m}$ also depends on $s_{t-1,m}$, e.g., if $s_{t-1,m} = 1$, then there is a higher probability that $s_{t,m}$ is also 1. Such 2D dynamic sparsity of hidden support vectors can be naturally modeled as the following 2D-MM [67]:

$$p\left(\mathbf{s}^{(T)}\right) = \underbrace{p\left(s_{1,1}\right)}_{h_{1,1}(s_{1,1})} \prod_{m=2}^{M} \underbrace{p\left(s_{1,m}|s_{1,m-1}\right)}_{h_{1,m}(s_{1,m},s_{1,m-1})} \times \prod_{t=2}^{T} \left(\underbrace{p\left(s_{t,1}|s_{t-1,1}\right)}_{h_{t,1}(s_{t,1},s_{t-1,1})} \prod_{m=2}^{M} \underbrace{p\left(s_{t,m}|s_{t,m-1}, s_{t-1,m}\right)}_{h_{t,m}(s_{t,m},s_{t,m-1},s_{t-1,m})} \right), \tag{3.3.5}$$

whose 2D transition probabilities are defined as $\rho_{01}^{S} = p\left(s_{1,m} = 1|s_{1,m-1} = 0\right)$, $\rho_{10}^{S} = p\left(s_{1,m} = 0|s_{1,m-1} = 1\right)$, $\rho_{01}^{T} = p\left(s_{t,1} = 1|s_{t-1,1} = 0\right)$, $\rho_{10}^{T} = p\left(s_{t,1} = 0|s_{t-1,1} = 1\right)$ and $\rho_{bca} = p\left(s_{t,m} = a|s_{t,m-1} = b, s_{t-1,m} = c\right)$, where $a, b, c \in \{0, 1\}$, $t > 1$ and $m > 1$. The factor graph of the 2D-MM of the hidden support vector is illustrated in Fig. 3.3-a, where the factor nodes $h_{t,m}$ are the conditional priors (priors) in (3.3.5). Note that we can always find a set of transition probabilities $\{\rho_{ba}^{S}, \rho_{ca}^{T}, \rho_{bca}\}$ to make the 2D-MM operate in a steady-state [68], such that $p(s_{t,m} = 1) = \lambda, \forall t, m$, where $\lambda > 0$ is the sparsity ratio.

Depending on how $\{\rho_{bca}\}$, $a, b, c \in \{0, 1\}$ are choosing, the prior distribution in (3.3.5) can favor nearly static support or substantially changing support in the temporal domain. For example, higher ρ_{b11} and smaller ρ_{b01}, $b \in \{0, 1\}$ lead to temporally highly correlated priors of $\mathbf{s}^{(T)}$. Meanwhile, the prior distribution in (3.3.5) can characterize the clustered structure of $\mathbf{s}_t$ in the spatial domain. For example, a higher $\rho_{1c1}, c \in \{0, 1\}$ leads to a larger average cluster size, and a smaller $\rho_{0c1}, c \in \{0, 1\}$ leads to a larger average gap between clusters in $\mathbf{s}_t$. As such, the 2D-MM in (3.3.5) is a general flexible model to characterize the 2D dynamic sparsity of $\mathbf{s}^{(T)}$.

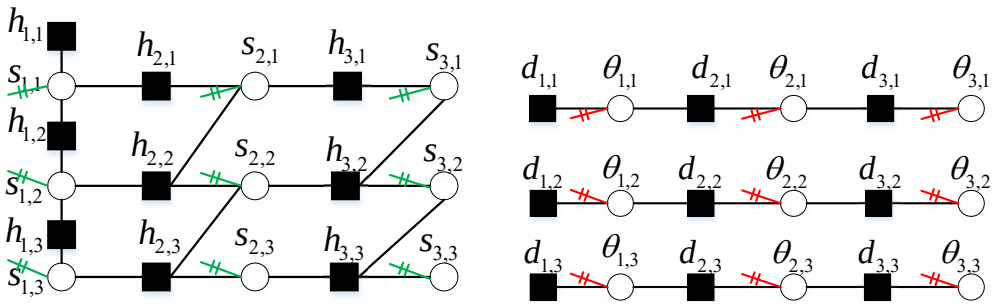

Figure 3.3: Factor graphs of hidden support and value vectors when $M = 3$ and $T = 3$. (a) Left: Factor graph of the 2D-MM of the hidden support vectors; (b) Right: Factor graph of Gauss-Markov model of the hidden value vectors.

3.3.3 Gauss-Markov Model of Hidden Value Vector

It has been shown in [34] and [35] that the path gains evolve smoothly over time. We can use the spatially independent steady-state Gauss-Markov processes to model the temporal evolution of the hidden value vector as follows [66]:

$$\theta_{t,m} = (1-\alpha)\left(\theta_{t-1,m} - \zeta\right) + \alpha w_{t,m} + \zeta, \tag{3.3.6}$$

where $\alpha \in [0,1]$ controls the temporal correlation, $\zeta \in \mathbb{C}$ is the mean of the process, and $w_{t,m} \sim \mathcal{CN}(0,\kappa)$ is the i.i.d Gaussian perturbation with mean 0 and variance κ. Specifically, if $\alpha = 0$, then $\theta_{t,m} = \theta_{t-1,m}$, which means the hidden value vector $\boldsymbol{\theta}_t$ is unchanged over time. If $\alpha = 1$, then $\theta_{t,m} = w_{t,m} + \zeta \sim \mathcal{CN}(\zeta,\kappa)$, which means the hidden value vector $\boldsymbol{\theta}_t$ is i.i.d Gaussian distributed over time. If $0 < \alpha < 1$, based on (3.3.6), the conditional probability of $\theta_{t,m}$ could be given by

$$p\left(\theta_{t,m}|\theta_{t-1,m}\right) \sim \mathcal{CN}\left(\theta_{t,m}; (1-\alpha)\,\theta_{t-1,m} + \alpha\zeta, \alpha^2\kappa\right). \tag{3.3.7}$$

In the steady-state

$$\theta_{t,m} \sim \mathcal{CN}\left(\zeta, \sigma^2\right), \forall t, m, \tag{3.3.8}$$

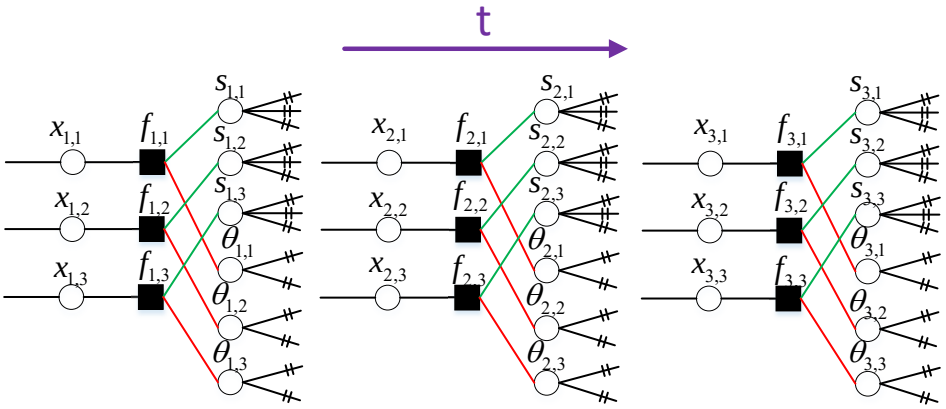

Figure 3.4: Factor graph of the 2D-MM channel when $M = 3$ and $T = 3$. The detailed factor graphs for hidden support vector $\mathbf{s}_t$ and hidden value vector $\boldsymbol{\theta}_t$ are illustrated in Fig. 3.3.

where the steady-state variance is $\sigma^2 = \frac{\alpha\kappa}{2-\alpha}$. The joint distribution of $\boldsymbol{\theta}^{(T)}$ can be calculated as follows:

$$p\left(\boldsymbol{\theta}^{(T)}\right) = \prod_{m=1}^{M} \underbrace{p\left(\theta_{1,m}\right)}_{d_{1,m}(\theta_{1,m})} \prod_{t=2}^{T} \underbrace{p\left(\theta_{t,m}|\theta_{t-1,m}\right)}_{d_{t,m}(\theta_{t,m},\theta_{t-1,m})}. \tag{3.3.9}$$

The factor graph of the Markov model of the hidden value vector is illustrated in Fig. 3.3-b, where the factor nodes $d_{t,m}$ are the conditional priors in (3.3.7) for $t > 1$ and priors in (3.3.8) for $t = 1$.

Finally, the overall factor graph of the 2D-MM channel is illustrated in Fig. 3.4, where the factor node $f_{t,m}$ is the conditional prior in (3.3.4).

3.3.4 Verification of Two-Dimensional Markov Channel Model

Compared to other static channel models [7, 9, 10], the proposed 2D-MM provides more flexibility to model more realistic channels. Specifically, in our model, the average cluster size and cluster number in the spatial domain, the average support changing frequency and the dependency of the channel gains across time are determined by a set of parameters $\boldsymbol{\rho} \triangleq \{\rho_{ba}^S, \rho_{ca}^T, \rho_{bca}, \alpha, \zeta, \kappa, \sigma^2\}$, where $a, b, c \in \{0, 1\}$. For given parameters, the channel realizations could have different spatial-temporal properties, i.e., different cluster numbers and cluster sizes, and different channel evolution across time. As such, the proposed 2D-MM channel can be used to model various channel realizations in practice, and thus works well for realistic channels. Moreover, the statistic parameters in our model $\boldsymbol{\rho}$ could be automatically learned by the proposed D-TOAMP algorithm based on the EM framework during the recovery process.

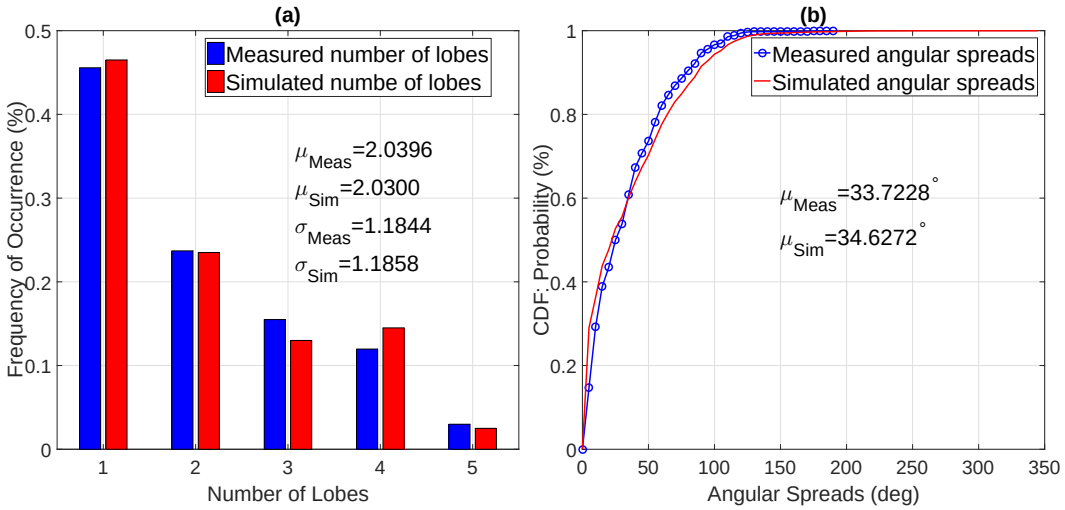

Figure 3.5: Comparison of the measured channel property extracted from the 28-GHz mm-SSCM [1] and the simulated channel property extracted from the 2D-MM channel. We set $M = 256$ and $T = 50$. The results are calculated through 200 channel series realizations. (a) Number of AoD SLs; (b) AoD global AS.

In this section, we will also provide some verifications of the proposed 2D-MM channel. Fig. 3.5a shows a typical empirical histogram plot of the number of AoD spatial lobes (SLs) extracted from the 28-GHz millimeter-wave statistical spatial channel model (mm-SSCM) proposed in [1], next to the simulated histograms extracted from the proposed 2D-MM channel. It can be seen that the proposed 2D-MM channel prior yields good agreement with the practical mmWave channels.

AoD global AS describes the degree of angular dispersion at the BS over the entire 2π azimuth plane. The AoD global AS in Fig. 3.5b are computed based on a -10 dB lobe threshold from the measured data in the 28-GHz mm-SSCM [1], and are compared to the simulated values using the 2D-MM channel prior. It can be seen that the statistics of the simulated and measured global AS match well.

3.4 Massive MIMO Channel Tracking with 2D Dynamic Sparsity

Using the angular domain channel representation, (3.2.1) can be rewritten as a standard CS model as in [7, 36]

$$\mathbf{y}_t = \mathbf{F}_t\left(\boldsymbol{\beta}_t\right)\mathbf{x}_t + \mathbf{n}_t, \forall t, \tag{3.4.1}$$

where the measurement matrix $\mathbf{F}_t(\boldsymbol{\beta}_t)$[3] is given by

$$\mathbf{F}_t(\boldsymbol{\beta}_t) = \mathbf{U}_t^H \mathbf{A}(\boldsymbol{\beta}_t) \in \mathbb{C}^{P\times M}. \tag{3.4.2}$$

It is well known that the choice of measurement matrix will affect the CE performance significantly. It is shown in [22, 23, 25] that a partial orthogonal measurement matrix achieves better performance than an i.i.d. Gaussian matrix under the OAMP algorithm. In order to reduce the signaling overhead between the BS and user, the pilot matrix is designed by assuming a fixed off-grid parameter $\boldsymbol{\beta}_t = \mathbf{0}$. Then the pilot matrix $\mathbf{U}_t$ is designed as $\mathbf{U}_t^H = \mathbf{S}_t\mathbf{D}\mathbf{R}_t\mathbf{A}(\mathbf{0})^{-1}$, such that $\mathbf{F}_t(\mathbf{0}) = \mathbf{S}_t\mathbf{D}\mathbf{R}_t$, which is referred to as a partial DFT-random permutation (RP) measurement matrix. $\mathbf{S}_t \in \{0,1\}^{P\times M}$ is a selection matrix consisting of randomly selected and reordered P rows of an $M \times M$ identity matrix, $\mathbf{D} \in \mathbb{C}^{M\times M}$ is the DFT matrix, and $\mathbf{R}_t \in \{0,1\}^{M\times M}$ is a RP matrix generated by a randomly reordered $M \times M$ identity matrix. For large antenna arrays, M is large and the measurement matrix $\mathbf{F}_t(\boldsymbol{\beta}_t) = \mathbf{S}_t\mathbf{D}\mathbf{R}_t\mathbf{A}(\mathbf{0})^{-1}\mathbf{A}(\boldsymbol{\beta}_t) \approx \mathbf{S}_t\mathbf{D}\mathbf{R}_t$ is approximately partial orthogonal. Simulations in Section 3.6 verify that such choice of pilot matrix can indeed achieve a good performance[4].

Remark 3.1. In the original OAMP algorithm, the measurement matrix is chosen as partial DFT matrix and the sparse signal follows i.i.d distribution. However, as shown in Fig. 3.6, we find that partial DFT matrix does not work well for signals with complicated correlations, i.e., 2D-MM priors. On the other hand, the partial DFT-RP matrix decorrelates the sparse signal by introducing a RP. Therefore, partial DFT-RP achieves better performance compared to the partial DFT measurement matrix for the D-TOAMP algorithm with non-i.i.d structure of the sparse signals.

Based on the observation model in (3.4.1) and the 2D-MM for the massive MIMO channels $\mathbf{x}^{(t)}$ in (3.3.3), our primary goal is to recursively track the time-varying channel vector $\mathbf{x}_t$ and optimize the off-grid parameter $\boldsymbol{\beta}_t$ at the t-th time slot, given the observations up to t time slots $\mathbf{y}^{(t)} = \{\mathbf{y}_\tau\}_{\tau=1}^t$ and the (approximate) optimal off-grid parameters up to $(t-1)$ time slot $\boldsymbol{\beta}^{*(t-1)} = \{\boldsymbol{\beta}_\tau^*\}_{\tau=1}^{t-1}$ in (3.4.1). Because $\boldsymbol{\beta}^{*(t-1)}$ is fixed for the channel tracking problem at the t-th time slot, we omit $\boldsymbol{\beta}^{*(t-1)}$ in the probability expressions for simplicity. In particular,

[3]We use $\mathbf{F}_t$ as the notation for measurement matrix in this chapter.

[4]Note that the pilot sequence in (3.4.2) can be static or time varying. In both cases, they achieve similar performances.

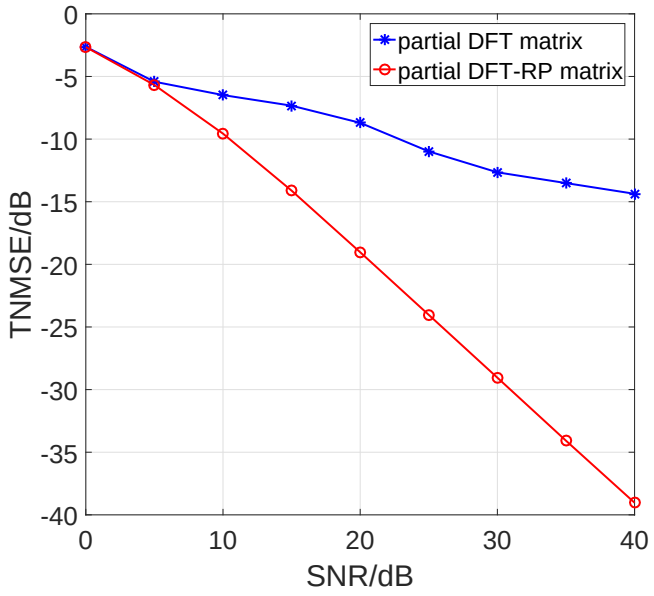

Figure 3.6: TNMSE performance (defined in (3.6.1)) of the proposed D-TOAMP algorithm versus SNR for different partial orthogonal measurement matrices design. Set $T = 50, P = 50, M = 128, \lambda = 0.25, \rho_{01}^S = 0.025, \rho_{10}^S = 0.075, \rho_{01}^T = 0.05, \rho_{10}^T = 0.05, \rho_{111} = 0.9958, \rho_{001} = 0.0013, \rho_{011} = 0.3276, \rho_{101} = 0.3936, \kappa = 1, \sigma^2 = \frac{1}{3}, \zeta = 0, \alpha = 0.5$. For partial DFT matrix, $\mathbf{F}_t = \mathbf{S}_t\mathbf{D}$; for partial DFT-RP matrix, $\mathbf{F}_t = \mathbf{S}_t\mathbf{D}\mathbf{R}_t$.

for given $\boldsymbol{\beta}_t$, we are interested in computing the marginal posterior $p\left(x_{t,m}|\mathbf{y}^{(t)}, \boldsymbol{\beta}_t\right)$, where

$$\begin{aligned}
&p\left(x_{t,m}|\mathbf{y}^{(t)}, \boldsymbol{\beta}_t\right) \\
\propto &\sum_{\mathbf{s}^{(t)}} \int_{\mathbf{x}_{-(t,m)}^{(t)}, \boldsymbol{\theta}^{(t)}} p\left(\mathbf{x}^{(t)}, \mathbf{s}^{(t)}, \boldsymbol{\theta}^{(t)}, \mathbf{y}^{(t)}|\boldsymbol{\beta}_t\right) \\
= &\sum_{\mathbf{s}^{(t)}} \int_{\mathbf{x}_{-(t,m)}^{(t)}, \boldsymbol{\theta}^{(t)}} p\left(\mathbf{s}^{(t)}\right) p\left(\boldsymbol{\theta}^{(t)}\right) \prod_{\tau,m} p\left(x_{\tau,m}|s_{\tau,m}, \theta_{\tau,m}\right) \prod_{i=1}^{P} \prod_{\tau=1}^{t-1} p\left(y_{\tau,i}|\mathbf{x}_\tau\right) p\left(y_{t,i}|\mathbf{x}_t, \boldsymbol{\beta}_t\right).
\end{aligned} \tag{3.4.3}$$

$\mathbf{x}_{-(t,m)}^{(t)}$ denotes the vector collections $\{\mathbf{x}_\tau\}_{\tau=1}^t$ excluding the element $x_{t,m}$, $p\left(y_{t,i}|\mathbf{x}_t, \boldsymbol{\beta}_t\right) = \mathcal{CN}\left(y_{t,i}; \mathbf{f}_{t,i}\mathbf{x}_t, \sigma_e^2\right)$, and $y_{t,i}$ is the i-th element of $\mathbf{y}_t$, $\mathbf{f}_{t,i}$ is the i-th row of $\mathbf{F}_t\left(\boldsymbol{\beta}_t\right)$. We use $\propto$ to denote equality after scaling. On the other hand, the optimal off-grid parameter $\boldsymbol{\beta}_t$ is obtained by maximum likelihood (ML) as follows:

$$\begin{aligned}
\boldsymbol{\beta}_t^\star &= \arg\max_{\boldsymbol{\beta}_t} \ln p\left(\mathbf{y}^{(t)}, \boldsymbol{\beta}_t\right) \\
&= \arg\max_{\boldsymbol{\beta}_t} \ln \int_{\mathbf{x}^{(t)}} p\left(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \boldsymbol{\beta}_t\right) d\mathbf{x}^{(t)}.
\end{aligned} \tag{3.4.4}$$

Once we obtain the ML estimate of $\boldsymbol{\beta}_t^\star$, and the associated conditional marginal posterior

$p\left(x_{t,m}|\mathbf{y}^{(t)},\boldsymbol{\beta}_t^\star\right)$, we can obtain the MMSE estimates of $\{x_{t,m}\}$, $\hat{x}_{t,m} = \mathrm{E}\left(x_{t,m}|\mathbf{y}^{(t)},\boldsymbol{\beta}_t^\star\right)$, where the expectation is over the marginal posterior $p\left(x_{t,m}|\mathbf{y}^{(t)},\boldsymbol{\beta}_t^\star\right)$.

It is very challenging to calculate the exact posterior in (3.4.3) because the factor graph of the underlying model in (3.4.3) has loops. In the next section, we propose a D-TOAMP algorithm to approximately calculate the marginal posteriors $\left\{p\left(x_{t,m}|\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)\right\}$ and the associated MMSE estimates, and use an inexact majorization-minimization (MM) method (which is a generalization of the EM method) [27] to find an approximate solution of (3.4.4). One major approximation in the proposed algorithm is that the approximate message passing algorithm (D-TOAMP) may not find the exact posterior in (3.4.3) due to the presence of loops in the associated factor graph. If we assume for fixed $\boldsymbol{\beta}_t$, D-TOAMP can find the exact posterior in (3.4.3), then we an show that the proposed algorithm will converge to a stationary point of the ML problem in (3.4.4) for $\boldsymbol{\beta}_t$. However, the state evolution analysis in [21–23] implies that the approximate posterior of $\mathbf{x}_t$ obtained by the OAMP-based algorithms can be quire accurate. As such, the proposed algorithm is expected to achieve a good performance. Indeed, the proposed D-TOAMP algorithm is shown in the simulations to have significant gain over various state-of-the-art baselines.

Challenge 2: There is no closed-form posterior distribution in (3.4.3) and it is difficult to obtain the closed-form expression of $\boldsymbol{\beta}_t^\star$ in (3.4.4).

3.5 Dynamic Turbo-OAMP Algorithm

The basic idea of the D-TOAMP algorithm is to simultaneously approximate the marginal posterior $p\left(x_{t,m}|\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)$ exploiting the 2D-MM prior and maximize the log-likelihood $\ln p\left(\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)$ with respect to $\boldsymbol{\beta}_t$ as in (3.4.4) at the t-th time slot. In summary, at the t-th time slot, the D-TOAMP algorithm (Algorithm 3.1) performs iterations between the following two major steps until convergence.

- **D-TOAMP-E Step:** Given $\boldsymbol{\beta}_t$, calculate the approximate marginal posterior $\hat{p}\left(x_{t,m}|\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)$ by combining the OAMP and 2D-MM prior via the turbo framework, as elaborated in Section 3.5.2. Then $p\left(\mathbf{x}_t|\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)$ can be approximated by $\hat{p}\left(\mathbf{x}_t|\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right) = \prod_m \hat{p}\left(x_{t,m}|\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)$.
- **D-TOAMP-M Step:** Given $p(\mathbf{x}_t|\mathbf{y}^{(t)},\boldsymbol{\beta}_t) \approx \hat{p}(\mathbf{x}_t|\mathbf{y}^{(t)},\boldsymbol{\beta}_t)$, construct a surrogate function (lower bound) for the objective function $\ln p\left(\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)$, then maximize the surrogate function with respect to $\boldsymbol{\beta}_t$, as elaborated in Section 3.5.1.

In the following, we first elaborate the D-TOAMP-M step, which is an extension of the inexact MM method in [27]. Because the surrogate function in D-TOAMP-M step requires the calculation of the posterior $p\left(\mathbf{x}_t|\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)$, we elaborate how to approximately calculate the posterior $p\left(\mathbf{x}_t|\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)$ in the D-TOAMP-E step.

3.5.1 D-TOAMP-M Step (Inexact MM)

It is difficult to directly maximize the log-likelihood function $\ln p\left(\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)$, because there is no closed-form expression due to the multi-dimensional integration over $\mathbf{x}^{(t)}$ as in (3.4.4). To make the problem tractable, in the D-TOAMP-M Step, we adopt an inexact MM method in [27], which maximizes a surrogate function of $\ln p\left(\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)$ with respect to $\boldsymbol{\beta}_t$. Specifically, let $u(\boldsymbol{\beta}_t;\dot{\boldsymbol{\beta}}_t)$ be the surrogate function constructed at some fixed point $\dot{\boldsymbol{\beta}}_t$,which satisfies the following properties:

$$u(\boldsymbol{\beta}_t;\dot{\boldsymbol{\beta}}_t) \leq \ln p\left(\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right), \forall \boldsymbol{\beta}_t, \tag{3.5.1}$$

$$u(\dot{\boldsymbol{\beta}}_t;\dot{\boldsymbol{\beta}}_t) = \ln p\left(\mathbf{y}^{(t)},\dot{\boldsymbol{\beta}}_t\right), \tag{3.5.2}$$

$$\left.\frac{\partial u(\boldsymbol{\beta}_t;\dot{\boldsymbol{\beta}}_t)}{\partial \boldsymbol{\beta}_t}\right|_{\boldsymbol{\beta}_t=\dot{\boldsymbol{\beta}}_t} = \left.\frac{\partial \ln p\left(\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)}{\partial \boldsymbol{\beta}_t}\right|_{\boldsymbol{\beta}_t=\dot{\boldsymbol{\beta}}_t}. \tag{3.5.3}$$

Inspired by the EM algorithm [30], we use the following surrogate function:

$$u(\boldsymbol{\beta}_t;\dot{\boldsymbol{\beta}}_t) = \int p\left(\mathbf{x}_t|\mathbf{y}^{(t)},\dot{\boldsymbol{\beta}}_t\right) \ln \frac{p\left(\mathbf{x}_t,\mathbf{y}^{(t)},\boldsymbol{\beta}_t\right)}{p\left(\mathbf{x}_t|\mathbf{y}^{(t)},\dot{\boldsymbol{\beta}}_t\right)} d\mathbf{x}_t. \tag{3.5.4}$$

It can be shown that the surrogate function in (3.5.4) satisfies (3.5.1)-(3.5.3). Then in the D-TOAMP-M step of the i-th iteration, we update $\boldsymbol{\beta}_t$ as

$$\boldsymbol{\beta}_t^{i+1} = \arg\max_{\boldsymbol{\beta}_t} u(\boldsymbol{\beta}_t;\boldsymbol{\beta}_t^i), \tag{3.5.5}$$

where $\boldsymbol{\beta}_t^i$ stands for the value of $\boldsymbol{\beta}_t$ at the i-th iteration. However, the maximization problem in (3.5.5) is non-convex and it is difficult to find its optimal solution. Therefore, we use an inexact MM algorithm, where $\boldsymbol{\beta}_t^{i+1}$ is obtained by applying gradient update as follows:

$$\boldsymbol{\beta}_t^{i+1} = \boldsymbol{\beta}_t^i + \Delta^i \cdot \left.\frac{\partial u(\boldsymbol{\beta}_t;\boldsymbol{\beta}_t^i)}{\partial \boldsymbol{\beta}_t}\right|_{\boldsymbol{\beta}_t=\boldsymbol{\beta}_t^i}, \tag{3.5.6}$$

Algorithm 3.1 Dynamic Turbo-OAMP Algorithm

Input: $\{\mathbf{y}_1, \cdots, \mathbf{y}_T\}$, measurement matrix $\mathbf{F}_t$ $(\mathbf{0})$, $\forall t$, and noise variance σ_e^2.
Output: $\{\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_T\}$.
for $t = 1, \cdots, T$ **do**
Initialize: $\mathbf{x}_{A,t}^{pri} = \mathbf{0}$, $v_{A,t}^{pri} = \lambda\sigma^2$, $\boldsymbol{\beta}_t^1 = \mathbf{0}$, $\forall t$, $i = 1$.
while not converge **do**
%D-TOAMP-E Step (for given $\boldsymbol{\beta}_t = \boldsymbol{\beta}_t^i$):
%Module A: LMMSE estimator
1: Update $\mathbf{x}_{A,t}^{post}$ and $v_{A,t}^{post}$ using (3.5.19) and (3.5.20).
2: Update $\mathbf{x}_{B,t}^{pri} = \mathbf{x}_{A,t}^{ext}$ and $v_{B,t}^{pri} = v_{A,t}^{ext}$ using (3.5.22) and (3.5.23).
%Module B: 2D-MM-MMSE estimator (Message Passing over $\mathcal{G}_t$)
3: Message passing over the path $\theta_{t,m} \to f_{t,m}$ using (3.5.25), with $\nu_{d_{t,m}\to\theta_{t,m}}$ as the input.
4: Message passing over the path $x_{t,m} \to f_{t,m} \to s_{t,m}$ using (3.5.27) and (3.5.28), with $\mathbf{x}_{B,t}^{pri}$ and $v_{B,t}^{pri}$ as the input.
5: Hidden support vector estimation in $\mathcal{G}_{s,t}$ through Algorithm 3.2 in Section 3.5.2.3, with $\nu_{s_{t-1,m}\to h_{t,m}}$ as the input.
6: Message passing over the path $s_{t,m} \to f_{t,m} \to x_{t,m}$ using (3.5.30) and (3.5.31).
7: Calculate the posterior distributions $p(x_{t,m}|\mathbf{x}_B^{pri(t)})$ using (3.5.32), and update $\mathbf{x}_{B,t}^{post}$ and $v_{B,t}^{post}$ using (3.5.33) and (3.5.34).
8: Update $\mathbf{x}_{A,t}^{pri} = \mathbf{x}_{B,t}^{ext}$ and $v_{A,t}^{pri} = v_{B,t}^{ext}$ using (3.5.35) and (3.5.36).
9: Repeat Module A and Module B until convergence.
10: Then $\hat{p}(x_{t,m}|\mathbf{y}^{(t)}, \boldsymbol{\beta}_t^i) = \mathcal{CN}(x_{B,t,m}^{post}, v_{B,t}^{post})$. Output $\hat{p}\left(\mathbf{x}_t|\mathbf{y}^{(t)}, \boldsymbol{\beta}_t^i\right) = \mathcal{CN}(\mathbf{x}_{B,t}^{post}, v_{B,t}^{post}\boldsymbol{I})$.
%D-TOAMP-M Step:
11: Construct the surrogate function $\hat{u}(\boldsymbol{\beta}_t; \boldsymbol{\beta}_t^i)$ in (3.5.7) using the approximate posterior output of D-TOAMP-E step, i.e., $\hat{p}\left(\mathbf{x}_t|\mathbf{y}^{(t)}, \boldsymbol{\beta}_t^i\right)$. Then update the off-grid parameter $\boldsymbol{\beta}_t^{i+1}$ as in (3.5.6).
12: $i = i + 1$.
end while
13: Output $\hat{\mathbf{x}}_t = \mathbf{x}_{B,t}^{post}$.
14: Update messages passed to the $(t+1)$-th time slot $\nu_{s_{t,m}\to h_{t+1,m}}(s_{t,m})$ and $\nu_{d_{t+1,m}\to\theta_{t+1,m}}(\theta_{t+1,m})$ using (3.5.37) and (3.5.41).
end for

where Δ^i is the stepsize, which can be determined by backtracking line search [69]. Alternatively, we may use a fixed stepsize as mentioned in [27] to reduce the computational complexity due to backtracking line search.

Based on the convergence proof of the EM algorithm in [70], we can prove that the inexact MM converges to a stationary solution of the optimization problem (3.4.4).

Lemma 3.1 (Convergence of Inexact MM)**.** *Suppose the surrogate function $u(\boldsymbol{\beta}_t; \boldsymbol{\beta}_t^i)$ satisfies (3.5.1)-(3.5.3). If at each iteration, we do inexact (gradient) update as in (3.5.6) for off-grid parameter $\boldsymbol{\beta}_t$, the iterates generated by the D-TOAMP algorithm converge to a stationary point of Problem (3.4.4).*

Therefore, if we can calculate the exact posterior $p\left(\mathbf{x}_t|\mathbf{y}^{(t)}, \dot{\boldsymbol{\beta}}_t\right)$ for given $\dot{\boldsymbol{\beta}}_t$, we can construct the surrogate function in (3.5.4) and the corresponding D-TOAMP algorithm converges to a stationary point of (3.4.4). Unfortunately, in our case, the exact posterior is intractable due to the loops in the factor graph. Thus, in the D-TOAMP-E step, we incorporate the 2D-MM channel prior into the OAMP algorithm to find an approximation of the marginal posterior $p\left(x_{t,m}|\mathbf{y}^{(t)}, \dot{\boldsymbol{\beta}}_t\right)$, i.e., $\hat{p}\left(x_{t,m}|\mathbf{y}^{(t)}, \dot{\boldsymbol{\beta}}_t\right)$ for any given $\dot{\boldsymbol{\beta}}_t$. Then the posterior $p\left(\mathbf{x}_t|\mathbf{y}^{(t)}, \dot{\boldsymbol{\beta}}_t\right)$ can be approximated by $\hat{p}\left(\mathbf{x}_t|\mathbf{y}^{(t)}, \dot{\boldsymbol{\beta}}_t\right) = \prod_m \hat{p}\left(x_{t,m}|\mathbf{y}^{(t)}, \dot{\boldsymbol{\beta}}_t\right)$. Based on the posterior approximation $\hat{p}\left(\mathbf{x}_t|\mathbf{y}^{(t)}, \dot{\boldsymbol{\beta}}_t\right)$, we can construct a tractable surrogate function as

$$\hat{u}(\boldsymbol{\beta}_t; \dot{\boldsymbol{\beta}}_t) = \int \hat{p}\left(\mathbf{x}_t|\mathbf{y}^{(t)}, \dot{\boldsymbol{\beta}}_t\right) \ln \frac{p\left(\mathbf{x}_t, \mathbf{y}^{(t)}, \boldsymbol{\beta}_t\right)}{\hat{p}\left(\mathbf{x}_t|\mathbf{y}^{(t)}, \dot{\boldsymbol{\beta}}_t\right)} d\mathbf{x}_t, \tag{3.5.7}$$

which is expected to approximately satisfy (3.5.1)-(3.5.3). Therefore, after the convergence of the D-TOAMP with the tractable surrogate function in (3.5.7), we not only obtain an approximate stationary solution $\boldsymbol{\beta}_t^*$ of (3.4.4), but also the associated (approximate) marginal conditional posterior $p\left(x_{t,m}|\mathbf{y}^{(t)}, \boldsymbol{\beta}_t^*\right) \approx \hat{p}\left(x_{t,m}|\mathbf{y}^{(t)}, \boldsymbol{\beta}_t^*\right)$.

The detailed update expression for $\boldsymbol{\beta}_t$ could be found in Appendix 3.9.1.

3.5.2 D-TOAMP-E Step

We first give an overview of the OAMP technique that will be used in the algorithm design in this subsection. Then, we elaborate the modules of the D-TOAMP-E step and the message passing in Module B at each time slot in the D-TOAMP-E step.

3.5.2.1 Overview of Orthogonal Approximate Message Passing

OAMP proposed in [21–23] is a variation of the well-known approximate message passing (AMP) [19]. OAMP can handle a wide range of partial orthogonal sensing matrices, and it is shown to achieve a better performance than AMP. Consider the linear observation model

$$\mathbf{y} = \mathbf{F}\mathbf{x} + \mathbf{n}, \tag{3.5.8}$$

where $\mathbf{x} \in \mathbb{C}^{Q\times 1}$ is a sparse signal to be estimated, $\mathbf{y} \in \mathbb{C}^{L\times 1}$ is the received signal, and $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma_e^2 \boldsymbol{I})$ is the Gaussian noise, $\mathbf{F} \in \mathbb{C}^{L\times Q}$ is a partial orthogonal matrix. The entries of the sparse signal $\mathbf{x}$ are assumed to be i.i.d., with the j-th entry following the Bernoulli-Gaussian distribution:

$$x_j \sim \begin{cases} 0 & \text{probability} = 1-\lambda, \\ \mathcal{CN}(0,\xi) & \text{probability} = \lambda. \end{cases} \tag{3.5.9}$$

OAMP is designed to recover the i.i.d. sparse signal $\mathbf{x}$ from the linear observation model in (3.5.8).

OAMP contains two modules: Module A is a linear MMSE (LMMSE) estimator based on the observation and the messages from Module B; Module B performs MMSE estimator that combines the i.i.d sparse prior in (3.5.9) and the messages from Module A. The extrinsic output [71] of a module is fed to the other module as a prior input. The two modules are executed iteratively until convergence. At the end, the estimation of $\mathbf{x}$ is given based on the posterior output of Module B.

Specifically, for Module A, it is based on the assumption that the entries of $\mathbf{x}$ are i.i.d with a prior mean $\mathbf{x}_A^{pri}$ and variance v_A^{pri}, where $\mathbf{x}_A^{pri}$ and v_A^{pri} are the messages passed from Module B. Then under this assumption, the LMMSE estimate and the MSE of $\mathbf{x}$ based on model (3.5.8) are respectively given by [23]

$$\mathbf{x}_A^{post} = \mathbf{x}_A^{pri} + \frac{v_A^{pri}}{v_A^{pri} + \sigma_e^2}\mathbf{F}^H(\mathbf{y} - \mathbf{F}\mathbf{x}_A^{pri}), \tag{3.5.10}$$

$$v_A^{post} = v_A^{pri} - \frac{L}{Q}\cdot\frac{(v_A^{pri})^2}{v_A^{pri} + \sigma_e^2}. \tag{3.5.11}$$

The extrinsic LMMSE estimate and the MSE of $\mathbf{x}$ can be computed by [23]

$$\mathbf{x}_B^{pri} = \mathbf{x}_A^{ext} = v_A^{ext}\left(\frac{\mathbf{x}_A^{post}}{v_A^{post}} - \frac{\mathbf{x}_A^{pri}}{v_A^{pri}}\right), \tag{3.5.12}$$

$$v_B^{pri} = v_A^{ext} = \left(\frac{1}{v_A^{post}} - \frac{1}{v_A^{pri}}\right)^{-1}, \tag{3.5.13}$$

which is passed to Module B as its input messages.

For Module B, it is based on the assumption that $\mathbf{x}_B^{pri}$ is modeled as an AWGN observation of $\mathbf{x}$, i.e.,

$$\mathbf{x}_B^{pri} = \mathbf{x} + \mathbf{z}, \tag{3.5.14}$$

where $\mathbf{z} \sim \mathcal{CN}\left(0, v_B^{pri}\boldsymbol{I}\right)$ is independent of $\mathbf{x}$. Based on this assumption, the posterior mean and variance can be respectively calculated as [23]

$$x_{B,j}^{post} = \mathrm{E}\left(x_j|\mathbf{x}_B^{pri}\right) = \mathrm{E}\left(x_j|x_{B,j}^{pri}\right), \tag{3.5.15}$$

$$v_B^{post} = \frac{1}{Q}\sum_{j=1}^{Q}\mathrm{Var}\left(x_j|x_{B,j}^{pri}\right) = \frac{1}{Q}\sum_{j=1}^{Q}\mathrm{E}\left(\left|x_j - \mathrm{E}\left(x_j|x_{B,j}^{pri}\right)\right|^2\right), \tag{3.5.16}$$

where $x_{B,j}^{post}$ and $x_{B,j}^{pri}$ denote the j-th entry of $\mathbf{x}_B^{post}$ and $\mathbf{x}_B^{pri}$. $\mathrm{E}(\cdot)$ is with respect to the posterior distribution of $\mathbf{x}$, which is given by $p\left(x_j\left|x_{B,j}^{pri}\right.\right) \propto p\left(x_{B,j}^{pri}\left|x_j\right.\right)p\left(x_j\right)$, where $p\left(x_j\right)$ is given in (3.5.9). Based on (3.5.14), $p\left(x_{B,j}^{pri}\left|x_j\right.\right)$ is given by $\mathcal{CN}\left(x_{B,j}^{pri}; x_j, v_B^{pri}\right)$. Note that as the entries of $\mathbf{x}$ are prior independent (from (3.5.9)) and according to the assumption in (3.5.14), the entries of $\mathbf{x}$ are also posterior independent. The extrinsic mean and variance of $\mathbf{x}$ can be computed by [23]

$$\mathbf{x}_A^{pri} = \mathbf{x}_B^{ext} = v_B^{ext}\left(\frac{\mathbf{x}_B^{post}}{v_B^{post}} - \frac{\mathbf{x}_B^{pri}}{v_B^{pri}}\right), \tag{3.5.17}$$

$$v_A^{pri} = v_B^{ext} = \left(\frac{1}{v_B^{post}} - \frac{1}{v_B^{pri}}\right)^{-1}. \tag{3.5.18}$$

3.5.2.2 Modules of the D-TOAMP-E Step within a Time Slot

At each time slot t, for given β_t, the D-TOAMP-E step contains two modules (as shown in Fig. 3.7): Module A is a LMMSE estimator based on the current observation $\mathbf{y}_t$ and messages from Module B. Module B performs MMSE estimation that combines the 2D-MM channel prior in (3.3.3), the messages from Module A and the messages passed from the last time slot (for $t > 1$). The two modules are executed iteratively until convergence. Because in the

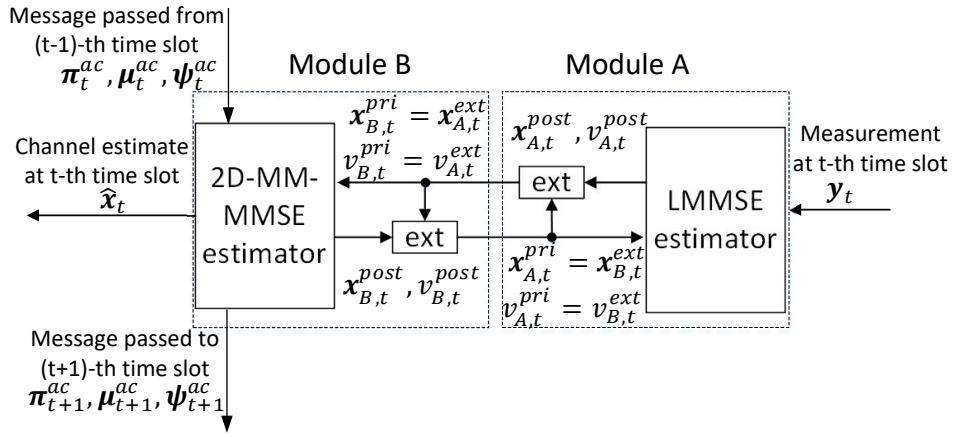

Figure 3.7: Modules of the D-TOAMP algorithm

D-TOAMP-E step, $\boldsymbol{\beta}_t$ is fixed, we will omit the argument $\boldsymbol{\beta}_t$ in the probability expressions for simplicity.

We now provide more details of each module. In Module A, the channel vector $\mathbf{x}_t$ is estimated based on the observation $\mathbf{y}_t$ with a prior distribution $\mathcal{CN}(\mathbf{x}_t; \mathbf{x}_{A,t}^{pri}, v_{A,t}^{pri}\boldsymbol{I})$, where $\mathbf{x}_{A,t}^{pri}$ and $v_{A,t}^{pri}$ are the extrinsic mean and variance, respectively, from the 2D-MM-MMSE estimator that will be elaborated in detail later. Then the posterior distribution of $\mathbf{x}_t$ is still complex Gaussian with mean and variance given by

$$\mathbf{x}_{A,t}^{post} = \mathbf{x}_{A,t}^{pri} + \frac{v_{A,t}^{pri}}{v_{A,t}^{pri} + \sigma_e^2}\mathbf{F}_t\left(\boldsymbol{\beta}_t\right)^H\left(\mathbf{y}_t - \mathbf{F}_t\left(\boldsymbol{\beta}_t\right)\mathbf{x}_{A,t}^{pri}\right) \tag{3.5.19}$$

and

$$v_{A,t}^{post} = v_{A,t}^{pri} - \frac{P}{M}\cdot\frac{(v_{A,t}^{pri})^2}{v_{A,t}^{pri} + \sigma_e^2}, \tag{3.5.20}$$

respectively. After that, we need to calculate the extrinsic message passing [71], which can decorrelate the input and output messages of the estimator. The extrinsic distribution of $\mathbf{x}_t$ satisfies

$$\mathcal{CN}(\mathbf{x}_t; \mathbf{x}_{A,t}^{post}, v_{A,t}^{post}\boldsymbol{I}) \propto \mathcal{CN}(\mathbf{x}_t; \mathbf{x}_{A,t}^{pri}, v_{A,t}^{pri}\boldsymbol{I})\mathcal{CN}(\mathbf{x}_t; \mathbf{x}_{A,t}^{ext}, v_{A,t}^{ext}\boldsymbol{I}). \tag{3.5.21}$$

Therefore, the extrinsic mean and variance are respectively given by

$$\mathbf{x}_{B,t}^{pri} = \mathbf{x}_{A,t}^{ext} = v_{A,t}^{ext}\left(\frac{\mathbf{x}_{A,t}^{post}}{v_{A,t}^{post}} - \frac{\mathbf{x}_{A,t}^{pri}}{v_{A,t}^{pri}}\right), \tag{3.5.22}$$

$$v_{B,t}^{pri} = v_{A,t}^{ext} = (\frac{1}{v_{A,t}^{post}} - \frac{1}{v_{A,t}^{pri}})^{-1}. \tag{3.5.23}$$

In Module B, the extrinsic calculation is similar to that in Module A, but the 2D-MM-MMSE estimator is more complicated.

Challenge 3: MMSE estimator design for the 2D-MM priors.

In OAMP [23], the MMSE estimator was designed for i.i.d. prior. However, in this chapter, the massive MIMO channels with 2D dynamic sparsity are not i.i.d distributed. The MMSE estimator needs to be redesigned based on the proposed 2D-MM channel prior. Therefore, the standard MMSE estimator for i.i.d. prior cannot be applied and we need to extend the MMSE estimator in order to exploit the 2D dynamic sparsity structure in the massive MIMO channels in (3.3.3). The details of the 2D-MM-MMSE estimator and the corresponding extrinsic update are presented in Section 3.5.2.3.

3.5.2.3 Message Passing in Module B (2D-MM-MMSE estimator)

In this subsection, we explain the details of Module B for 2D-MM channel priors at time slot t, $1 \leq t \leq T$. A basic assumption is to model $\mathbf{x}_{B,t}^{pri}$, the extrinsic mean from the LMMSE estimator as given in (3.5.22), as an AWGN observation [23], i.e.,

$$\mathbf{x}_{B,t}^{pri} = \mathbf{x}_t + \mathbf{z}_t, \tag{3.5.24}$$

where $\mathbf{z}_t \sim \mathcal{CN}(\mathbf{0}, v_{B,t}^{pri}\boldsymbol{I})$ is independent of $\mathbf{x}_t$, and $v_{B,t}^{pri}$ is the extrinsic variance from the LMMSE estimator as given in (3.5.23). Similar assumptions have been widely used in message-passing-based iterative signal recovery algorithms [19, 22, 23, 25]. A formal proof of Assumption (3.5.24) has been provided in [41] for the AMP with an i.i.d. Gaussian measurement matrix. Extensive simulations have also been conducted in [22, 23, 25] to verify the validity of Assumption (3.5.24) for the OAMP. The main advantage of replacing the original observation model in (3.4.1) with the approximate AWGN observation model in (3.5.24) is that the per iteration complexity of the message passing algorithm can be reduced from $O(PM)$ to only $O(M)$.

Denote the collection of measurement vectors in AWGN observation model as $\mathbf{x}_B^{pri(T)} = \left\{\mathbf{x}_{B,t}^{pri}\right\}_{t=1}^{T}$. Under Assumption (3.5.24), the factor graph of joint distribution $p(\mathbf{x}_B^{pri(T)}, \mathbf{x}^{(T)}, \mathbf{s}^{(T)}, \boldsymbol{\theta}^{(T)})$, denoted by $\mathcal{G}$, is shown in Fig. 3.8, where the function expression of each factor node is

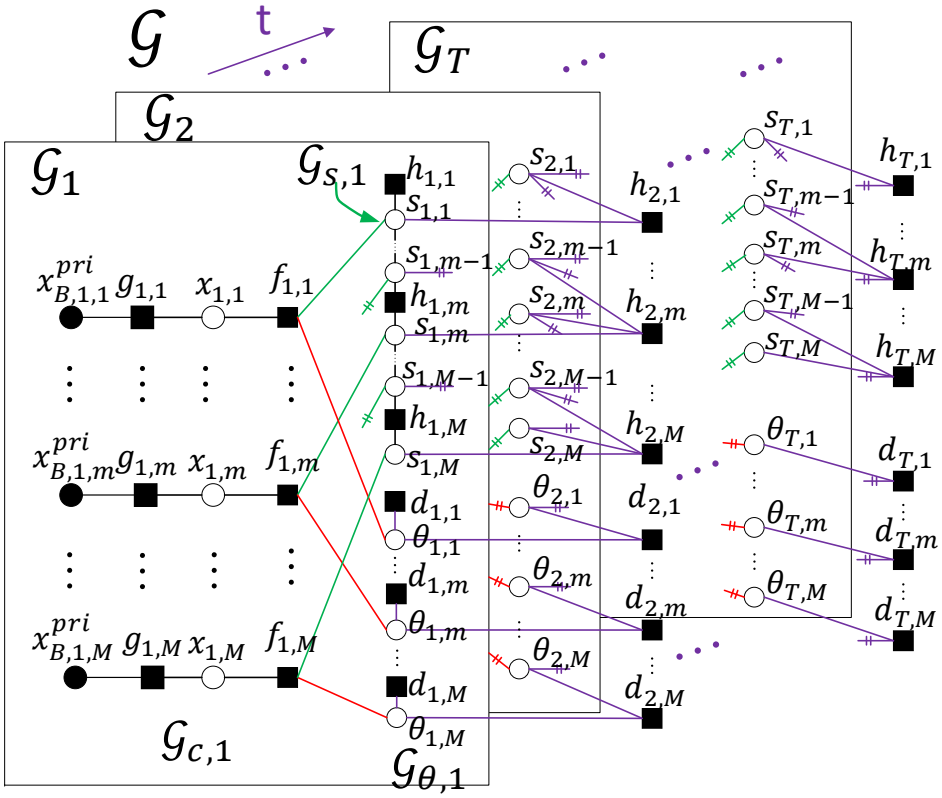

Figure 3.8: Factor graph of the D-TOAMP

Table 3.1: Factors, distributions and functional forms in our signal model

Factor	Distribution	Functional form
$g_{t,m}\left(x_{t,m}, x_{B,t,m}^{pri}\right)$	$p\left(x_{B,t,m}^{pri}\middle\vert x_{t,m}\right)$	$\mathcal{CN}\left(x_{t,m}; x_{B,t,m}^{pri}, v_{B,t}^{pri}\right)$
$f_{t,m}\left(x_{t,m}, s_{t,m}, \theta_{t,m}\right)$	$p\left(x_{t,m}\middle\vert s_{t,m}, \theta_{t,m}\right)$	$\delta\left(x_{t,m} - \theta_{t,m} s_{t,m}\right)$
$h_{1,1}\left(s_{1,1}\right)$	$p\left(s_{1,1}\right)$	$(1-\lambda)^{1-s_{1,1}}\left(\lambda\right)^{s_{1,1}}$
$h_{1,m}\left(s_{1,m}, s_{1,m-1}\right)$	$p\left(s_{1,m}\middle\vert s_{1,m-1}\right)$	$\begin{cases}\left(\rho_{01}^{S}\right)^{s_{1,m}}\left(1-\rho_{01}^{S}\right)^{1-s_{1,m}}, & s_{1,m-1}=0\\ \left(1-\rho_{10}^{S}\right)^{s_{1,m}}\left(\rho_{10}^{S}\right)^{1-s_{1,m}} & s_{1,m-1}=1\end{cases}$
$h_{t,1}\left(s_{t,1}, s_{t-1,1}\right)$	$p\left(s_{t,1}\middle\vert s_{t-1,1}\right)$	$\begin{cases}\left(\rho_{01}^{T}\right)^{s_{t,1}}\left(1-\rho_{01}^{T}\right)^{1-s_{t,1}}, & s_{t-1,1}=0\\ \left(1-\rho_{10}^{T}\right)^{s_{t,1}}\left(\rho_{10}^{T}\right)^{1-s_{t,1}} & s_{t-1,1}=1\end{cases}$
$h_{t,m}\left(s_{t,m}, s_{t,m-1}, s_{t-1,m}\right)$	$p\left(s_{t,m}\middle\vert s_{t,m-1}, s_{t-1,m}\right)$	$\left(\rho_{bc1}\right)^{s_{t,m}}\left(1-\rho_{bc1}\right)^{1-s_{t,m}}, s_{t,m-1}=b, s_{t-1,m}=c, b, c\in\{0,1\}$
$d_{1,m}\left(\theta_{1,m}\right)$	$p\left(\theta_{1,m}\right)$	$\mathcal{CN}\left(\theta_{1,m}; \zeta, \sigma^2\right)$
$d_{t,m}\left(\theta_{t,m}, \theta_{t-1,m}\right)$	$p\left(\theta_{t,m}\middle\vert \theta_{t-1,m}\right)$	$\mathcal{CN}\left(\theta_{t,m}; \left(1-\alpha\right)\theta_{t-1,m}+\alpha\zeta, \alpha^2\kappa\right)$

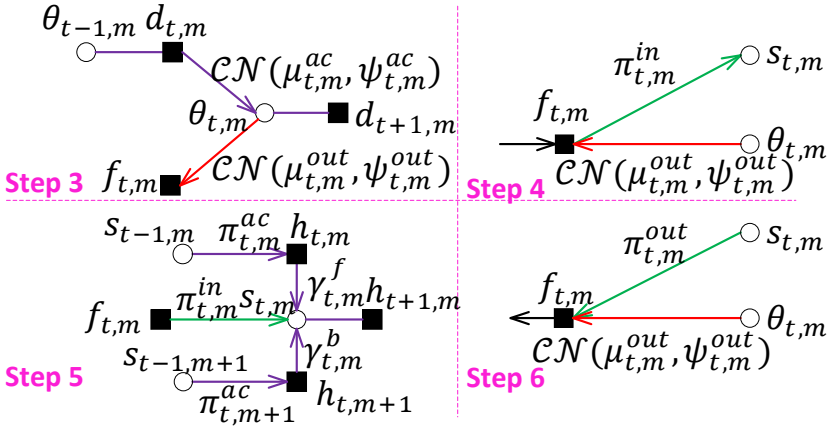

Figure 3.9: Message passing of Step 3-6 in Algorithm 3.1

listed in Table 3.1. At time slot t, the factor graph denoted by $\mathcal{G}_t$ can be decomposed into three parts: the channel coefficient subgraph $\mathcal{G}_{c,t}$ (which represents the AWGN measurement model (3.5.24)); the hidden support subgraph $\mathcal{G}_{s,t}$ (which represents the 2D-MM $\mathbf{s}_t$); and the hidden value subgraph $\mathcal{G}_{\theta,t}$ (which represents the Gauss-Markov $\boldsymbol{\theta}_t$). Module B aims at calculating the posterior distributions $\left\{p\left(x_{t,m}|\mathbf{x}_B^{pri(t)}\right)\right\}$ using the SPMP rule.

The message passing order at time slot t is elaborated in Fig. 3.9. This figure also introduces the notations that we adopt to parameterize the different messages. For hidden support variables $s_{t,m}$ with a binary distribution, the associated message is represented by its nonzero probability, e.g., $\pi^{in}_{t,m} = \nu_{f_{t,m}\to s_{t,m}}(s_{t,m}=1)$. For hidden value variable $\theta_{t,m}$ with a complex Gaussian distribution, the associated message is parameterized using its mean and variance, i.e., $\nu_{\theta_{t,m}\to f_{t,m}}(\theta_{t,m}) = \mathcal{CN}\left(\theta_{t,m}; \mu^{out}_{t,m}, \psi^{out}_{t,m}\right)$. Note that at the first time slot, the variable nodes $s_{t-1,m}, \theta_{t-1,m}$ and associated edges will be removed. At time slot T, the factor nodes $h_{t+1,m}, d_{t+1,m}$ and associated edges will be removed. The details are elaborated as follows.

Step 3 in Algorithm 3.1: Based on the prior information passed from the last time slot $\nu_{d_{t,m}\to\theta_{t,m}} = \mathcal{CN}\left(\theta_{t,m}; \mu^{ac}_{t,m}, \psi^{ac}_{t,m}\right)$, the message passed from variable node $\theta_{t,m}$ to factor node $f_{t,m}$ can be calculated as

$$\nu_{\theta_{t,m}\to f_{t,m}}(\theta_{t,m}) = \mathcal{CN}\left(\theta_{t,m}; \mu^{out}_{t,m}, \psi^{out}_{t,m}\right), \tag{3.5.25}$$

where

$$\left(\mu^{out}_{t,m}, \psi^{out}_{t,m}\right) = \left(\mu^{ac}_{t,m}, \psi^{ac}_{t,m}\right). \tag{3.5.26}$$

Note that since the proposed algorithm is a recursive algorithm, there are no messages passed from $d_{t+1,m}$ due to causality. If $t=1$, we set $\left(\mu^{ac}_{1,m}, \psi^{ac}_{1,m}\right) = (\zeta, \sigma^2)$.

Step 4 in Algorithm 3.1: According to the sum-product rule, the message from variable node $x_{t,m}$ to factor node $f_{t,m}$ is

$$\nu_{x_{t,m}\to f_{t,m}}\left(x_{t,m}\right)=\mathcal{CN}\left(x_{t,m};x_{B,t,m}^{pri},v_{B,t}^{pri}\right). \tag{3.5.27}$$

The message from factor node $f_{t,m}$ to variable node $s_{t,m}$ is

$$\begin{aligned}&\nu_{f_{t,m}\to s_{t,m}}\left(s_{t,m}\right)\\=&\int_x\int_\theta\nu_{\theta_{t,m}\to f_{t,m}}\left(\theta\right)\nu_{x_{t,m}\to f_{t,m}}\left(x\right)\delta\left(x-\theta s_{t,m}\right)d\theta dx\\=&\pi_{t,m}^{in}\delta\left(s_{t,m}-1\right)+\left(1-\pi_{t,m}^{in}\right)\delta\left(s_{t,m}\right),\end{aligned} \tag{3.5.28}$$

where

$$\pi_{t,m}^{in}=\left(1+\frac{\mathcal{CN}\left(0;x_{B,t,m}^{pri},v_{B,t}^{pri}\right)}{\mathcal{CN}\left(0;x_{B,t,m}^{pri}-\mu_{t,m}^{out},v_{B,t}^{pri}+\psi_{t,m}^{out}\right)}\right)^{-1}. \tag{3.5.29}$$

Step 5 in Algorithm 3.1: A forward-backward message passing is performed over the 2D-MM of $\mathbf{s}_t$ given input messages $\left\{\nu_{f_{t,m}\to s_{t,m}}(s_{t,m})\right\}$, $\left\{\nu_{s_{t-1,m}\to h_{t,m}}(s_{t-1,m})\right\}$. Details are summarized in Algorithm 3.2. Note that there is no message passed from $h_{t+1,m}$ due to causality.

Step 6 in Algorithm 3.1: After this, the message from variable node $s_{t,m}$ to factor node $f_{t,m}$ can be calculated as

$$\nu_{s_{t,m}\to f_{t,m}}\left(s_{t,m}\right)=\pi_{t,m}^{out}\delta\left(s_{t,m}-1\right)+\left(1-\pi_{t,m}^{out}\right)\delta\left(s_{t,m}\right), \tag{3.5.30}$$

where $\pi_{t,m}^{out}$ is given by the output of Algorithm 3.2. The message from factor node $f_{t,m}$ back to variable node $x_{t,m}$ is

$$\nu_{f_{t,m}\to x_{t,m}}\left(x_{t,m}\right)=\pi_{t,m}^{out}\mathcal{CN}\left(x_{t,m};\mu_{t,m}^{out},\psi_{t,m}^{out}\right)+\left(1-\pi_{t,m}^{out}\right)\delta\left(x_{t,m}\right). \tag{3.5.31}$$

After calculating the updated messages $\left\{\nu_{f_{t,m}\to x_{t,m}}\right\}$, the posterior distributions are given by

$$p\left(x_{t,m}|\mathbf{x}_B^{pri(t)}\right)\propto\nu_{f_{t,m}\to x_{t,m}}\left(x_{t,m}\right)\nu_{g_{t,m}\to x_{t,m}}\left(x_{t,m}\right), \tag{3.5.32}$$

Algorithm 3.2 Channel Support Estimation Procedure

Input: $\pi_{t,m}^{ac}, \pi_{t,m}^{in}, \forall m$
Output: $\pi_{t,m}^{out}, \forall m$
If $t = 1$
Initialize: $\gamma_{t,1}^{f} = \lambda$, $\gamma_{t,M}^{b} = \frac{1}{2}$
for $m = 2, \cdots, M$
$\gamma_{t,m}^{f} = \frac{\rho_{01}^{S}(1-\pi_{t,m-1}^{in})(1-\gamma_{t,m-1}^{f})+\rho_{11}^{S}\pi_{t,m-1}^{in}\gamma_{t,m-1}^{f}}{(1-\pi_{t,m-1}^{in})(1-\gamma_{t,m-1}^{f})+\pi_{t,m-1}^{in}\gamma_{t,m-1}^{f}}$.
for $m = 1, \cdots, M-1$
$\gamma_{t,m}^{b} = \frac{\rho_{10}^{S}(1-\pi_{t,m+1}^{in})(1-\gamma_{t,m+1}^{b})+(1-\rho_{10}^{S})\pi_{t,m+1}^{in}\gamma_{t,m+1}^{b}}{(\rho_{00}^{S}+\rho_{10}^{S})(1-\pi_{t,m+1}^{in})(1-\gamma_{t,m+1}^{b})+(\rho_{11}^{S}+\rho_{01}^{S})\pi_{t,m+1}^{in}\gamma_{t,m+1}^{b}}$.
Else If $t > 1$
Initialize: $\gamma_{t,1}^{f} = \rho_{01}^{T}\left(1-\pi_{t,1}^{ac}\right) + \left(1-\rho_{10}^{T}\right)\pi_{t,1}^{ac}$, $\gamma_{t,M}^{b} = \frac{1}{2}$.
for $m = 2, \cdots, M$
$\gamma_{t,m}^{f} = \frac{\rho_{111}\pi_{t,m}^{ac}+\rho_{101}\left(1-\pi_{t,m}^{ac}\right)}{1+\left(\left(\pi_{t,m-1}^{in}\right)^{-1}-1\right)\left(\left(\gamma_{t,m-1}^{f}\right)^{-1}-1\right)} + \frac{\rho_{011}\pi_{t,m}^{ac}+\rho_{001}\left(1-\pi_{t,m}^{ac}\right)}{\left(\left(\pi_{t,m-1}^{in}\right)^{-1}-1\right)^{-1}\left(\left(\gamma_{t,m-1}^{f}\right)^{-1}-1\right)^{-1}+1}$.
for $m = 1, \cdots, M-1$
$\gamma_{t,m}^{b} = \frac{1}{1+\gamma}$, where $\gamma = \frac{\left(1-\pi_{t,m+1}^{in}-\gamma_{t,m+1}^{b}\right)\left[\rho_{010}\pi_{t,m+1}^{ac}+\rho_{000}\left(1-\pi_{t,m+1}^{ac}\right)\right]+\pi_{t,m+1}^{in}\gamma_{t,m+1}^{b}}{\left(1-\pi_{t,m+1}^{in}-\gamma_{t,m+1}^{b}\right)\left[\rho_{110}\pi_{t,m+1}^{ac}+\rho_{100}\left(1-\pi_{t,m+1}^{ac}\right)\right]+\pi_{t,m+1}^{in}\gamma_{t,m+1}^{b}}$.
end
Then $\pi_{t,m}^{out} = \frac{\gamma_{t,m}^{f}\gamma_{t,m}^{b}}{\gamma_{t,m}^{f}\gamma_{t,m}^{b}+\left(1-\gamma_{t,m}^{f}\right)\left(1-\gamma_{t,m}^{b}\right)}$.

where $\nu_{g_{t,m}\to x_{t,m}}(x_{t,m}) = \mathcal{CN}(x_{t,m}; x_{B,t,m}^{pri}, v_{B,t}^{pri})$. Then the posterior mean and variance can be respectively calculated as

$$x_{B,t,m}^{post} = \mathrm{E}\left(x_{t,m}|\mathbf{x}_{B}^{pri(t)}\right) = \int_{x_{t,m}} x_{t,m} p\left(x_{t,m}|\mathbf{x}_{B}^{pri(t)}\right) \tag{3.5.33}$$

and

$$v_{B,t}^{post} = \frac{1}{M}\sum_{m=1}^{M} \mathrm{Var}\left(x_{t,m}|\mathbf{x}_{B}^{pri(t)}\right) = \frac{1}{M}\sum_{m=1}^{M}\int_{x_{t,m}} \left|x_{t,m} - \mathrm{E}\left(x_{t,m}|\mathbf{x}_{B}^{pri(t)}\right)\right|^{2} p(x_{t,m}|\mathbf{x}_{B}^{pri(t)}). \tag{3.5.34}$$

Based on the derivation in [23], the corresponding extrinsic update can be calculated as

$$\mathbf{x}_{A,t}^{pri} = \mathbf{x}_{B,t}^{ext} = v_{A,t}^{pri}\left(\frac{\mathbf{x}_{B,t}^{post}}{v_{B,t}^{post}} - \frac{\mathbf{x}_{B,t}^{pri}}{v_{B,t}^{pri}}\right), \tag{3.5.35}$$

$$v_{A,t}^{pri} = v_{B,t}^{ext} = \left(\frac{1}{v_{B,t}^{post}} - \frac{1}{v_{B,t}^{pri}}\right)^{-1}. \tag{3.5.36}$$

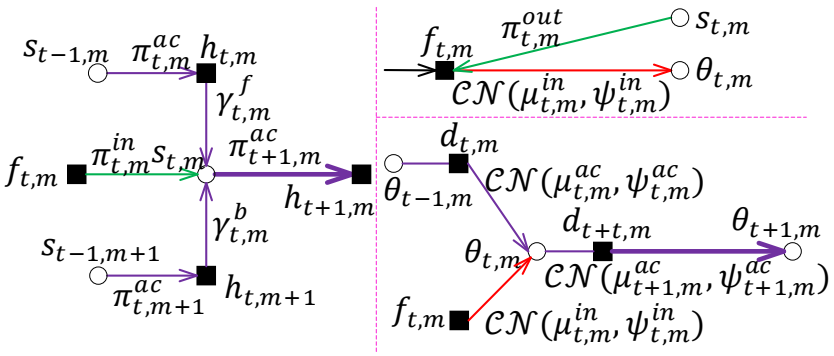

Figure 3.10: Message passing across time slots

3.5.3 Message Passing Across Time Slots

After the convergence of the message passing over $\mathcal{G}_t$, we forward the latest messages about $\mathbf{s}_t$ and $\boldsymbol{\theta}_t$ to the next time slot to provide prior information about $\mathbf{s}_{t+1}$ and $\boldsymbol{\theta}_{t+1}$. The message passing procedure is shown in Fig. 3.10. We firstly calculate channel support prior information passed to the next time slot $\nu_{s_{t,m}\to h_{t+1,m}}(s_{t,m})$ as follows:

$$\begin{aligned}
&\nu_{s_{t,m}\to h_{t+1,m}}(s_{t,m}) \\
=&\nu_{h_{t,m}\to s_{t,m}}(s_{t,m})\nu_{h_{t,m+1}\to s_{t,m}}(s_{t,m})\nu_{f_{t,m}\to s_{t,m}}(s_{t,m}) \\
=&\pi^{ac}_{t+1,m}\delta(s_{t,m}-1)+(1-\pi^{ac}_{t+1,m})\delta(s_{t,m}),
\end{aligned} \tag{3.5.37}$$

where $\pi^{ac}_{t+1,m}$ is given by

$$\pi^{ac}_{t+1,m}=\frac{\gamma^{f}_{t,m}\gamma^{b}_{t,m}\pi^{in}_{t,m}}{\gamma^{f}_{t,m}\gamma^{b}_{t,m}\pi^{in}_{t,m}+\left(1-\gamma^{f}_{t,m}\right)\left(1-\gamma^{b}_{t,m}\right)\left(1-\pi^{in}_{t,m}\right)}.$$

In order to calculate the hidden value vector prior information passed to next time slot, we firstly calculate the message from factor node $f_{t,m}$ to variable node $\theta_{t,m}$ based on the latest information as follows:

$$\begin{aligned}
&\nu^{\text{exact}}_{f_{t,m}\to\theta_{t,m}}(\theta_{t,m}) \\
=&\int_x\sum_s\nu_{s_{t,m}\to f_{t,m}}(s)\,\nu_{x_{t,m}\to f_{t,m}}(x)\,\delta(x-\theta_{t,m}s)\,dx \\
=&\pi^{out}_{t,m}\mathcal{CN}\left(\theta_{t,m};x^{pri}_{B,t,m},v^{pri}_{B,t}\right)+\left(1-\pi^{out}_{t,m}\right)\mathcal{CN}\left(0;x^{pri}_{B,t,m},v^{pri}_{B,t}\right).
\end{aligned} \tag{3.5.38}$$

This is an inappropriate message, because $\mathcal{CN}\left(0;x^{pri}_{B,t,m},v^{pri}_{B,t}\right)$ which is irrelevant to $\theta_{t,m}$

prevents us from normalizing it. Intuitively, $\nu_{f_{t,m}\to\theta_{t,m}}(\theta_{t,m})$ conveys the information about hidden values $\theta_{t,m}$ based on the channel support $s_{t,m}$ and actual channel coefficient $x_{t,m}$. If $s_{t,m}=0$, then by (3.3.2), $x_{t,m}=0$. We can not obtain any useful information about $\theta_{t,m}$, which makes $\theta_{t,m}$ unobservable. The constant term in (3.5.38) reveals the uncertainty caused by this unobservability through an infinitely broad and uninformative distribution of $\theta_{t,m}$. A similar problem also occurs in [66], we adopt the similar approach to solve it by introducing a threshold. The resulting proper message is given by

$$\nu_{f_{t,m}\to\theta_{t,m}}(\theta_{t,m})=\mathcal{CN}\left(\theta_{t,m};\mu_{t,m}^{in},\psi_{t,m}^{in}\right) \tag{3.5.39}$$

where

$$\left(\mu_{t,m}^{in},\psi_{t,m}^{in}\right)=\begin{cases}\left(\frac{1}{\epsilon}x_{B,t,m}^{pri},\frac{1}{\epsilon^2}v_{B,t}^{pri}\right) & \pi_{t,m}^{out}\leq \mathrm{Thr},\\ \left(x_{B,t,m}^{pri},v_{B,t}^{pri}\right) & \pi_{t,m}^{out}>\mathrm{Thr},\end{cases} \tag{3.5.40}$$

the threshold Thr is slightly less than 1 and ϵ is close to 0. Then the prior information about the hidden value vector passed to the next time slot can be calculated as follows:

$$\nu_{d_{t+1,m}\to\theta_{t+1,m}}(\theta_{t+1,m})=\mathcal{CN}\left(\theta_{t+1,m};\mu_{t+1,m}^{ac},\psi_{t+1,m}^{ac}\right), \tag{3.5.41}$$

where

$$\mu_{t+1,m}^{ac}=(1-\alpha)\left(\frac{\psi_{t,m}^{ac}\psi_{t,m}^{in}}{\psi_{t,m}^{ac}+\psi_{t,m}^{in}}\right)\left(\frac{\mu_{t,m}^{ac}}{\psi_{t,m,n}^{ac}}+\frac{\mu_{t,m}^{in}}{\psi_{t,m}^{in}}\right)+\alpha\zeta,$$

$$\psi_{t+1,m}^{ac}=(1-\alpha)^2\left(\frac{\psi_{t,m}^{ac}\psi_{t,m}^{in}}{\psi_{t,m}^{ac}+\psi_{t,m}^{in}}\right)+\alpha^2\kappa.$$

Finally, the overall D-TOAMP algorithm is summarized in Algorithm 3.1. Note that the proposed D-TOAMP can also be applied to a general array geometry at the BS, by replacing the array response matrix $\mathbf{A}(\boldsymbol{\beta})$ with $\mathbf{A}(\boldsymbol{\beta},\hat{\boldsymbol{\varphi}})$, and adding an additional gradient update for $\hat{\boldsymbol{\varphi}}$. The details are omitted for conciseness.

3.5.4 Complexity Analysis

The computational complexity of the proposed algorithm is analyzed as follows.

- Complexity of Module A (LMMSE estimator): The computational complexity of LMMSE is dominated by the matrix multiplication, whose complexity is $\mathcal{O}(PM)$.

- Complexity of Module B (2D-MM-MMSE estimator): This module is used to handle the 2D-MM prior (capture the 2D dynamic sparsity of a MIMO channel). Since this module is just a simple sum-product algorithm over a tree graph, and each message could be parameterized by one or two variables, the complexity is very low, which is $\mathcal{O}(M)$.

- Complexity of parameter update: The complexity of updating the off-grid parameter $\boldsymbol{\beta}_t$ is $\mathcal{O}(PM^2)$ per iteration if the fixed stepsize is used.

This suggests the total computational requirement of the proposed method with off-grid model is $\mathcal{O}\left(PM^2\right)$ per iteration. Empirical evidence shows that the proposed method usually converges within 30 iterations[5]. Fig. 3.11 shows the simulation time of various schemes versus the number of antennas for not exactly sparse signals (the AoDs do not exactly lie on the grid points). The details of each baseline scheme are introduced in Section 3.6. It shows that the complexity of the proposed D-TOAMP without grid refinement is comparable to the message-passing-based algorithms with i.i.d. or structured priors, such as OAMP [23], DCS-AMP [66] and Structured Turbo-CS [25], and is lower than that of some popular CS algorithms, such as Modified SP [36], SBL [27] which require matrix multiplication and matrix inversion with high computational complexity, and Burst-LASSO [7] which needs to solve a large dimensional optimization problem. With off-grid basis, the off-grid parameter updating will introduce more complexity in order to eliminate the grid mismatch. However, the off-grid based D-TOAMP could achieve much better performance compared to the D-TOAMP without considering the grid mismatch, as shown in Section 3.6.

3.6 Simulation Results

In this section, we evaluate the performance of the proposed algorithm under two widely used channel models which are not exactly sparse. The spatial channel model (SCM) [72] is developed in 3GPP/3GPP2 for low frequency bands (less than 6 GHz) and has been widely used to evaluate the performance of LTE systems. We consider urban microcell environment,

[5]It is difficult to theoretically analyze the convergence rate of the proposed algorithm. Based on the numerical studies, there are several factors that would affect the convergence rate of D-TOAMP. For example, if the temporal correlation of massive MIMO channels is stronger, the prior information provided to the next time slot would be more accurate, and the D-TOAMP algorithm executed at the next time slot would converge to the stationary points much faster. The initial values of the statistical parameters $\boldsymbol{\rho}$ would also affect the convergence rate. The more accurate the initial values of $\boldsymbol{\rho}$ are, the faster the algorithm would converge to the optimal values.

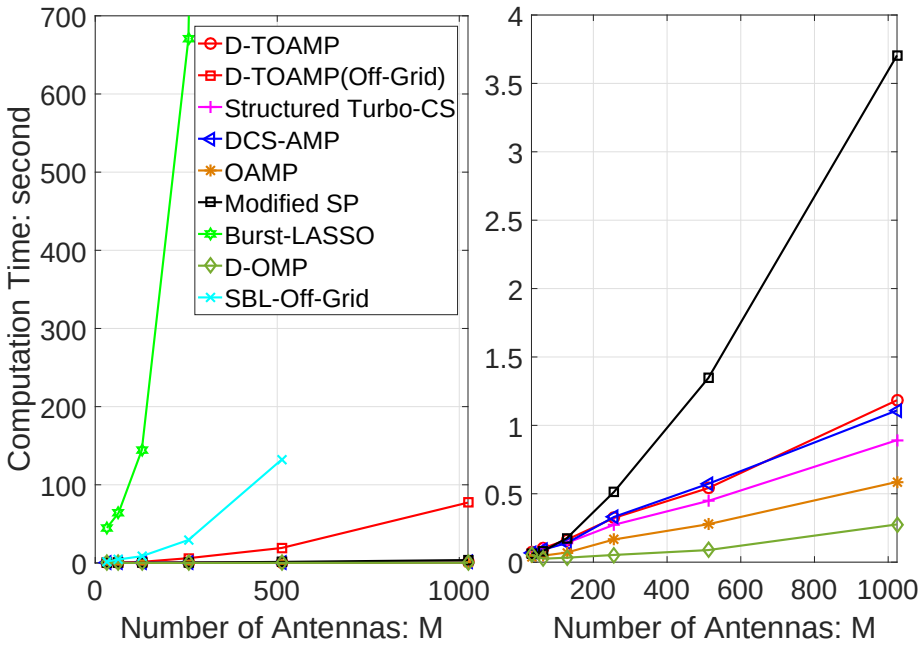

Figure 3.11: Computation time of various schemes versus the number of antennas for not exactly sparse signals. Set $P/M = 0.18$, $T = 50$, SNR$= 15dB$, $\lambda = 0.125$, $\rho_{01}^{S} = 0.025$, $\rho_{10}^{S} = 0.175$, $\rho_{01}^{T} = 0.025$, $\rho_{10}^{T} = 0.025$, $\rho_{111} = 0.9946$, $\rho_{001} = 0.0007$, $\rho_{011} = 0.5$, $\rho_{101} = 0.1078$, $\kappa = 1$, $\sigma = 0.23$,$\alpha = 0.1$, $\zeta = 0$. For Burst-LASSO and SBL baseline, we only simulate the case when $M = 64, 128, 256, 512$ due to their long computation time.

and each channel realization consists of $N_c = 3$ random scattering clusters ranging from $-40°$ to $40°$, and each cluster contains $N_b = 10$ sub-paths concentrated in a $15°$ angular spread. The detailed realization could be found in [72]. Another realistic channel model is mm-SSCM proposed in [1] for high frequency bands (28-73 GHz). The mm-SSCM was developed based on 28- and 73-GHz ultrawideband propagation measurements in New York City, and has been shown to faithfully reproduce realistic impulse responses of measured urban channels. We consider 28 GHz outdoor environment. The number of AoD spatial lobes, the number of sub-paths, the central AoD angle of each AoD spatial lobe and the angular spread are generated according to the distribution in [1]. We consider the following baseline algorithms:

- **D-OMP** [34]: This algorithm proposes the differential orthogonal matching pursuit (D-OMP) algorithm to track a dynamic sparse channel by exploiting its temporal correlation.

- **Modified-SP** [36]: This algorithm exploits the prior support and quality information provided by the previous estimated channel to enhance the current CE performance. We

denote $\hat{\Omega}_{t-1}$ and $\hat{\Omega}_{t-2}$ as the estimated channel supports at the $(t-1)$-th and $(t-2)$-th time slots. Then at t-th time slot, $\mathcal{T}_0$, $\bar{s}$ and s_c in [36] are set to be $\hat{\Omega}_{t-1}$, $|\hat{\Omega}_{t-1}|$ and $|\hat{\Omega}_{t-1} \cap \hat{\Omega}_{t-2}|$, respectively.

- **OAMP** [23]**:** The OAMP assumes i.i.d. sparse channel prior.
- **Burst-LASSO** [7]: The Burst-LASSO exploits the bursty structure of the channel in the spatial domain.
- **Structured Turbo-CS** [25]: The structured Turbo-CS exploits the clustered structure of channel in spatial domain.
- **DCS-AMP** [66]: The DCS-AMP exploits the temporal correlation of sparse signal sequences.
- **SBL-Off-Grid** [27]: The SBL-Off-Grid obtained the true AoD values of the massive MIMO channels by the SBL based algorithm.
- **Optimally-Tuned Weighted LASSO** [73]: The Optimally-Tuned Weighted LASSO proposed in Chapter 2 exploits the prior support information obtained from the previously estimated channels to enhance the current CE performance.

To apply the proposed algorithm in a realistic channel, we update the statistic parameters $\boldsymbol{\rho} \triangleq \{\rho_{ba}^S, \rho_{ca}^T, \rho_{bca}, \alpha, \zeta, \kappa, \sigma^2\}$ where $a, b, c \in \{0, 1\}$ using the EM framework [66]. Specifically, the statistic parameters $\boldsymbol{\rho}$ are initialized using available prior knowledge[6]. In each iteration of each time slot, they are updated based on the latest estimated marginal posterior distribution of $\mathbf{s}_t$, $\boldsymbol{\theta}_t$ and $\mathbf{x}_t$. The detailed EM update equations for $\boldsymbol{\rho}$ are omitted in this thesis for conciseness, interested readers can refer to [66] for detailed derivations. Similarly, to apply the message-passing-based baselines in practical channel tracking, the channel statistical parameters ($\{\lambda, \zeta, \sigma^2\}$ for OAMP, $\{\rho_{01}^S, \rho_{10}^S, \zeta, \sigma^2\}$ for Structured Turbo-CS, $\{\rho_{01}^T, \rho_{10}^T, \alpha, \zeta, \kappa, \sigma^2\}$ for DCS-AMP are also updated by the EM framework. Both low-frequency and high-frequency MIMO systems will be considered. In the low-frequency massive MIMO system, the BS has $M = 128$ antennas and the SCM will be used to generate

[6]Even though the recovery performance of the algorithm is not sensitive to the initial values of the statistic parameters, their initial values would affect the convergence rate of the proposed algorithm. The more accurate the initial values are, the faster the algorithm would converge. Therefore, we could set $\boldsymbol{\rho}$ based on some prior knowledge to speed up the convergence of the algorithm.

channels[7]. In the high-frequency massive MIMO systems, the BS has $M = 256$ antennas and the mm-SSCM will be used to generate channels. We focus on the simulations for the ULA. For the baselines, we use the fixed DFT basis except for the SBL-Off-Grid baseline, which is using the off-grid basis, and the off-grid parameters are updated using the SBL. For the proposed D-TOAMP algorithm, we verify its performance for both with and without grid refinement. In the following simulation results, D-TOAMP (DFT) means the proposed D-TOAMP algorithm with fixed DFT basis, i.e., $\boldsymbol{\beta}_t$ in (3.2.4) is set to be $\mathbf{0}$; D-TOAMP (Off-Grid) means the proposed D-TOAMP algorithm with off-grid basis, i.e., $\boldsymbol{\beta}_t$ in (3.2.4) is updated based on (3.5.6). We set $\mathrm{Thr} = 1 - 10^{-2}, \epsilon = 10^{-7}$. The primary performance metric that we used in all of our experiments, which we refer to as the time-averaged normalized MSE (TNMSE), is defined as

$$\mathrm{TNMSE} \triangleq \frac{1}{T}\sum_{t=1}^{T}\frac{\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2}{\|\mathbf{x}_t\|^2}, \tag{3.6.1}$$

where $\hat{\mathbf{x}}_t$ is the estimate of $\mathbf{x}_t$ at t-th time slot.

3.6.1 Impact of SNR

In Fig. 3.12 and Fig. 3.13, we compare the TNMSE performance of different algorithms versus SNR under the SCM and the mm-SSCM, respectively. For each channel model, we also consider the effect of the user velocity on the channel tracking performance. It can be seen that under each user velocity value, the proposed D-TOAMP achieves sufficient performance gain over all the baseline algorithms, under both less sparse SCM and more sparse mm-SSCM. Moreover, the off-grid based D-TOAMP could further improve the channel tracking performance by mitigating the off-grid leakage. This demonstrates that the proposed algorithm can effectively track the realistic dynamic channels in a massive MIMO system by exploiting the 2D dynamic sparsity of channels.

3.6.2 Impact of Pilot Number

In Fig. 3.14 and Fig. 3.15, we compare the TNMSE performance of different algorithms versus the number of pilot sequences P under the SCM and the mm-SSCM, respectively.

[7]We consider wide sense stationary massive MIMO channel models in this thesis. How to extend the proposed compressive CE algorithm to the non-wide sense stationary massive MIMO channels will be left as part of future work.

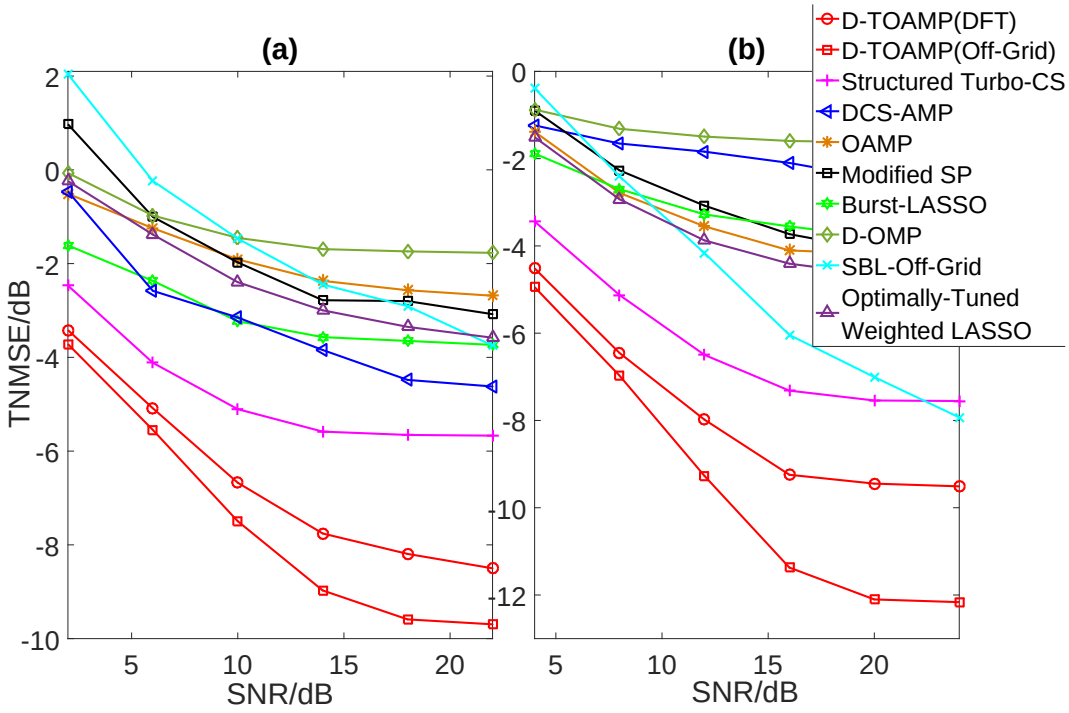

Figure 3.12: TNMSE versus SNR under the SCM. Set $M = 128, P = 26$, and $T = 50$. (a) user velocity is $0.1m/s$; (b) user velocity is $1m/s$.

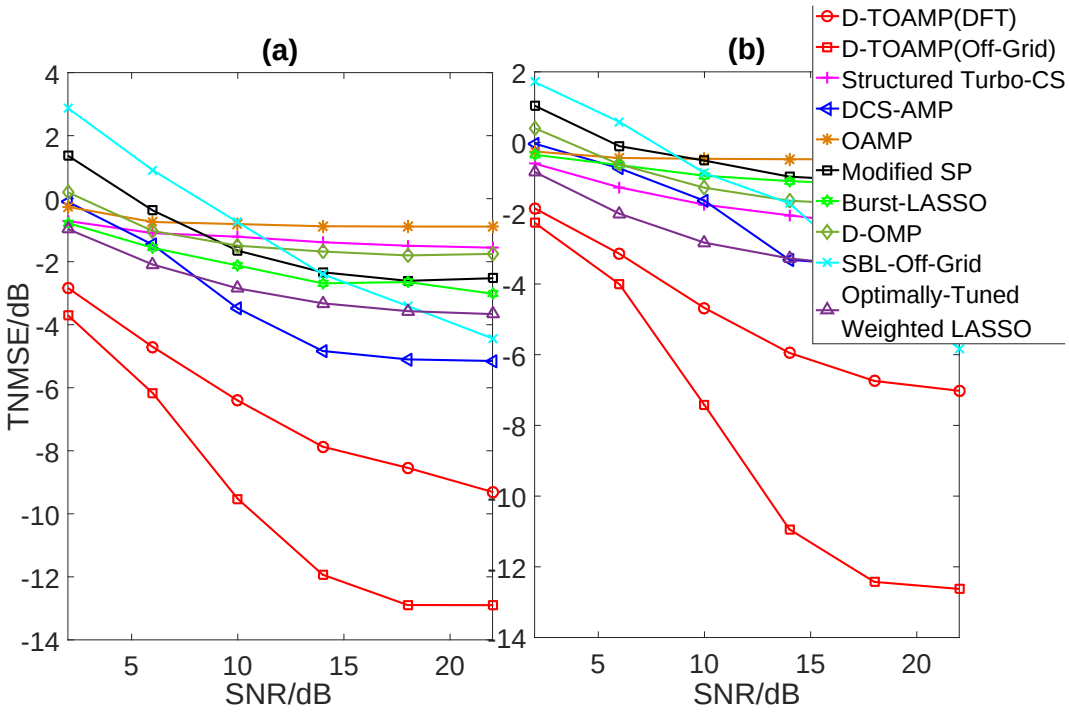

Figure 3.13: TNMSE versus SNR under the mm-SSCM. Set $M = 256, P = 22$, and $T = 50$. (a) user velocity is $0.1m/s$; (b) user velocity is $1m/s$.

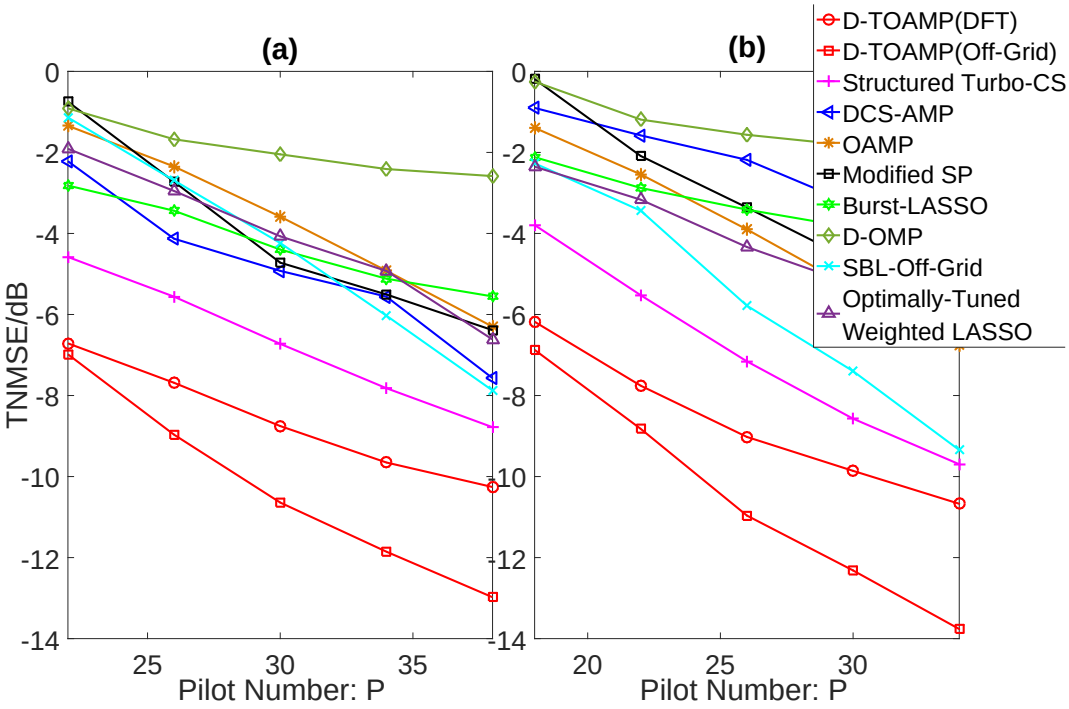

Figure 3.14: TNMSE versus pilot number under the SCM. Set $M = 128$, SNR$= 15$ dB, and $T = 50$. (a) user velocity is $0.1m/s$; (b) user velocity is $1m/s$.

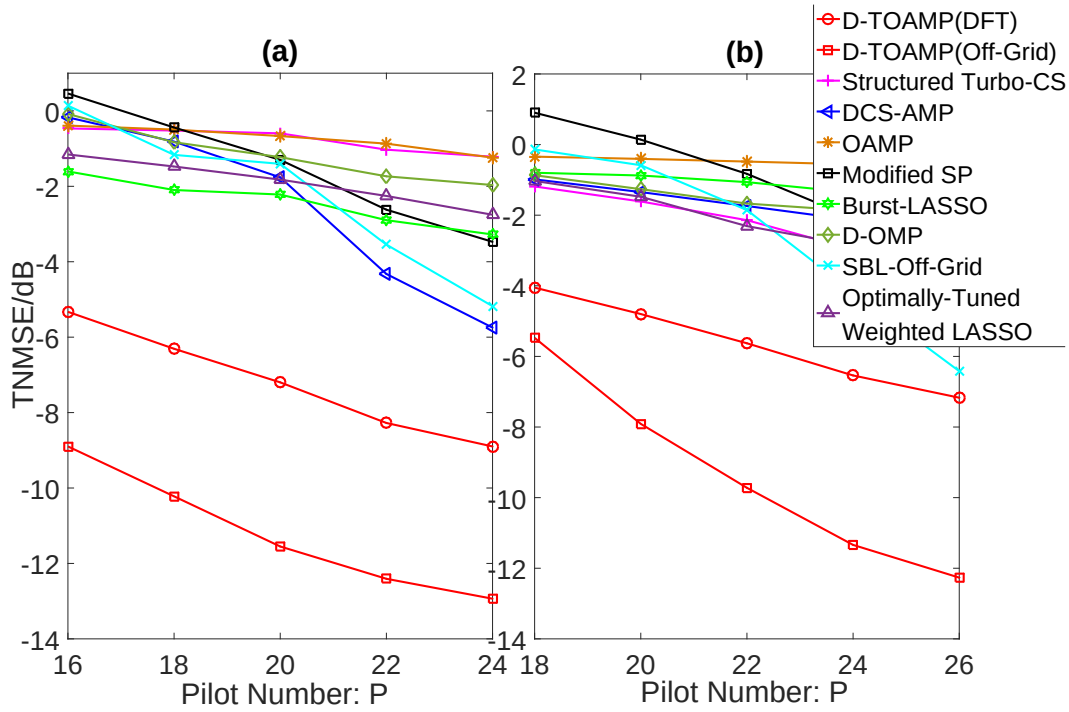

Figure 3.15: TNMSE versus pilot number under the mmWave. Set $M = 256$, SNR$= 15$ dB, and $T = 50$. (a) user velocity is $0.1m/s$; (b) user velocity is $1m/s$.

For each channel model, we also consider the effect of the user velocity on the channel tracking performance. It can be observed that the TNMSE performance decreases as the number of pilots increases for all schemes. The proposed D-TOAMP algorithm can achieve large performance gain over various baselines for different user velocity values and different channel models. Moreover, the off-grid based D-TOAMP could further improve the channel tracking performance. This verifies that the proposed algorithm can accurately recover a time series of realistic channels with low pilot overhead.

3.7 Performance Comparison with Weighted LASSO

From Fig. 3.12 to Fig. 3.15, it can be seen that the proposed D-TOAMP algorithm achieves better channel tracking performance than that of the weighted LASSO algorithm proposed in Chapter 2 under practical massive MIMO channel models. The performance gain comes from the following aspects:

- **The type of measurement matrix**: The optimal weight policy proposed in weighted LASSO algorithm is based on the i.i.d. Gaussian measurement matrix. However, the measurement matrix in D-TOAMP algorithm is set to be partial orthogonal, under which, the weighted LASSO algorithm is no longer optimally tuned. As a result, the performance of weighted LASSO algorithm will deteriorate.

- **The structured sparsity of massive MIMO channels**: The weighted LASSO algorithm can only exploit the prior support information of massive MIMO channel induced by temporal correlation, but cannot exploit its spatial correlation, i.e., clustered structured sparsity in the spatial domain. However, the proposed D-TOAMP algorithm can exploit the 2D dynamic sparsity of massive MIMO channels both in spatial domain and temporal domain to further improve the channel tracking performance.

- **Off-grid mismatch tuning in D-TOAMP algorithm**: The weighted LASSO algorithm doesn't consider the angular grid mismatch effect when representing the channel in angular domain, which can jeopardize the CS recovery performance. However, the D-TOAMP adaptively tunes the off-grid parameters to mitigate the grid mismatch, which can significantly improve the compressive CE performance especially when SNR is high.

Considering the computational complexity of these two algorithms, because the weighted LASSO algorithm has the same computational complexity as the Burst LASSO algorithm, from Fig. 3.11, it can be seen that the AMP based algorithm, e.g., D-TOAMP has much lower computational complexity than the optimization based algorithm, e.g., weighted LASSO or Burst LASSO.

3.8 Summary

We consider the downlink channel tracking problem for a massive MIMO system. Firstly, we propose a statistical channel model called the 2D-MM to model the 2D dynamic sparsity of massive MIMO channels. Then we propose a D-TOAMP algorithm that can be used to recursively track sparse massive MIMO channels with 2D-MM prior. At each time slot, the message passing will be performed based on the prior information passed from the previous time slot and current measurements. Then we verify the superior performance of the proposed channel tracking algorithm under two realistic channel models: SCM and mm-SSCM. Extensive simulations show that the proposed off-grid based D-TOAMP algorithm derived from the 2D-MM channel prior can effectively exploit the 2D dynamic sparsity of practical massive MIMO channels to achieve significant gain over various baseline algorithms.

For clarify, we focus on frequency-flat fading channels in this chapter. In frequency-selective fading channels, there is also structured sparsity in the delay/frequency domain [74]. An interesting future work is to propose a proper probability model to jointly capture the structured sparsity in the spatial/delay/frequency domain and the temporal domain. The proposed D-TOAMP framework can be extended to handle more practical frequency-selective fading channel tracking problem, by modifying the MMSE estimator (Module B) according to the new probability model. For multi-user massive MIMO channel tracking problem, another possibility is to further exploit the common support structure of the multi-user channels in spatial domain to enhance the channel tracking performance in multi-user massive MIMO system.

3.9 Appendix

3.9.1 Gradient Update for Off-grid Parameters

After the D-TOAMP-E step, the posterior estimation of $\mathbf{x}_t$ at the i-th iteration is given by $\hat{p}\left(\mathbf{x}_t|\mathbf{y}^{(t)},\boldsymbol{\beta}_t^i\right) = \mathcal{CN}(\mathbf{x}_{B,t}^{post}, v_{B,t}^{post}\boldsymbol{I})$ ($\mathbf{x}_{B,t}^{post}$ and $v_{B,t}^{post}$ are given by (3.5.33) and (3.5.34), respectively). Then the surrogate function $\hat{u}(\boldsymbol{\beta}_t;\boldsymbol{\beta}_t^i)$ in (3.5.7) can be calculated as

$$\begin{aligned}
&\hat{u}(\boldsymbol{\beta}_t;\boldsymbol{\beta}_t^i)\\
\propto&\mathrm{E}_{\hat{p}\left(\mathbf{x}_t|\mathbf{y}^{(t)},\boldsymbol{\beta}_t^i\right)}\left[-\frac{1}{\sigma_e^2}\left\|\mathbf{y}_t-\mathbf{F}_t\left(\boldsymbol{\beta}_t\right)\mathbf{x}_t\right\|^2\right]\\
\propto&-\frac{1}{\sigma_e^2}\left(\left\|\mathbf{y}_t-\mathbf{F}_t\left(\boldsymbol{\beta}_t\right)\mathbf{x}_{B,t}^{post}\right\|^2+v_{B,t}^{post}\mathrm{tr}\left(\mathbf{F}_t\left(\boldsymbol{\beta}_t\right)\mathbf{F}_t\left(\boldsymbol{\beta}_t\right)^H\right)\right).
\end{aligned}$$

The derivative of $\hat{u}(\boldsymbol{\beta}_t;\boldsymbol{\beta}_t^i)$ w.r.t. $\boldsymbol{\beta}_t$ can be calculated as $\boldsymbol{\xi}_{\boldsymbol{\beta}_t}^{(i)} = \left[\xi^{(i)}\left(\beta_{t,1}\right),\cdots,\xi^{(i)}\left(\beta_{t,M}\right)\right]^T$, with

$$\begin{aligned}
&\xi^{(i)}\left(\beta_{t,m}\right)\\
=&2\mathrm{Re}\left(\mathbf{a}'\left(\hat{\vartheta}_m+\beta_{t,m}\right)^H\mathbf{U}_t\mathbf{U}_t^H\mathbf{a}\left(\hat{\vartheta}_m+\beta_{t,m}\right)\right)c_1^{(i)}+2\mathrm{Re}\left(\mathbf{a}'\left(\hat{\vartheta}_m+\beta_{t,m}\right)^H\mathbf{U}_t\mathbf{c}_2^{(i)}\right),
\end{aligned}\tag{3.9.1}$$

where $c_1^{(i)} = -\frac{1}{\sigma_e^2}\left(\left|x_{B,t,m}^{post}\right|^2+v_{B,t}^{post}\right)$, $\mathbf{c}_2^{(i)} = \frac{1}{\sigma_e^2}\left(x_{B,t,m}^{post}\right)^\star\mathbf{y}_{t,-m}$,

$$\mathbf{y}_{t,-m} = \mathbf{y}_t - \mathbf{U}_t^H\sum_{j\neq m}\mathbf{a}\left(\hat{\vartheta}_j+\beta_{t,j}\right)x_{B,t,j}^{post}$$

and

$$\mathbf{a}'\left(\hat{\vartheta}_m+\beta_{t,m}\right) = d\mathbf{a}\left(\hat{\vartheta}_m+\beta_{t,m}\right)/d\beta_{t,m}.$$

Then the off-grid parameter $\boldsymbol{\beta}_t$ is updated in the derivative direction, i.e., $\boldsymbol{\beta}_t^{i+1} = \boldsymbol{\beta}_t^i+\Delta^i\cdot\boldsymbol{\xi}_{\boldsymbol{\beta}_t^i}^{(i)}$.

Chapter 4

Turbo-VBI for Robust Recovery of Structured Sparse Signals with Uncertain Measurement Matrix

4.1 Introduction

Consider the following linear measurement model:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}, \tag{4.1.1}$$

where $\boldsymbol{x} = [x_1, \cdots, x_N]^T \in \mathbb{C}^N$ is the high dimensional sparse signal need recovered, $\boldsymbol{y} \in \mathbb{C}^M$ is the measurements with $M \ll N$, $\boldsymbol{A} \in \mathbb{C}^{M\times N}$ is the measurement matrix[1], $\boldsymbol{w} = [w_1, ..., w_M]^T \in \mathbb{C}^M$ is the AWGN noise vector with independent Gaussian entries $w_m \sim \mathcal{CN}(w_m; 0, \kappa_m^{-1})$, κ_m is the precision (inverse of the variance) of w_m. In the standard CS model, the measurement matrix $\boldsymbol{A}$ is assumed to be perfectly known and $\boldsymbol{x}$ is assumed to be a simple i.i.d. sparse signal. However, in many practical applications, the measurement matrix $\boldsymbol{A}(\boldsymbol{\theta})$ may contain some uncertain parameters $\boldsymbol{\theta} \in \mathbb{R}^K$. Moreover, in specific applications, the sparse signal $\boldsymbol{x}$ usually has structured sparsity that cannot be modeled easily by i.i.d. priors. Even though the AMP-based algorithms, such as Turbo-AMP [24] and Turbo-CS [25] can exploit the sophisticated priors, they perform badly under a general measurement matrix, especially when the measurement matrix is ill-conditioned. The performance of SBL/VBI is

[1] We use $\mathbf{A}$ as the notation for measurement matrix in this chapter.

insensitive to the measurement matrix. However, the two-layer hierarchical prior in SBL/VBI can not handle more complicated sparse priors.

In this chapter, we propose a novel Turbo-VBI framework to overcome the drawbacks of the existing methods and achieve robust recovery of structured sparse signals with more general uncertain measurement matrix. The proposed Turbo-VBI framework can exploit sophisticated structured sparsity to improve the recovery performance. It is robust w.r.t. the uncertain parameters in the measurement matrix and prior distribution and it works well for more general measurement matrices with possibly correlated columns. The main contributions of this chapter are summarized below.

- **Three-layer hierarchical probability model for structured sparsity:** The choice of sparse probability model is paramount to robust and accurate recovery of structured sparse signals. A good sparse probability model should satisfy the following criteria: it is flexible to capture different structured sparsities in various applications, it is robust w.r.t. the imperfect prior information, it is tractable to enable low-complexity algorithm design. We propose a three-layer hierarchical structured (3LHS) sparse prior model to meet these criteria. Specifically, there are sufficient freedom in the model which can be used to fit the specific structure of sparse signals in different applications as well as incorporate the uncertainty of imperfect prior information.

- **Turbo-VBI algorithm design:** There still lacks efficient algorithms for solving CS problems with 3LHS sparse prior and potentially ill-conditioned measurement matrix. By combining the message passing and VBI approaches via the turbo framework, we propose a Turbo-VBI algorithm which is able to fully exploit the structured sparsity (as captured by the 3LHS sparse prior) under an uncertain (and possibly correlated) measurement matrix to achieve significant gain over the-state-of-art CS recovery algorithms.

The rest of this chapter is organized as follows. In Section 4.2, we present the 3LHS sparsity model. In Section 4.3, we formulate the CS recovery problem under 3LHS sparse prior and uncertain measurement matrix. In Section 4.4, we present the Turbo-VBI framework. Finally, the summaries are given in Section 4.6.

4.2 Three-Layer Hierarchical Structured Sparsity Model

4.2.1 Motivation of 3LHS Structured Sparsity

The probability model for structured sparsity provides the foundation for exploiting the specific sparse structures in different applications. There are two major existing probability models for structured sparsity, as elaborated below.

4.2.1.1 Support-based Probability Model

In the AMP-based algorithms, a support-based probability model is used to capture the structured sparsity [24], where a hidden binary vector $\boldsymbol{s}$ is introduced to indicate the support of the sparse signal $\boldsymbol{s} = [s_1, ..., s_N]^T$. In particular, $s_n = 1$ indicates that the signal coefficient x_n is active (non-zero), while $s_n = 0$ indicates that x_n is inactive (zero). Given the support vector $\boldsymbol{s}$, $\boldsymbol{x}$ is assumed to have independent but non-identically distributed entries, i.e., $p(\boldsymbol{x}|\boldsymbol{s}) = \prod_{n=1}^{N} p(x_n|s_n)$, where

$$p(x_n|s_n) = s_n g_n(x_n) + (1 - s_n)\,\delta(x_n), \tag{4.2.1}$$

where $g_n(x_n)$ denotes the PDF of x_n conditioned on $s_n = 1$, which is often chosen as a Gaussian distribution. The structured sparsity is captured by the prior distribution $p(\boldsymbol{s})$ of the support vector. By choosing a proper $p(\boldsymbol{s})$, the support-based probability model has the flexibility to cover a wide range of structured sparsities, such as Markov sparsity [25] and Markov tree sparsity [26]. However, it is difficult to handle the binary vector $\boldsymbol{s}$ using the optimization-based algorithms and thus AMP-based algorithms are usually used to recover sparse signals $\boldsymbol{x}$ with the support-based probability model, which limits its application since AMP-based algorithms only work well for certain types of measurement matrix (e.g., i.i.d. or partial orthogonal sensing matrices).

4.2.1.2 Two-layer Hierarchical Probability Model

In SBL/VBI, a two-layer hierarchical prior is used to promote i.i.d. or group sparsity [28, 75, 76], where a precision vector $\boldsymbol{\rho} = [\rho_1, ..., \rho_N]^T$ (i.e., $1/\rho_n$ denotes the variance of x_n) is introduced to indicate whether the n-th element x_n is active ($\rho_n = \Theta(1)$) or inactive ($\rho_n \gg 1$). Given the precision vector $\boldsymbol{\rho}$, $\boldsymbol{x}$ is assumed to have independent but non-identically

distributed Gaussian entries, i.e., $p(\boldsymbol{x}|\boldsymbol{\rho}) = \prod_{n=1}^{N} p(x_n|\rho_n)$, where

$$p(x_n|\rho_n) = \mathcal{CN}\left(x_n; 0, \rho_n^{-1}\right), \forall n. \tag{4.2.2}$$

and $\rho_n, \forall n$ are modeled as independent Gamma distributions, i.e., $p(\boldsymbol{\rho}) = \prod_{n=1}^{N} p(\rho_n)$ with

$$p(\rho_n) = \Gamma(\rho_n; a_n, b_n), \tag{4.2.3}$$

where a_n, b_n are set to be a small number to promote sparsity of $\boldsymbol{x}$ [77]. The performance of SBL/VBI with such two-layer hierarchical prior is insensitive to the measurement matrix. It is also possible to model the group sparsity by assigning the same precision ρ_i to the i-th group of elements in $\boldsymbol{x}$. However, it is not flexible enough to model more complicated sparse structures, such as the Markov (tree) priors or Hidden Markov priors considered in [25, 31], because the precision vector $\boldsymbol{\rho}$ is fixed to be (independent) Gamma distributions to enable tractable/low-complexity algorithm design based on SBL/VBI. Moreover, in many practical applications, it is possible to obtain some statistical prior support information (PSI) which indicates the probability of each element being active (i.e., $\Pr(s_n = 1), \forall n$) [73]. However, it is difficult to incorporate such statistical PSI into the two-layer hierarchical prior.

In the following, we shall introduce a 3LHS sparse model to capture the more complicated structured sparsity that may occur in practice, by combining the advantages of the support-based probability model and two-layer hierarchical prior.

4.2.2 Probability Model for the 3LHS Structured Sparsity

Without loss of generality, suppose the index set $\{1, ..., N\}$ of $\boldsymbol{x}$ can be partitioned into Q non-overlapping subsets $\mathcal{I}_1, ..., \mathcal{I}_Q$ such that all the elements of $\boldsymbol{x}[\mathcal{I}_i], \forall i \in \{1, ..., Q\}$ are simultaneously active or inactive. Correspondingly, we introduce a support vector $\boldsymbol{s} = [s_1, ..., s_Q]^T \in \{0, 1\}^Q$ to indicate whether the i-th subvector $\boldsymbol{x}[\mathcal{I}_i]$ is active ($s_i = 1$) or inactive ($s_i = 0$). Specifically, let $\boldsymbol{\rho} = [\rho_1, ..., \rho_N]^T$ denote the precision vector of $\boldsymbol{x}$ (i.e., $1/\rho_n$ denotes the variance of x_n). When $s_i = 0$, the distribution of the associated precision parameters $\rho_n, \forall n \in \mathcal{I}_i$ is chosen to satisfy $\mathbb{E}[\rho_n] \gg 1, \forall n \in \mathcal{I}_i$ such that the expected variance of $x_n, \forall n \in \mathcal{I}_i$ is close to zero (inactive). Moreover, to improve the robustness w.r.t. the imperfect prior knowledge, we assume that the prior distribution $p(\boldsymbol{s}|\boldsymbol{\phi})$ of the support vector depends on some uncertain parameter $\boldsymbol{\phi}$ with a known prior distribution $p(\boldsymbol{\phi})$. Then

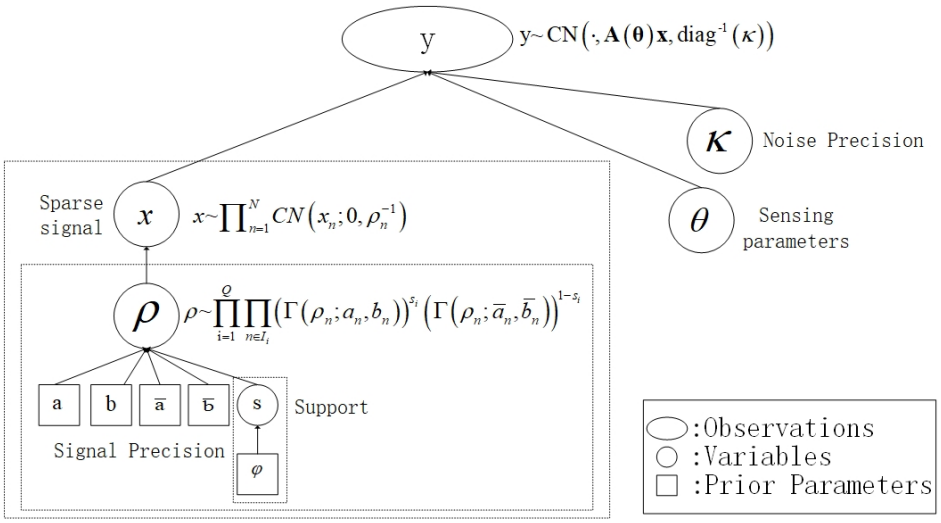

Figure 4.1: Three-layer hierarchical structured sparse prior model.

for given uncertain prior parameter $\boldsymbol{\phi}$, the 3LHS sparse prior distribution (joint distribution of $\boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{s}$) is given by

$$p(\boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{s}|\boldsymbol{\phi}) = \underbrace{p(\boldsymbol{s}|\boldsymbol{\phi})}_{\text{Support}} \underbrace{p(\boldsymbol{\rho}|\boldsymbol{s})}_{\text{Precision}} \underbrace{p(\boldsymbol{x}|\boldsymbol{\rho})}_{\text{Sparse signal}} . \tag{4.2.4}$$

The 3LHS sparse model is illustrated in Fig. 4.1 and the details are elaborated below.

4.2.2.1 Probability Model for the Support Vector s (Layer 1)

The prior distribution $p(\boldsymbol{s}|\boldsymbol{\phi})$ of the support vector is used to capture the structured sparsity in specific applications. In practice, $p(\boldsymbol{s}|\boldsymbol{\phi})$ is chosen based on the nature of the problem. For example, in [25], $p(\boldsymbol{s}|\boldsymbol{\phi})$ is chosen to be a Markov chain to model the clustered scattering environment in massive MIMO channel, where $\boldsymbol{\phi}$ denotes the (possibly unknown) transition probabilities in the Markov chain. In [31], $p(\boldsymbol{s}|\boldsymbol{\phi})$ is chosen to be a hidden Markov model to model the structured sparsity in multi-user massive MIMO channels, where $\boldsymbol{\phi}$ denotes the transition probabilities and other parameters involved in the prior model. The uncertain parameter $\boldsymbol{\phi}$ can be automatically learned from the observations, as will be detailed later.

4.2.2.2 Probability Model for the Precision Vector $\boldsymbol{\rho}$ (Layer 2)

The conditional probability $p(\boldsymbol{\rho}|\boldsymbol{s})$ for the precision vector is given by

$$p(\boldsymbol{\rho}|\boldsymbol{s}) = \prod_{i=1}^{Q} \prod_{n \in \mathcal{I}_i} \left(\Gamma(\rho_n; a_n, b_n)\right)^{s_i} \left(\Gamma\left(\rho_n; \overline{a}_n, \overline{b}_n\right)\right)^{1-s_i}, \tag{4.2.5}$$

where $\Gamma(\rho; a, b)$ is a Gamma hyperprior with shape parameter a and rate parameter b. When $s_i = 1$, $\boldsymbol{x}[\mathcal{I}_i]$ is active. In this case, the shape and rate parameters a_n, b_n of its precision $\rho_n, \forall n \in \mathcal{I}_i$ should be chosen such that $\frac{a_n}{b_n} = \mathbb{E}[\rho_n] = \Theta(1)$ since the variance $1/\rho_n$ of $x_n, \forall n \in \mathcal{I}_i$ is $\Theta(1)$ when it is active. On the other hand, when $s_i = 0$, $\boldsymbol{x}[\mathcal{I}_i]$ is inactive. In this case, the shape and rate parameters $\overline{a}_n, \overline{b}_n$ of its precision $\rho_n, \forall n \in \mathcal{I}_i$ should be chosen to satisfy $\frac{\overline{a}_n}{\overline{b}_n} = \mathbb{E}[\rho_n] \gg 1$ such that the inactive coefficient $x_n, \forall n \in \mathcal{I}_i$ is close to zero. The motivation of considering Gamma hyperprior for $p(\boldsymbol{\rho}|\boldsymbol{s})$ is twofold. First, it is conjugate to Gaussian, hence the associated Bayesian inference can be performed in closed form as will be detailed later. Moreover, as explained above, the conditional probability $p(\boldsymbol{\rho}|\boldsymbol{s})$ can be used to capture the sparsity structure by controlling the mean of the precisions (i.e., the shape parameter a and rate parameter b of the Gamma hyperprior) based on the support vector $\boldsymbol{s}$.

4.2.2.3 Probability Model for the Sparse Signal $\boldsymbol{x}$ (Layer 3)

The conditional probability $p(\boldsymbol{x}|\boldsymbol{\rho})$ for the sparse signal is assumed to have a product form $p(\boldsymbol{x}|\boldsymbol{\rho}) = \prod_{n=1}^{N} p(x_n|\rho_n)$ and each $p(x_n|\rho_n)$ is modeled as a complex Gaussian prior distribution

$$p(x_n|\rho_n) = \mathcal{CN}\left(x_n; 0, \rho_n^{-1}\right), \forall n = 1, ..., N. \tag{4.2.6}$$

The motivation of considering complex Gaussian distribution for $p(x_n|\rho_n)$ is twofold. First, random signals in the nature tend to have a Gaussian distribution due to the central limit theorem. Second, assuming conditional Gaussian prior distribution facilitates low-complexity VBI algorithm design with closed-form update equations [28]. It is well known that the performance of CS recovery algorithms is usually not sensitive to the true distribution of the sparse signal $\boldsymbol{x}$ [28, 58], as long as the proposed probability model can capture the first-order sparse structure of $\boldsymbol{x}$.

The proposed 3LHS sparse model can enjoy the benefits of both the support-based probability model and two-layer hierarchical prior. On one hand, the flexibility of the support-based probability model is preserved as we can choose a proper $p(\boldsymbol{s}|\boldsymbol{\phi})$ to model different structured sparsities in various applications. For example, in [26], $p(\boldsymbol{s}|\boldsymbol{\phi})$ is chosen to be a Markov tree prior to model the wavelet structure in image processing, where $\boldsymbol{\phi}$ denotes the statistical parameters (e.g., the transition probabilities) in the Markov tree prior, and it can be automatically learned using, e.g., the EM-like method. Such complicated sparse structure, however, cannot be modeled by the two-layer hierarchical prior in (4.2.2) and (4.2.3). On the other hand, the 3LHS sparse model also facilitates the design of a Turbo-VBI algorithm, in which the observation model $\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x} + \boldsymbol{w}$ with a general measurement matrix is handled using the VBI approach, while the structured sparsity captured by the prior $p(\boldsymbol{s})$ is handled using the message passing approach. Note that there is no need to specify the layer 2 and 3 distributions $p(\boldsymbol{\rho}|\boldsymbol{s})$ and $p(\boldsymbol{x}|\boldsymbol{\rho})$ for each application since they are fixed as in (4.2.5) and (4.2.6) for all applications.

4.3 CS Problem Formulation with 3LHS Sparse Prior

Recall the CS model with an uncertain measurement matrix

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x} + \boldsymbol{w}. \tag{4.3.1}$$

Let $p(\boldsymbol{\theta})$, $p(\boldsymbol{\phi})$ and $p(\boldsymbol{\kappa})$ denote the known (or assumed) prior distributions of the uncertain parameter in measurement matrix, uncertain parameter in structured support model and noise precision $\boldsymbol{\kappa} = [\kappa_1, ..., \kappa_M]^T$ respectively. Our primary goal is to estimate the sparse signal $\boldsymbol{x}$, its support $\boldsymbol{s}$, and the uncertain parameters $\boldsymbol{\xi} = [\boldsymbol{\theta}; \boldsymbol{\phi}; \boldsymbol{\kappa}]$, given the observations $\boldsymbol{y}$ in model (4.3.1). In particular, for given $\boldsymbol{\xi}$, we are interested in computing the conditional marginal posteriors $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\xi})$ and $p(s_i|\boldsymbol{y}, \boldsymbol{\xi}), \forall i$ (i.e., perform Bayesian inference for $\boldsymbol{x}$ and $s_i, \forall i$), where

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\xi}) &\propto \sum_{\boldsymbol{s}} \int p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{s}|\boldsymbol{\xi})\, d\boldsymbol{\rho} \\ &= \sum_{\boldsymbol{s}} \int p(\boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{s}|\boldsymbol{\phi})\, p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\kappa})\, d\boldsymbol{\rho}, \end{aligned} \tag{4.3.2}$$

$$p\left(s_i|\boldsymbol{y},\boldsymbol{\xi}\right) \;\propto\; \sum_{\boldsymbol{s}_{-i}} \iint p\left(\boldsymbol{x},\boldsymbol{\rho},\boldsymbol{s}|\boldsymbol{\phi}\right) p\left(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\kappa}\right) d\boldsymbol{\rho} d\boldsymbol{x}, \tag{4.3.3}$$

where $p\left(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\kappa}\right) = \mathcal{CN}\left(\boldsymbol{y};\boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x},\mathrm{diag}^{-1}\left(\boldsymbol{\kappa}\right)\right)$. We use $\propto$ to denote equality after scaling, $\boldsymbol{s}_{-i}$ to denote $\left\{s_{i'},\forall i' \neq i\right\}$. On the other hand, the uncertain parameters $\boldsymbol{\xi}$ are obtained by MAP estimation as follows

$$\begin{aligned}\boldsymbol{\xi}^* &= \underset{\boldsymbol{\xi}}{\mathrm{argmax}} \ln p(\boldsymbol{\xi}|\boldsymbol{y}) \\ &= \underset{\boldsymbol{\xi}}{\mathrm{argmax}} \ln \sum_{\boldsymbol{s}} \iint p\left(\boldsymbol{y},\boldsymbol{v},\boldsymbol{\xi}\right) d\boldsymbol{\rho} d\boldsymbol{x},\end{aligned} \tag{4.3.4}$$

where $\boldsymbol{v} = \{\boldsymbol{x},\boldsymbol{\rho},\boldsymbol{s}\}$ is the collection of variables. Once we obtain the MAP estimate of $\boldsymbol{\xi}$, i.e., $\boldsymbol{\xi}^*$, and the associated conditional marginal posteriors $p\left(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\xi}^*\right)$, $p\left(s_i|\boldsymbol{y},\boldsymbol{\xi}^*\right),\forall i$, we can obtain the MAP estimation of $\boldsymbol{x}$ and s_i (conditioned on $\boldsymbol{\xi}=\boldsymbol{\xi}^*$) as $\boldsymbol{x}^* = \mathrm{argmax}_{\boldsymbol{x}} p\left(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\xi}^*\right)$ and $\boldsymbol{s}_i^* = \mathrm{argmax}_{s_i} p\left(s_i|\boldsymbol{y},\boldsymbol{\xi}^*\right)$.

It is very challenging to calculate the exact posterior in (4.3.2) because the factor graph of the underlying model in (4.3.2) has loops. In the next section, we shall propose a Turbo-VBI algorithm which approximately calculates the marginal posteriors $p\left(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\xi}\right)$ and $p\left(s_i|\boldsymbol{y},\boldsymbol{\xi}\right),\forall i$ by combining the message passing and VBI approaches via the turbo framework, and use an inexact block majorization-minimization (MM) method (which is a generalization of the EM method) [27] to find an approximate solution for (4.3.4).

The above CS problem formulation embraces many applications. In the next chapter, we will apply the proposed Turbo-VBI algorithm to user location tracking problem in massive MIMO systems to show its superior performance.

4.4 Turbo-VBI Algorithm

The basic idea of Turbo-VBI is to simultaneously approximate the intractable posterior $p\left(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi}\right)$ with a tractable variational distribution $q\left(\boldsymbol{v};\boldsymbol{\xi}\right)$ and maximize the marginal posterior $\ln p(\boldsymbol{y},\boldsymbol{\xi})$ with respect to the uncertain parameter $\boldsymbol{\xi}$ as in (4.3.4). In summary, the Turbo-VBI algorithm performs iterations between the following two major steps until convergence.

- **Turbo-VBI-E Step:** For given $\boldsymbol{\xi}$, evaluate $q\left(\boldsymbol{v};\boldsymbol{\xi}\right)$ to approximate the posterior $p\left(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi}\right)$ by combining the message passing and VBI approaches via the turbo framework, as will be elaborated in Section 4.4.3 and 4.4.4;

- **Turbo-VBI-M Step:** Given $q\left(\boldsymbol{v};\boldsymbol{\xi}\right) \approx p\left(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi}\right)$, construct a surrogate function (lower bound) for the objective function $\ln p(\boldsymbol{y},\boldsymbol{\xi})$, properly partition $\boldsymbol{\xi}$ into B blocks $\boldsymbol{\xi} = (\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_B)$, and then alternatively maximize the surrogate function with respect to each $\boldsymbol{\xi}_j$, as will be elaborated in Section 4.4.1.

In the following, we first elaborate the Turbo-VBI-M step, which is an extension of the inexact block majorization-minimization (MM) method in [27]. Then we show how to construct the surrogate function based on the EM method, which requires the calculation of the posterior $p\left(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi}\right)$. Finally, we elaborate how to approximately calculate the posterior $p\left(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi}\right)$ in the Turbo-VBI-E Step.

4.4.1 Turbo-VBI-M Step (Inexact Block MM)

It is difficult to directly maximize $\ln p(\boldsymbol{y},\boldsymbol{\xi})$ because there is no closed-form expression due to the multi-dimensional integration over $\boldsymbol{v}$. To make the problem tractable, in the Turbo-VBI-M Step, we first properly partitions $\boldsymbol{\xi}$ into B blocks $\boldsymbol{\xi} = (\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_B)$, such that the resultant subproblem w.r.t. each block can be solved efficiently (e.g., has a closed-form or low complexity solution). In many cases, $\boldsymbol{\xi}$ consists of several subsets of parameters $\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_B$, where each subset $\boldsymbol{\xi}_j$ has a distinct physical meaning. In this case, $\boldsymbol{\xi}$ can be naturally partitioned into blocks according to the physical meaning of each block. Then we alternatively maximizes a surrogate function of $\ln p(\boldsymbol{y},\boldsymbol{\xi})$ with respect to each $\boldsymbol{\xi}_j$, $j \in \{1, \cdots, B\}$. The surrogate function is chosen such that the alternating maximization w.r.t. each variable $\boldsymbol{\xi}_j$ has a closed-form/simple solution. Specifically, let $u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ be the surrogate function constructed at some fixed point $\dot{\boldsymbol{\xi}}$, which satisfies the following properties:

$$u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right) \leq \ln p(\boldsymbol{y},\dot{\boldsymbol{\xi}}),\ \forall \boldsymbol{\xi}, \tag{4.4.1}$$

$$u\left(\dot{\boldsymbol{\xi}};\dot{\boldsymbol{\xi}}\right) = \ln p(\boldsymbol{y},\dot{\boldsymbol{\xi}}), \tag{4.4.2}$$

$$\frac{\partial u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)}{\partial \boldsymbol{\xi}}|_{\xi=\dot{\xi}} = \frac{\partial \ln p\left(\mathbf{y},\boldsymbol{\xi}\right)}{\partial \boldsymbol{\xi}}|_{\xi=\dot{\xi}}. \tag{4.4.3}$$

Then in the Turbo-VBI-M Step of the i-th iteration, we update $\boldsymbol{\xi}_j$ alternatively for $j = 1, ..., B$ as

$$\boldsymbol{\xi}_j^{(i+1)} = \underset{\boldsymbol{\xi}_j}{\operatorname{argmax}}\, u\left(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right), \tag{4.4.4}$$

where $(\cdot)^{(i)}$ stands for the i-th iteration, $\boldsymbol{\xi}_{-j}^{(i)} = \left(\boldsymbol{\xi}_1^{(i+1)}, ..., \boldsymbol{\xi}_{j-1}^{(i+1)}, \boldsymbol{\xi}_{j+1}^{(i)}, ..., \boldsymbol{\xi}_B^{(i)}\right)$. The update rule in (4.4.4) guarantees the convergence of the algorithm to a stationary point of (4.3.4) [27]. The initial value of $\boldsymbol{\xi}$ is set according to the specific application scenario based on the available prior knowledge of $\boldsymbol{\xi}$. If it is difficult to find the global optimal solution of (4.4.4) for some $j \in \overline{\mathcal{J}} \subseteq \{1, ..., B\}$ (e.g., when $u\left(\boldsymbol{\xi}; \dot{\boldsymbol{\xi}}\right), \forall j \in \overline{\mathcal{J}}$ is non-convex w.r.t. $\boldsymbol{\xi}_j$), we can partition the index set $\{1, ..., B\}$ into to two subsets $\overline{\mathcal{J}}$ and $\mathcal{J} = \{1, ..., B\} \setminus \overline{\mathcal{J}}$ such that for $j \in \mathcal{J}$, $u\left(\boldsymbol{\xi}; \dot{\boldsymbol{\xi}}\right)$ is strongly convex w.r.t. $\boldsymbol{\xi}_j$, while for $j \in \overline{\mathcal{J}}$, we do the following gradient update:

$$\boldsymbol{\xi}_j^{(i+1)} = \boldsymbol{\xi}_j^{(i)} + \gamma^{(i)} \frac{\partial u\left(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right)}{\partial \boldsymbol{\xi}_j} |_{\boldsymbol{\xi}_j = \boldsymbol{\xi}_j^{(i)}}, \tag{4.4.5}$$

where $\gamma^{(i)}$ is the step size determined by the Armijo rule [78].

In the original inexact block MM method for massive MIMO channel estimation in [27], there is only one non-convex block (i.e., $\left|\overline{\mathcal{J}}\right| = 1$), and the solution of maximizing the surrogate function over each convex block in (4.4.4) for all $j \in \mathcal{J}$ is unique. The convergence proof in [27] also relies on this fact. However, in the more general problem considered in this chapter, it is possible that there are multiple non-convex blocks (i.e., $\left|\overline{\mathcal{J}}\right| > 1$). As a result, the convergence proof in [27] can no longer be applied to our problem. To address this challenge, we impose an additional condition that $u\left(\boldsymbol{\xi}; \dot{\boldsymbol{\xi}}\right)$ must be strongly convex w.r.t. $\boldsymbol{\xi}_j, \forall j \in \mathcal{J}$, and obtain the following convergence theorem for the above Turbo-VBI algorithm. Please refer to Appendix 4.7.1 for the detailed proof.

Theorem 4.1 (Convergence of Inexact MM)**.** *Suppose the surrogate function $u\left(\boldsymbol{\xi}; \dot{\boldsymbol{\xi}}\right)$ satisfies (4.4.1) - (4.4.3) and it is strongly convex w.r.t. $\boldsymbol{\xi}_j, \forall j \in \mathcal{J}$. If at each iteration, we do the exact update as in (4.4.4) for $j \in \mathcal{J}$, and inexact (gradient) update as in (4.4.5) for $j \in \overline{\mathcal{J}}$, the iterates generated by the Turbo-VBI algorithm converge to a stationary point of Problem (4.3.4).*

Curious readers may wonder how the Turbo-VBI-E step plays a role in the convergence proof. It turns out that in order to construct a surrogate function $u\left(\boldsymbol{\xi}; \dot{\boldsymbol{\xi}}\right)$ that satisfies the conditions in (4.4.1) - (4.4.3) based on the EM method, we have to obtain the posterior $p\left(\boldsymbol{v}|\boldsymbol{y}, \boldsymbol{\xi}\right)$ using the Turbo-VBI-E step. Therefore, the Turbo-VBI-E step is implicitly required in the construction of surrogate function, as explained in the next subsection.

4.4.2 EM-based Surrogate Function

Inspired by the EM method [30], we use the following surrogate function:

$$u\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)=u^{\text{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)+\sum_{j\in\mathcal{J}^1}\tau_j\left\|\boldsymbol{\xi}_j-\dot{\boldsymbol{\xi}}_j\right\|^2, \tag{4.4.6}$$

where $u^{\text{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)=\int p\left(\boldsymbol{v}|\boldsymbol{y},\dot{\boldsymbol{\xi}}\right)\ln\frac{p(\boldsymbol{v},\boldsymbol{y},\boldsymbol{\xi})}{p\left(\boldsymbol{v}|\boldsymbol{y},\dot{\boldsymbol{\xi}}\right)}d\boldsymbol{v}$ is the EM surrogate function used in [27], $\mathcal{J}^1\subseteq\{1,...,B\}$ is the index set such that $u^{\text{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ is convex but not strongly convex w.r.t. $\boldsymbol{\xi}_j,\forall j\in\mathcal{J}^1$, and $\tau_j>0$ can be any constant. The second term is added to ensure that (4.4.6) is strongly convex w.r.t. $\boldsymbol{\xi}_j,\forall j\in\mathcal{J}$, where $\mathcal{J}$ is the index set such that $u^{\text{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ is convex w.r.t. $\boldsymbol{\xi}_j,\forall j\in\mathcal{J}$. It can be shown that the surrogate function in (4.4.6) satisfies (4.4.1) - (4.4.3). Therefore, if we can calculate the exact posterior $p\left(\boldsymbol{v}|\boldsymbol{y},\dot{\boldsymbol{\xi}}\right)$ for given $\dot{\boldsymbol{\xi}}$, we can construct the surrogate function in (4.4.6) and the corresponding Turbo-VBI algorithm converges to a stationary point of (4.3.4). Unfortunately, in many cases, the exact posterior $p\left(\boldsymbol{v}|\boldsymbol{y},\dot{\boldsymbol{\xi}}\right)$ is intractable. Thus, we propose to combine the message passing and VBI approaches via the turbo framework to find an alternative probability density function $q\left(\boldsymbol{v};\dot{\boldsymbol{\xi}}\right)$ to approximate the posterior $p\left(\boldsymbol{v}|\boldsymbol{y},\dot{\boldsymbol{\xi}}\right)$ for any given $\dot{\boldsymbol{\xi}}$ in the Turbo-VBI-E Step, which is expected to be close to the true posterior [28]. Then we construct a tractable surrogate function as

$$\hat{u}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)=\hat{u}^{\text{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)+\sum_{j\in\hat{\mathcal{J}}^1}\tau_j\left\|\boldsymbol{\xi}_j-\dot{\boldsymbol{\xi}}_j\right\|^2, \tag{4.4.7}$$

where

$$\hat{u}^{\text{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)=\int q\left(\boldsymbol{v};\dot{\boldsymbol{\xi}}\right)\ln\frac{p\left(\boldsymbol{v},\boldsymbol{y},\boldsymbol{\xi}\right)}{q\left(\boldsymbol{v};\dot{\boldsymbol{\xi}}\right)}d\boldsymbol{v}$$

is an approximation of $u^{\text{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$, $\hat{\mathcal{J}}^1$ is defined based on the convexity of $\hat{u}^{\text{EM}}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ w.r.t. each block similar to $\mathcal{J}^1$. Since the posterior approximation $q\left(\boldsymbol{v};\dot{\boldsymbol{\xi}}\right)$ obtained in the Turbo-VBI-E Step is usually accurate enough [28], $\hat{u}\left(\boldsymbol{\xi};\dot{\boldsymbol{\xi}}\right)$ is expected to approximately satisfy (4.4.1) - (4.4.3), and thus such approximation has little effect on the convergence of the proposed algorithm, as verified in the simulations. Therefore, after the convergence of the Turbo-VBI with the tractable surrogate function in (4.4.7), we not only obtain an approximate stationary solution $\hat{\boldsymbol{\xi}}$ of (4.3.4), but also the associated (approximate) conditional marginal posteriors $p\left(\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{\xi}}\right)\approx q\left(\boldsymbol{x};\hat{\boldsymbol{\xi}}\right)$ and $p\left(s_i|\boldsymbol{y},\hat{\boldsymbol{\xi}}\right)\approx q\left(s_i;\hat{\boldsymbol{\xi}}\right),\forall i$.

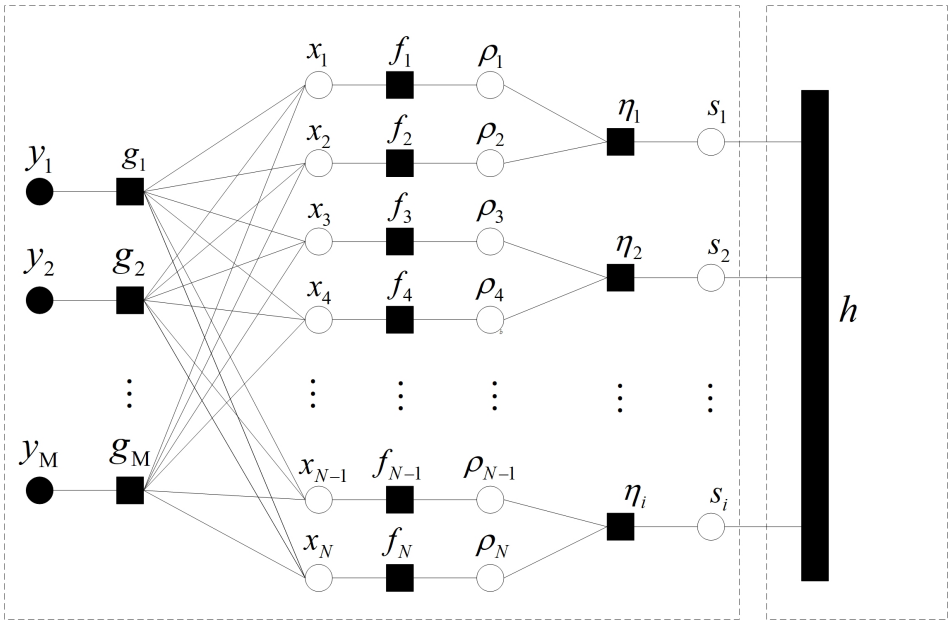

Figure 4.2: Factor graph of the joint distribution in (4.3.2). For easy illustration, we assume every two adjacent elements of the sparse signal $\boldsymbol{x}$ form a group, i.e., $Q = N/2$ and $\mathcal{I}_i = \{2i-1, 2i\}$.

Factor	Distribution	Functional form
$g_m(y_m, \boldsymbol{x})$	$p\left(y_m \vert \boldsymbol{x}, \boldsymbol{\xi}\right)$	$\mathcal{CN}(y_m; \boldsymbol{A}_m\left(\boldsymbol{\theta}\right)\boldsymbol{x}, \kappa_m^{-1})$
$f_n\left(x_n, \rho_n\right)$	$p\left(x_n \vert \rho_n\right)$	$\mathcal{CN}\left(x_n; 0, \rho_n^{-1}\right)$
$\eta_i\left(\boldsymbol{\rho}\left[\mathcal{I}_i\right], s_i\right)$	$p\left(\boldsymbol{\rho}\left[\mathcal{I}_i\right] \vert s_i\right)$	$\prod_{n\in\mathcal{I}_i}\left(\Gamma\left(\rho_n; a_n, b_n\right)\right)^{s_i}\left(\Gamma\left(\rho_n; \overline{a}_n, \overline{b}_n\right)\right)^{1-s_i}$
$h\left(\boldsymbol{s}\right)$	$p\left(\boldsymbol{s}\right)$	depends on application

Table 4.1: Factors, distributions and functional forms in Fig. 4.2. $\boldsymbol{A}_m\left(\boldsymbol{\theta}\right)$ denotes the m-th row of $\boldsymbol{A}\left(\boldsymbol{\theta}\right)$.

4.4.3 Modules of the Turbo-VBI-E Step

The factor graph of the joint distribution $p\left(\boldsymbol{y}, \boldsymbol{v} \vert \boldsymbol{\xi}\right)$, denoted by $\mathcal{G}$, is illustrated in Fig. 4.2, where the function expression of each factor node is listed in Table 4.1. Since $\mathcal{G}$ is a dense graph with many loops, directly applying the sum-product message passing (SPMP) [79] over the entire factor graph $\mathcal{G}$ usually cannot achieve a good performance. When the measurement matrix is i.i.d. Gaussian or partial orthogonal, Turbo-AMP [26] and Turbo-CS [25] algorithms can be used to achieve approximate message passing over dense graphs. However, in our problem, the measurement matrix $\boldsymbol{A}(\boldsymbol{\theta})$ can be ill conditioned and the performance of Turbo-AMP or Turbo-CS is very poor, as verified by the numerical simulations. To overcome this challenge, we combine the VBI [28] and message passing approaches [79] via the turbo framework to design the Turbo-VBI-E step that can achieve approximate message passing over $\mathcal{G}$ with a good performance.

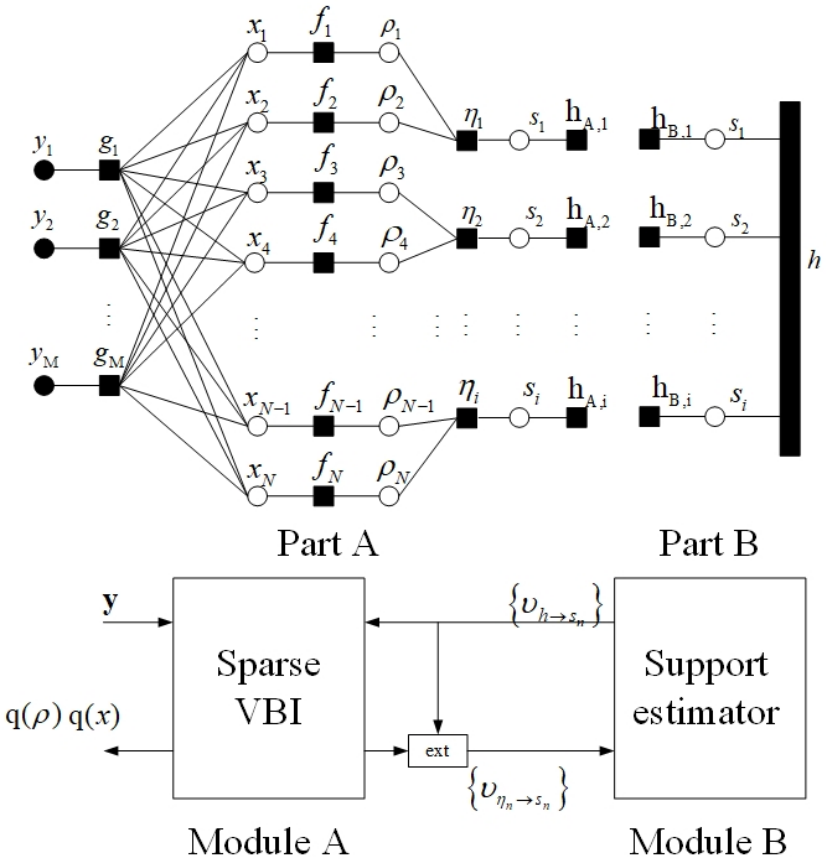

Figure 4.3: Modules of the Turbo-VBI algorithm and message flows between different modules.

Specifically, we follow the turbo framework and partition the factor graph $\mathcal{G}$ into two parts, as shown in Fig. 4.3, where Part A contains the dense subgraph $\mathcal{G}_x$ (as well as a copy of the support vector $\boldsymbol{s}$ and an additional set of factor nodes $h_{A,i}, i = 1, ..., Q$), and Part B contains subgraph $\mathcal{G}_s$ (and an additional set of factor nodes $h_{B,i}, i = 1, ..., Q$). Correspondingly, the Turbo-VBI-E step has two modules to perform Bayesian inference (message passing) over Part A and Part B, respectively. Moreover, Module A and Module B also need to exchange messages between them, as shown in Fig. 4.3. In particular, the messages $\{\nu_{\eta_i \to s_i}(\cdot)\}$ form the outputs of Module A and the inputs of Module B, while the messages $\{\nu_{h \to s_i}(\cdot)\}$ form the outputs of Module B and the inputs of Module A. The two modules are executed iteratively until convergence. In the following, we first outline Module A and Module B. The details of Module A are presented in Subsection 4.4.4.

Based on the observation $\boldsymbol{y}$ and messages $\{\nu_{h \to s_i}(\cdot)\}$ from Module B, Module A performs the sparse VBI [28] to calculate the approximate conditional marginal posteriors $q(\boldsymbol{x}; \boldsymbol{\xi}) \approx p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\xi})$; $q(\boldsymbol{\rho}; \boldsymbol{\xi}) \approx p(\boldsymbol{\rho}|\boldsymbol{y}, \boldsymbol{\xi})$; and $q(s_i; \boldsymbol{\xi}) \approx p(s_i|\boldsymbol{y}, \boldsymbol{\xi}), \forall i$. To be more specific, in Part

A, the factor nodes

$$h_{A,i}\left(s_i\right) \triangleq \nu_{h\to s_i}\left(s_i\right), i = 1, ..., Q,$$

incorporate the prior information from Module B. Therefore, the following prior distribution is assumed when performing the sparse VBI in Module A:

$$\hat{p}\left(\boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{s}\right) = \hat{p}\left(\boldsymbol{s}\right) p\left(\boldsymbol{\rho}|\boldsymbol{s}\right) p\left(\boldsymbol{x}|\boldsymbol{\rho}\right), \hat{p}\left(\boldsymbol{s}\right) = \prod_i \left(\pi_i\right)^{s_i} \left(1-\pi_i\right)^{1-s_i}, \tag{4.4.8}$$

where

$$\pi_i \triangleq \hat{p}\left(s_i = 1\right) = \frac{\nu_{h\to s_i}\left(1\right)}{\nu_{h\to s_i}\left(1\right) + \nu_{h\to s_i}\left(0\right)}.$$

Note that the only difference between the prior in (4.4.8) and the original prior in (4.2.4) is that the prior distribution $p\left(\boldsymbol{s}|\boldsymbol{\phi}\right)$ of the support vector is replaced with a prior $\hat{p}\left(\boldsymbol{s}\right)$ with independent entries. The corresponding posterior distribution of $\boldsymbol{x}$ obtained by the sparse VBI is complex Gaussian, as will be given in (4.4.14), and the posterior distribution $q\left(s_i\right)$ of s_i will be given by (4.4.20). After that, the messages $\{\nu_{\eta_i\to s_i}\left(s_i\right)\}$ from Module A to Module B can be calculated from the posterior distribution $q\left(s_i\right)$ by subtracting the input message $\{\nu_{h\to s_i}\left(s_i\right)\}$ as [22]

$$\nu_{\eta_i\to s_i}\left(s_i\right) = \frac{q\left(s_i\right)}{\nu_{s_i\to\eta_i}\left(s_i\right)}, \tag{4.4.9}$$

where the denominator of (4.4.9) equals to $\nu_{h\to s_i}(s_i)$ according to the sum product rule.

Based on the messages $\{\nu_{\eta_i\to s_i}\left(\cdot\right)\}$ from Module A, Module B further exploits the structured sparsity as captured by the prior distribution $p\left(\boldsymbol{s}\right)$ of the support vector to improve the estimation performance, by performing the SPMP algorithm over the support subgraph $\mathcal{G}_s$ in Part B. To be more specific, in Part B, the factor nodes

$$h_{B,i}\left(s_i\right) \triangleq \nu_{\eta_i\to s_i}\left(s_i\right), i = 1, ..., Q$$

incorporate the prior information from Module A, and the factor node h incorporates the structured sparsity. After that, the messages $\{\nu_{h\to s_i}\left(\cdot\right)\}$ from Module B to Module A can be calculated according to the sum-product rule.

4.4.4 Sparse VBI Estimator (Module A)

4.4.4.1 Outline of Sparse VBI

For convenience, we use $\boldsymbol{v}^k$ to denote an individual variable in $\boldsymbol{v}$, such as $\boldsymbol{x}$, $\boldsymbol{\rho}$ and s_i. Let $\mathcal{H} = \{k|\forall \boldsymbol{v}^k \in \boldsymbol{v}\}$. Based on the VBI method, the approximate conditional marginal posterior could be calculated by minimizing the Kullback-Leibler divergence (KLD) [28] between $\hat{p}\left(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi}\right)$ and $q\left(\boldsymbol{v};\boldsymbol{\xi}\right)$ subject to a factorized form constraint on $q\left(\boldsymbol{v};\boldsymbol{\xi}\right)$ as

$$\mathscr{A}_{\mathrm{VBI}}:\ q^{\star}\left(\boldsymbol{v};\boldsymbol{\xi}\right) = \arg\min_{q(\boldsymbol{v};\boldsymbol{\xi})} \int q\left(\boldsymbol{v};\boldsymbol{\xi}\right) \ln \frac{q\left(\boldsymbol{v};\boldsymbol{\xi}\right)}{\hat{p}\left(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi}\right)} d\boldsymbol{v} \tag{4.4.10}$$

$$\text{s.t.}\ \ q\left(\boldsymbol{v};\boldsymbol{\xi}\right) = \prod_{k\in\mathcal{H}} q\left(\boldsymbol{v}^k;\boldsymbol{\xi}\right), \tag{4.4.11}$$

$$\int q\left(\boldsymbol{v}^k;\boldsymbol{\xi}\right) d\boldsymbol{v}^k = 1, \forall k \in \mathcal{H} \tag{4.4.12}$$

where $\hat{p}\left(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi}\right)$ is the posterior distribution of $\boldsymbol{v}$ with the prior $\hat{p}\left(\boldsymbol{x},\boldsymbol{\rho},\boldsymbol{s}\right)$ in (4.4.8), and for discrete variable $\boldsymbol{s}$, $\int(\cdot)d\boldsymbol{v}^k$ means the summation over the set of all possible discrete values of $\boldsymbol{s}$. In this section, $\boldsymbol{\xi}$ is fixed and we will omit the argument $\boldsymbol{\xi}$ in $q\left(\boldsymbol{v};\boldsymbol{\xi}\right)$ for simplicity. Solving $\mathscr{A}_{\mathrm{VBI}}$ yields a good approximation of the true posterior $\hat{p}\left(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi}\right)$ and such a VBI method has been widely used in Bayesian inference with great success [28]. Since problem $\mathscr{A}_{\mathrm{VBI}}$ is non-convex, the uniqueness of the optimal solution may not be guaranteed. Fortunately, the existence of the optimal solution has been proved in [28]. In the following, we aim at finding a stationary solution (denoted by $q^*\left(\boldsymbol{v}\right)$) of $\mathscr{A}_{\mathrm{VBI}}$, as defined below.

Definition 4.1 (Stationary Solution of $\mathscr{A}_{\mathrm{VBI}}$)**.** $q^*\left(\boldsymbol{v}\right) = \prod_{k\in\mathcal{H}} q^*\left(\boldsymbol{v}^k\right)$ is called a stationary solution of Problem $\mathscr{A}_{\mathrm{VBI}}$ if it satisfies all the constraints in $\mathscr{A}_{\mathrm{VBI}}$ and $\forall k \in \mathcal{H}$,

$$q^*\left(\boldsymbol{v}^k\right) = \arg\min_{q\left(\boldsymbol{v}^k\right)} \int \prod_{l\neq k} q^*\left(\boldsymbol{v}^l\right) q\left(\boldsymbol{v}^k\right) \ln \frac{\prod_{l\neq k} q^*\left(\boldsymbol{v}^l\right) q\left(\boldsymbol{v}^k\right)}{\hat{p}\left(\boldsymbol{v}|\boldsymbol{y},\boldsymbol{\xi}\right)}.$$

By finding a stationary solution $q^*\left(\boldsymbol{v}\right)$ of $\mathscr{A}_{\mathrm{VBI}}$, we could obtain the approximate posterior $q^*\left(\boldsymbol{v}^k\right) \approx p\left(\boldsymbol{v}^k|\boldsymbol{y},\boldsymbol{\xi}\right), \forall k \in \mathcal{H}$.

A stationary solution of $\mathscr{A}_{\mathrm{VBI}}$ can be obtained via alternately optimizing each individual density $q\left(\boldsymbol{v}^k\right), k \in \mathcal{H}$, as will be proved by Lemma 4.1. Specifically, for given $q\left(\boldsymbol{v}^l\right), \forall l \neq k$, the optimal $q\left(\boldsymbol{v}^k\right)$ that minimizes the KLD in $\mathscr{A}_{\mathrm{VBI}}$ is given by [28]

$$q\left(\boldsymbol{v}^k\right) \propto \exp\left(\left\langle \ln \hat{p}\left(\boldsymbol{v},\boldsymbol{y}|\boldsymbol{\xi}\right)\right\rangle_{\prod_{l\neq k} q\left(\boldsymbol{v}^l\right)}\right), \tag{4.4.13}$$

where $\langle f(x)\rangle_{q(x)} = \int f(x)\, q(x) dx$. Based on (4.4.13), the update equations of all variables are given in the subsequent subsections. The detailed derivation can be found in Appendix 4.7.2. Note that the operator $\langle\cdot\rangle_{\boldsymbol{v}^k}$ is equivalent to $\langle\cdot\rangle_{q\left(\boldsymbol{v}^k\right)}$ and the expectation $\left\langle f\left(\boldsymbol{v}^k\right)\right\rangle_{q\left(\boldsymbol{v}^k\right)}$ w.r.t. its own approximate posterior is simplified as $\left\langle f\left(\boldsymbol{v}^k\right)\right\rangle$.

4.4.4.2 Initialization of Sparse VBI

In order to trigger the alternating optimization (AO) algorithm, we use the following initializations for the distribution functions $q(\boldsymbol{s})$ and $q(\boldsymbol{\rho})$.

- In the first outer iteration, initialize $q(\boldsymbol{s}) = \hat{p}(\boldsymbol{s}) = \prod_{i=1}^{Q} \hat{p}(s_i)$ with $\hat{p}(s_i) = (\pi_i)^{s_i} (1-\pi_i)^{1-s_i}$. In the rest outer iterations, initialize $q(\boldsymbol{s}) = \prod_{i=1}^{Q} (\tilde{\pi}_i)^{s_i} (1-\tilde{\pi}_i)^{1-s_i}$, where $\tilde{\pi}_i$ is the (approximate) posterior probability of $s_i = 1$ calculated from Module A (sparse VBI estimator) in the previous outer iteration.

- Initialize a gamma distribution for $\boldsymbol{\rho}$: $q(\boldsymbol{\rho}) = \prod_{n=1}^{N} \Gamma\left(\rho_n; \tilde{a}_n, \tilde{b}_n\right)$, where $\tilde{a}_n = \pi_i a_n + (1-\pi_i)\,\overline{a}_n$, $\tilde{b}_n = \pi_i b_n + (1-\pi_i)\,\overline{b}_n$, $\forall n \in \mathcal{I}_i, \forall i$.

4.4.4.3 Update for $\boldsymbol{x}$

From (4.4.13), the update for $q(\boldsymbol{x})$ only depends on $q(\boldsymbol{\rho})$. For given $q(\boldsymbol{\rho})$, $q(\boldsymbol{x})$ can be derived as

$$q(\boldsymbol{x}) = \mathcal{CN}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{4.4.14}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be calculated through

$$\boldsymbol{\Sigma} = \left(\operatorname{diag}(\langle\boldsymbol{\rho}\rangle) + \boldsymbol{A}(\boldsymbol{\theta})^H \operatorname{diag}(\boldsymbol{\kappa})\, \boldsymbol{A}(\boldsymbol{\theta})\right)^{-1}, \tag{4.4.15}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma} \boldsymbol{A}(\boldsymbol{\theta})^H \operatorname{diag}(\boldsymbol{\kappa})\, \boldsymbol{y}. \tag{4.4.16}$$

4.4.4.4 Update for $\boldsymbol{\rho}$

From (4.4.13), for given $q(\boldsymbol{s})$ and $q(\boldsymbol{x})$, $q(\boldsymbol{\rho})$ can be derived as

$$q(\boldsymbol{\rho}) = \prod_{n=1}^{N} \Gamma\left(\rho_n; \tilde{a}_n, \tilde{b}_n\right), \tag{4.4.17}$$

where the approximate posterior parameters $\tilde{a}_n, \tilde{b}_n$ are given by:

$$\tilde{a}_n = \langle s_i \rangle a_n + \langle 1 - s_i \rangle \overline{a}_n + 1, \tag{4.4.18}$$

$$\tilde{b}_n = \left\langle |x_n|^2 \right\rangle + \langle s_i \rangle b_n + \langle 1 - s_i \rangle \overline{b}_n, \forall n \in \mathcal{I}_i, \forall i. \tag{4.4.19}$$

4.4.4.5 Update for s

From (4.4.13), the update for $q(\boldsymbol{s})$ only depends on $q(\boldsymbol{\rho})$. From (4.4.13), for given $q(\boldsymbol{\rho})$, $q(\boldsymbol{s})$ can be derived as

$$q(\boldsymbol{s}) = \prod_{i=1}^{Q} (\tilde{\pi}_i)^{s_i} (1 - \tilde{\pi}_i)^{1 - s_i}, \tag{4.4.20}$$

where $\tilde{\pi}_i$ is given by

$$\tilde{\pi}_i = \frac{1}{C} \prod_{n \in \mathcal{I}_i} \frac{\pi_i b_n^{a_n}}{\Gamma(a_n)} e^{(a_n - 1)\langle \ln \rho_n \rangle - b_n \langle \rho_n \rangle}, \tag{4.4.21}$$

and C is the normalization constant, given by

$$C = \prod_{n \in \mathcal{I}_i} \frac{\pi_i b_n^{a_n}}{\Gamma(a_n)} e^{(a_n - 1)\langle \ln \rho_n \rangle - b_n \langle \rho_n \rangle} + \prod_{n \in \mathcal{I}_i} \frac{(1 - \pi_i) \overline{b}_n^{\overline{a}_n}}{\Gamma(\overline{a}_n)} e^{(\overline{a}_n - 1)\langle \ln \rho_n \rangle - \bar{b}_n \langle \rho_n \rangle}.$$

The involved expectations are given as follows for $\forall i \in \{1, ..., Q\}, n \in \{1, ..., N\}$:

$$\langle s_i \rangle = \tilde{\pi}_i, \langle 1 - s_i \rangle = 1 - \tilde{\pi}_i, \tag{4.4.22}$$

$$\langle \rho_n \rangle = \frac{\tilde{a}_n}{\tilde{b}_n}, \langle \ln \rho_n \rangle = \psi(\tilde{a}_n) - \ln\left(\tilde{b}_n\right), \tag{4.4.23}$$

$$\langle \boldsymbol{x} \rangle = \boldsymbol{\mu}, \left\langle |x_n|^2 \right\rangle = |\mu_n|^2 + \Sigma_n, \tag{4.4.24}$$

where $\psi(x) = \frac{d}{dx} \ln(\Gamma(x))$ is the digamma function, defined as the logarithmic derivative of the gamma function, μ_n is the n-th element of $\boldsymbol{\mu}$, and Σ_n is the n-th diagonal element of matrix $\boldsymbol{\Sigma}$.

4.4.4.6 Convergence of Sparse VBI

The sparse VBI can be viewed as an AO method [80] to solve $\mathscr{A}_{\text{VBI}}$. It is clear that the sparse VBI can monotonically decreasing the objective value in $\mathscr{A}_{\text{VBI}}$, and thus the objective value will converge to a limit. Moreover, for given $q\left(\boldsymbol{v}^l\right), \forall l \neq k$, the optimal $q\left(\boldsymbol{v}^k\right)$ that minimizes the KLD in $\mathscr{A}_{\text{VBI}}$ is unique. Then according to the convergence of AO in [80], we

Algorithm 4.1 Turbo-VBI algorithm

Input: $\boldsymbol{y}$, prior distributions $p(\boldsymbol{\theta})$, $p(\boldsymbol{\phi})$ and $p(\boldsymbol{\kappa})$, measurement matrix $\mathbf{A}(\boldsymbol{\theta})$, which is a function of unknown variable $\boldsymbol{\theta}$.

Output: $\boldsymbol{\xi}^*$, $\boldsymbol{x}^*$, $\boldsymbol{s}_i^*, \forall i$.

Initialize the uncertain parameters $\boldsymbol{\xi}$, and the message $\pi_i \triangleq \nu_{h \to s_i}(1)$.
while not converge **do**
  Turbo-VBI-E Step:
  %Module A: Sparse VBI Estimator
  Initialize the distribution functions $q(\boldsymbol{s})$ and $q(\boldsymbol{\rho})$.
  while not converge **do**
    Update $q(\boldsymbol{x}; \boldsymbol{\xi})$ using (4.4.14) and the related expectations using (4.4.24).
    Update $q(\boldsymbol{\rho}; \boldsymbol{\xi})$ using (4.4.17) and the related expectations using (4.4.23).
    Update $q(\boldsymbol{s}; \boldsymbol{\xi})$ using (4.4.20) and the related expectations using (4.4.22).
  end while
  Calculate the extrinsic information of s_i based on (4.4.9), send $\nu_{\eta_i \to s_i}(s_i)$ to Module B.
  % Module B:
  Perform the SPMP over the support subgraph $\mathcal{G}_s$, send $\nu_{h \to s_i}(s_i)$ to Module A.
  Turbo-VBI-M Step:
  Construct the surrogate function $\hat{u}$ in (4.4.7) using the approximate posterior output of Module A, i.e., $q(\boldsymbol{v}; \boldsymbol{\xi})$.
  Update $\boldsymbol{\xi}_j$ alternatively for $j = 1, ..., B$ using (4.4.4) for $j \in \mathcal{J}$ and (4.4.5) for $j \in \overline{\mathcal{J}}$.
end while
Output $\boldsymbol{\xi}^*$, $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} q(\boldsymbol{x}; \boldsymbol{\xi}) = \boldsymbol{\mu}$ and $s_i^* = \arg\max_{s_i} q(s_i; \boldsymbol{\xi})$.

have the following convergence theorem for the sparse VBI.

Lemma 4.1 (Convergence of Sparse VBI)**.** *Every limiting point* $q^*(\boldsymbol{v}) = \prod_{k \in \mathcal{H}} q^*\left(\boldsymbol{v}^k\right)$ *generated by the sparse VBI using (4.4.14), (4.4.17) and (4.4.20) with the initialization in Section 4.4.4.2 is a stationary solution of Problem* $\mathscr{A}_{\mathrm{VBI}}$.

Finally, the overall Turbo-VBI algorithm is summarized in Algorithm 4.1.

4.5 Comparison with Weighted LASSO and Turbo-OAMP Algorithm

Compared to the weighted LASSO algorithm in Chapter 2 and Turbo-OAMP based algorithm in Chapter 3, the Turbo-VBI algorithm proposed in this chapter has the following advantages:

- The Turbo-VBI algorithm can be applied to a general measurement matrix, e.g., ill-conditioned measurement matrix or measurement matrix with uncertain parameters. However, the optimal weight policy in weighted LASSO algorithm is proposed for

i.i.d. Gaussian measurement matrix, and Turbo-OAMP algorithm is proposed for partial orthogonal measurement matrix. Under a more general measurement matrix, the performance of weighted LASSO and Turbo-OAMP algorithm will deteriorate or even diverge.

- The Turbo-VBI and Turbo-OAMP algorithm can exploit complicated structured sparsities, e.g., Markov structure or Markov tree structure. However, the LASSO based algorithm can only exploit simple structured sparsity, such as group sparsity or statistical prior support information. For complicated structured sparsities, it's hard to design a proper regularization function and find the optimal regularizer parameter in a LASSO based algorithm.

- The Turbo-VBI and Turbo-OAMP algorithm can automatically learn the model parameters from the data through EM framework. However, the LASSO based algorithms cannot handle uncertain parameters involved in the model.

4.6 Summary

We propose a novel Turbo-VBI framework for robust recovery of structured sparse signals under uncertain (and possibly correlated) measurement matrices. To capture various sophisticated structured sparsities in practice, we propose a new 3LHS sparse prior model which is not only more flexible than existing commonly used sparse models, but also tractable for low complexity algorithm design. To handle the 3LHS sparse prior model and uncertain/correlated measurement matrix, we propose a Turbo-VBI algorithm which approximately calculates the marginal posteriors of the sparse signal by combining the message passing and VBI approaches via the turbo framework, and use an inexact block MM method (which is a generalization of the EM method) to find an approximate MAP estimator for the uncertain parameters in the measurement matrix. We further establish the convergence of the inexact block MM and sparse VBI components in the Turbo-VBI framework. In the next chapter, we will apply the proposed Turbo-VBI framework to solve the location tracking problem in massive MIMO systems.

4.7 Appendix

4.7.1 Proof of Theorem 4.1

Using the property of surrogate function and gradient update, we have

$$\begin{aligned}\ln p(\boldsymbol{y}, \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}) &= u\left(\boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right) \\ &\leq u\left(\boldsymbol{\xi}_j^{(i+1)}, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right) \\ &\leq \ln p(\boldsymbol{y}, \boldsymbol{\xi}_j^{(i+1)}, \boldsymbol{\xi}_{-j}^{(i)}), \forall j \in \overline{\mathcal{J}},\end{aligned}$$

where the equality holds only when the gradient w.r.t. $\boldsymbol{\xi}_j$ is zero, i.e., the gradient update in (4.4.5) always strictly increases the surrogate function and the original objective function whenever $\frac{\partial u\left(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right)}{\partial \boldsymbol{\xi}_j} \neq 0$. It is clear that the update in (4.4.4) obtained by maximizing the surrogate function also always strictly increases the surrogate function and the original objective function whenever $\frac{\partial u\left(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right)}{\partial \boldsymbol{\xi}_j} \neq 0$. Therefore, the objective value will keep increasing until converging to a certain value p^*, and we must have

$$\lim_{i\to\infty} \frac{\partial u\left(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right)}{\partial \boldsymbol{\xi}_j} = 0, \forall j, \tag{4.7.1}$$

(otherwise, the objective value will keep increasing to infinity, which contradicts with the fact that $\ln p(\boldsymbol{y}, \boldsymbol{\xi}_j^{(i+1)}, \boldsymbol{\xi}_{-j}^{(i)})$ must be bounded above). Then according to (4.7.1) and the property of gradient update, we must have $\lim_{i\to\infty} \left\|\boldsymbol{\xi}_j^{(i+1)} - \boldsymbol{\xi}_j^{(i)}\right\| = 0, \forall j \in \overline{\mathcal{J}}$. Moreover, it follows from (4.7.1) and the strong convexity of $u\left(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{-j}^{(i)}; \boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right)$ w.r.t. $\boldsymbol{\xi}_j, \forall j \in \mathcal{J}$ that $\lim_{i\to\infty} \left\|\boldsymbol{\xi}_j^{(i+1)} - \boldsymbol{\xi}_j^{(i)}\right\| = 0, \forall j \in \mathcal{J}$. Therefore, we have

$$\lim_{i\to\infty} \left\|\boldsymbol{\xi}_j^{(i+1)} - \boldsymbol{\xi}_j^{(i)}\right\| = 0, \forall j. \tag{4.7.2}$$

It follows from (4.7.2) that all the B sequences $\left\{\boldsymbol{\xi}_j^{(i)}, \boldsymbol{\xi}_{-j}^{(i)}\right\}, j = 1, ..., B$ have the same set of limiting points. Let $\left\{\boldsymbol{\xi}_j^{(i_t)}, \boldsymbol{\xi}_{-j}^{(i_t)}, t = 1, 2, ...\right\}$ denote a subsequence that converges to a limiting point $\boldsymbol{\xi}^*$. Suppose $\boldsymbol{\xi}^*$ is not a stationary point of $\ln p(\boldsymbol{y}, \boldsymbol{\xi})$, then $\frac{\partial \ln p(\boldsymbol{y}, \boldsymbol{\xi}^*)}{\partial \boldsymbol{\xi}} \neq 0$ and it follows from (4.7.2) that $\lim_{t\to\infty} \frac{\partial u\left(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{-j}^{(i_t)}; \boldsymbol{\xi}_j^{(i_t)}, \boldsymbol{\xi}_{-j}^{(i_t)}\right)}{\partial \boldsymbol{\xi}_j} \neq 0$ must hold at least for some j, which contradicts with (4.7.1). Therefore, every limiting point $\boldsymbol{\xi}^*$ must be a stationary point of $\ln p(\boldsymbol{y}, \boldsymbol{\xi})$.

4.7.2 Derivation of (4.4.14)-(4.4.21)

Based on (4.4.13), $q(\boldsymbol{x})$ in (4.4.14) can be obtained as

$$\begin{aligned}
&\ln q(\boldsymbol{x}) \\
\propto &\left\langle \ln p\left(\boldsymbol{x}|\boldsymbol{\rho}\right)\right\rangle_{\boldsymbol{\rho}} + \ln p\left(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\xi}\right) \\
\propto &-\boldsymbol{x}^H \mathrm{diag}\left(\left\langle\boldsymbol{\rho}\right\rangle\right)\boldsymbol{x} - \left\|\mathrm{diag}^{1/2}\left(\boldsymbol{\kappa}\right)\left(\boldsymbol{y}-\boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}\right)\right\|^2 \\
\propto &-\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^H \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right).
\end{aligned}$$

$q(\boldsymbol{\rho})$ in (4.4.17) can be obtained as

$$\begin{aligned}
&\ln q(\boldsymbol{\rho}) \\
\propto &\left\langle \ln p\left(\boldsymbol{x}|\boldsymbol{\rho}\right)\right\rangle_{\boldsymbol{x}} + \left\langle \ln p\left(\boldsymbol{\rho}|\boldsymbol{s}\right)\right\rangle_{\boldsymbol{s}} \\
\propto &\sum_{i=1}^{Q}\sum_{n\in\mathcal{I}_i}\left(\left\langle s_i\right\rangle a_n + \left\langle 1-s_i\right\rangle \overline{a}_n\right)\ln\rho_n - \left(\left\langle\left|x_n\right|^2\right\rangle + \left\langle s_i\right\rangle b_n + \left\langle 1-s_i\right\rangle \overline{b}_n\right)\rho_n \\
\propto &\sum_{i=1}^{Q}\sum_{n\in\mathcal{I}_i}\left(\tilde{a}_n - 1\right)\ln\rho_n - \tilde{b}_n\rho_n.
\end{aligned}$$

$q(\boldsymbol{s})$ in (4.4.20) can be obtained as

$$\begin{aligned}
&\ln q(\boldsymbol{s}) \\
\propto &\left\langle \ln p\left(\boldsymbol{s}|\boldsymbol{\rho}\right)\right\rangle_{\boldsymbol{\rho}} + \ln\hat{p}\left(\boldsymbol{s}\right) \\
\propto &\sum_{i=1}^{Q}\sum_{n\in\mathcal{I}_i} s_i\left(\ln b_n^{a_n} + \left(a_n-1\right)\left\langle\ln\rho_n\right\rangle - b_n\left\langle\rho_n\right\rangle - \ln\Gamma(a_n)\right) \\
&+ \left(1-s_i\right)\left(\ln \overline{b}_n^{\overline{a}_n} + \left(\overline{a}_n-1\right)\left\langle\ln\rho_n\right\rangle - \overline{b}_n\left\langle\rho_n\right\rangle - \ln\Gamma(\overline{a}_n)\right) + \sum_{i=1}^{Q}\left(s_i\ln\pi_i + \left(1-s_i\right)\ln\left(1-\pi_i\right)\right) \\
\propto &\sum_{i=1}^{Q}\sum_{n\in\mathcal{I}_i}\left(s_i\ln\frac{\pi_i b_n^{a_n}}{\Gamma(a_n)}e^{(a_n-1)\langle\ln\rho_n\rangle - b_n\langle\rho_n\rangle} + \left(1-s_i\right)\ln\frac{\left(1-\pi_i\right)\overline{b}_n^{\overline{a}_n}}{\Gamma(\overline{a}_n)}e^{(\overline{a}_n-1)\langle\ln\rho_n\rangle - \overline{b}_n\langle\rho_n\rangle}\right) \\
\propto &\ln\prod_{i=1}^{Q}\left(\tilde{\pi}_i\right)^{s_i}\left(1-\tilde{\pi}_i\right)^{1-s_i}.
\end{aligned}$$

Chapter 5

D-VBI for User Location Tracking in Massive MIMO Systems

In this chapter, we apply the Turbo-VBI framework proposed in Chapter 4 to user location tracking problem in massive MIMO system, and propose a variant of Turbo-VBI framework, i.e., D-VBI to tackle the challenges arising in this specific application.

5.1 Introduction

Location-based services (LBSs) are gaining increasing popularity these days resulting from the increasing importance of ubiquitous computing and context-dependent information, and the advances in localization-based technologies [81–83]. The crucial point when realizing this class of service is continuous position tracking at regular intervals to monitor the spatial objects, detect the relationship between a user and his or her surroundings and proactively perform actions [83]. To fulfill the needs of tracking, current mobile devices are equipped with several positioning methods that are based on the Global Positioning System (GPS), WiFi or cell-identity (CID). However, GPS positioning is not suitable for indoor and non-line-of-sight (NLOS) positioning, and has huge power consumption on mobile devices [82]. Additionally, positioning performances of WiFi and CID are poor [82, 83].

On the other hand, massive MIMO, which operates with a large number of antennas at the base station (BS), is a promising technology to meet the capacity demand in 5G wireless networks due to its increased spectral efficiency, high directivity and low complexity [32]. In addition to the communication benefits, the massive MIMO technique could also be exploited

to enable high-accuracy localization [38, 39]. In this chapter, we focus on using massive MIMO systems for efficient tracking of user location.

However, massive-MIMO-based localization is far from mature and there are only a few recent works on this topic. [84] and [85] proposed received-signal-strength (RSS)-based positioning by fingerprinting and machine learning methods in a distributed massive MIMO system, respectively. [86] proposed to locate a mobile station (MS) by exploiting changes in the statistics of the sparse beam space channel matrix. However, these methods are based on data training, which requires a huge amount of training data and cannot work well in a dynamic environment with random wireless fading channels. Angle of arrivals (AoA) estimation was used in [39] and [87], and the combined estimation of time-of-arrival (ToA), angle of departure (AoD) and AoA was used in [38] and [88] for positioning users in massive MIMO systems. In [89], a novel hybrid RSS-AoA-based localization approach was proposed for a millimeter-wave massive MIMO system. All of the above works belong to the indirect localization method, i.e., the intermediate parameters such as the AoA, ToA or RSS are first estimated from the received signals, and then the user's position is determined via trilateration or triangulation. However, in dense multipath environments, such as urban or indoor areas, the performance of such an indirect localization method will be degraded due to the inability to correctly detect and/or estimate the intermediate parameters (AoA, ToA, RSS) of the line-of-sight (LOS) path [90].

One alternative way to address the localization problem is the direct localization approach [91]. Instead of calculating through the intermediate parameters, the user's location is estimated directly from the received signals in direct localization. Initially, the direct localization approach was applied to the pure LOS environment [91]. Later on, it was extended to the multi-path environment in [90,92]. Specifically, a direct localization was proposed in [90] for massive MIMO systems, which models the LOS channels as group sparse directly using the user's location, and assumes the NLOS paths have arbitrary AoAs. In a cellular network, direct localization requires the received signals from the BSs to be sent to the cloud radio access network (C-RAN) to cooperatively estimate the location.

These works focus on the static localization problem, which considers the localization at one instant of time. Instead of performing individual localization at each time slot, we could exploit the user mobility and the temporal correlation of wireless channels to improve the location tracking accuracy. In [93], the user is assumed to be moving among neighboring

grid cells, and a Markov model (MM) is used to predict the user's movement. Such a user mobility model induces probabilistic temporal correlation (PTC) of massive MIMO channel support (indices of the non-zero elements). Moreover, in direct localization, the LOS channels between the user and different BSs all originate from the same user position; therefore, the energies of the LOS channels associated with different BSs all concentrate on the same location index, which will induce a group-sparsity (GS) structure of the LOS channels. We call this joint property of massive MIMO channels PTC-GS in this paper. Many works have studied the recovery of dynamic sparse signals. For example, [94] proposed a block SBL framework to exploit the temporal correlation of sources; [95] proposed a fast SBL algorithm to handle the time-varying states. However, the simple row sparse model considered in [94] and the diagonal first-order vector auto-regressive process considered in [95] cannot capture the PTC-GS structure considered in this paper. The message-passing-based algorithms, such as [24, 25, 96], could capture more complicated priors. However, they only work for the i.i.d. Gaussian or partial orthogonal measurement matrix, which is not the case in the localization problem. To the best of our knowledge, how to exploit the PTC-GS to enhance the performance of user location tracking in massive MIMO systems has not been fully studied. In this chapter, we propose a 3LHS sparse model, i.e., temporal Markov group-sparse (TMGS) model to capture the PTC-GS of massive MIMO channels. Based on this, a variant of the Turbo-VBI framework, i.e., dynamic variational Bayesian inference (D-VBI) algorithm is proposed to track the user's location in massive MIMO systems. The main contributions are summarized below.

- **Probability model for PTC-GS in user location tracking:** The direct localization algorithm in [90] is based on a simple group-sparse model that cannot exploit the PTC effect. The conventional user mobility models, such as the hidden Markov in [93], Gauss-Markov in [97] and random walk in [98] are all Markovian, but they cannot capture the GS structure of the channel vectors induced by the cooperative localization. Moreover, their observation models are not suitable for massive MIMO systems. Therefore, we propose a TMGS model to capture the joint GS and PTC structure of the massive MIMO channels in user location tracking problem.

- **Dynamic variational Bayesian inference for TMGS prior with ill-conditioned measurement matrix:** The location tracking is formulated as a sparse Bayesian inference problem, which could be solved by Bayesian-inference-based approaches like VBI [28]

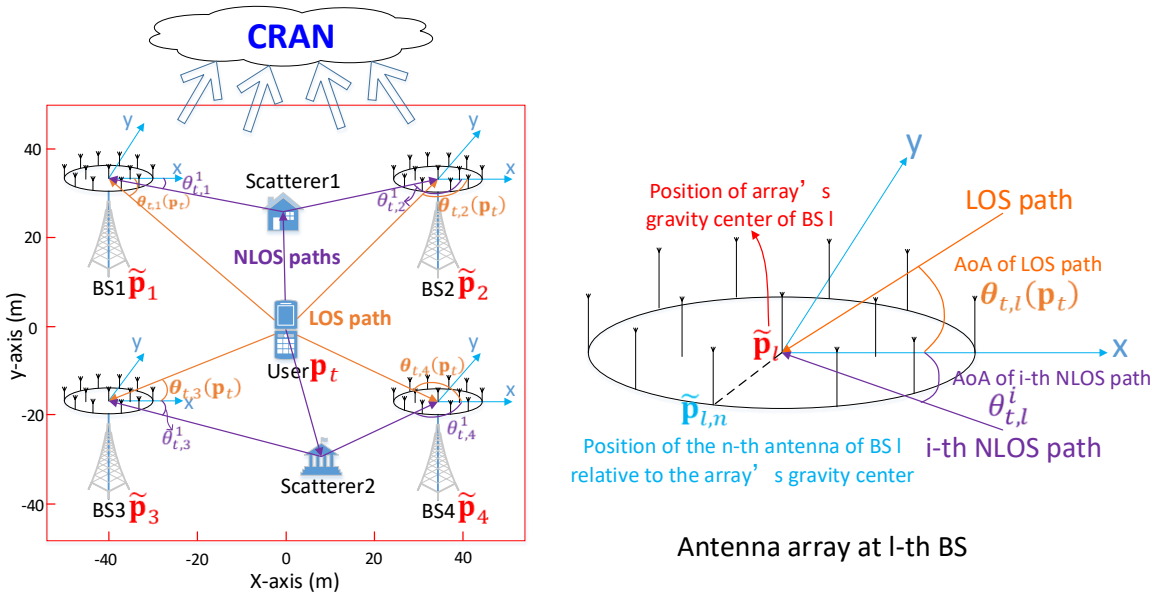

Figure 5.1: Illustration of the localization model in massive MIMO systems.

and SBL [27, 99]. However, the existing Bayesian methods cannot handle the complicated TMGS prior directly. Even though the message-passing-based algorithms, such as Turbo-AMP [24] and Turbo-CS [25], can be adopted to handle complicated priors, they perform poorly under more general measurement matrix. Unfortunately, the measurement matrix in the location tracking problem is ill-conditioned. Therefore, we resort to the Turbo-VBI framework proposed in Chapter 4 to address these challenges and propose a D-VBI algorithm to fully exploit the PTC-GS (as captured by the TMGS prior) and works well for the ill-conditioned measurement matrix.

The rest of the chapter is organized as follows. In Section 5.2, we give the system model for the user location tracking in massive MIMO systems. In Section 5.3, we propose the TMGS probability model to capture the PTC-GS of massive MIMO channels, and formulate the resulting problem. The proposed D-VBI algorithm is presented in Section 5.4. Finally, the simulation results are given in Section 5.5 to verify the advantages of the proposed solution, and the summaries are given in Section 5.6.

5.2 System Model

5.2.1 Localization Model

Consider a 2D geographical area $\mathcal{X}$, which is known as a priori[1]. Within this area, there is a mobile user, whose location is being tracked by L massive MIMO BSs over T time slots based on the uplink signals from the user. We consider a single antenna user, and the BS l has a phased array with N_l antennas. The array responses are assumed to be known by the BSs. At the t-th time slot ($1 \leq t \leq T$), the user is located at $\mathbf{p}_t = [p_t^x, p_t^y]^T$ in $\mathcal{X}$, the center of the gravity of the stations' arrays are located at $\widetilde{\mathbf{p}}_l = [\tilde{p}_l^x, \tilde{p}_l^y]^T$ and assumed to be in the far field with respect to the user, as shown in Fig. 5.1. In this paper, we consider massive MIMO at the BS side and the practical physical finite scattering channel model. For a typical cellular configuration with a tower-mounted BS, there is limited scatterers around the BS. This leads to a small angular spread in the angular domain at the BS side [100–102], implying that only a small fraction of angular bins contain almost all the energy from the multipath signals. As a result, the channel has a sparse or an approximate sparse representation on the virtual angular domain [2, 90]. We consider flat fading channel and narrow band system in this paper, however the proposed algorithm can be extended to frequency-selective fading channel easily[2].

At time slot t, the user broadcasts a signal u_t, which propagates through the multi-path environment resulting in a received signal at BS l given by [2, 90]

$$\mathbf{y}_{t,l} = \mathbf{a}_l\left(\theta_{t,l}\left(\mathbf{p}_t\right)\right)\alpha_{t,l}u_t + \sum_{i=1}^{P_l}\mathbf{a}_l\left(\theta_{t,l}^i\right)\alpha_{t,l}^i u_t + \mathbf{n}_{t,l}, \tag{5.2.1}$$

in which $\mathbf{a}_l(\theta) \in \mathbb{C}^{N_l \times 1}$ is the array response vector at BS l for the AoA θ, $\mathbf{n}_{t,l} \in \mathbb{C}^{N_l \times 1}$ stands for the additive complex Gaussian noise with each element zero mean and $\sigma_{t,l}^2$ variance, $\alpha_{t,l}$ and $\theta_{t,l}$ stand for the complex channel gain and the AoA corresponding to the LOS path, respectively, for BS l, $\alpha_{t,l}^i$ and $\theta_{t,l}^i$ are the channel gain and AoA of the i-th NLOS path,

[1] By saying this, we mean the geographical area where the user may appear is known in advance. However, the propagation environment between user and BSs is unknown, such as the locations of the scatterers, the number of the scatterers, etc.

[2] In frequency-selective fading channel, the PTC-GS structure still holds for the channel at each subcarrier. The proposed algorithm can be directly used to track user's locations employing one subcarrier. On the other hand, the proposed algorithm can be extended to incorporate the ToA information in wideband system by introducing a two-dimensional (2D) grid of both AoAs and ToAs. However, this will increase the computation burden. For simplicity and clarity, we concentrate on a narrowband system in this chapter to achieve a good compromise between the performance and complexity.

respectively, for BS l, and P_l is the number of NLOS paths arriving at BS l. The LOS AoA is related to the user location $\mathbf{p}_t$ through

$$\theta_{t,l}\left(\mathbf{p}_t\right) = \arctan\left(\frac{p_t^y - \tilde{p}_l^y}{p_t^x - \tilde{p}_l^x}\right) + \pi \cdot 1\left(p_t^x < \tilde{p}_l^x\right), \tag{5.2.2}$$

which is computed with respect to the x-axis and is anticlockwise, as shown in Fig. 5.1. $1\left(E\right)$ is one if the logical expression E is true. Let $\widetilde{\mathbf{p}}_{l,n} = [\widetilde{\mathbf{p}}_{l,n}^x, \widetilde{\mathbf{p}}_{l,n}^y]^T$ be the position of the n-th antenna of BS l relative to the array's gravity center. For arrays without mutual antenna coupling and isotropic antennas, the array response vector $\mathbf{a}_l(\theta)$ for the given AoA θ has the following expression:

$$\left[\mathbf{a}_l(\theta)\right]_n = \exp\left(\frac{2\pi i}{\lambda}\widetilde{\mathbf{p}}_{l,n}^T \begin{bmatrix} \cos\left(\theta\right) \\ \sin\left(\theta\right) \end{bmatrix}\right), \tag{5.2.3}$$

where $\left[\mathbf{a}_l(\theta)\right]_n$ denotes the n-th element of $\mathbf{a}_l(\theta)$ and λ denotes the wavelength of the uplink propagation. In practice, for non-ideal arrays with mutual coupling and different antenna gains, $\mathbf{a}_l(\theta)$ is not computed mathematically but can be measured during the array calibration process [90].

5.2.2 Off-Grid Basis for Localization

Directly recovering the user location from (5.2.1) through maximum likelihood (ML) or least square (LS) is difficult because the resulting optimization problem is highly nonconvex and suffers from a lot of local optima. Therefore, we introduce a grid-based model to locate a source. By exploiting the sparsity inherent to the grid-based model, a D-VBI algorithm is developed to obtain the maximum a posteriori (MAP) estimation of the user's location. First, we introduce a uniform grid of Q locations, denoted by $\mathcal{A}$, for user location and a uniform grid of M_l ($M_l \gg P_l$) angles, denoted by Θ_l, for AoAs of NLOS paths at BS l, which are given by

$$\mathcal{A} = \{\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_Q\} \in \mathcal{X}, \tag{5.2.4}$$

and

$$\Theta_l = \{\vartheta_1, \cdots, \vartheta_{M_l}\} \in (0, 2\pi], \forall l \in \{1, \cdots, L\}, \tag{5.2.5}$$

respectively. However, in practice, the true position $\mathbf{p}_t$ and AoAs of NLOS paths $\{\theta_{t,l}^i\}_{i=1}^{P_l}, \forall l$ usually do not lie exactly on the grid point. In this case, there will be mismatches between the true positions/AoAs and the grid points in $\mathcal{A}/\{\Theta_l\}$. To handle this issue, we adopt an off-grid basis for the sparse representation. Specifically, if $\mathbf{p}_t$ is located within the s_t-th grid cell (i.e., the square centered at the s_t-th grid location $\boldsymbol{\phi}_{s_t}$ in $\mathcal{A}$), we write $\mathbf{p}_t$ as

$$\mathbf{p}_t = \boldsymbol{\phi}_{s_t} + \boldsymbol{\kappa}_{t,s_t}, \tag{5.2.6}$$

where $\boldsymbol{\kappa}_{t,s_t} = \left[\kappa_{t,s_t}^x, \kappa_{t,s_t}^y\right]^T$ corresponds to the off-grid gap. Similarly, if $\vartheta_{m_i}, m_i \in \{1, \cdots, M_l\}$ is the nearest grid point to $\theta_{t,l}^i$ in Θ_l, we write $\theta_{t,l}^i$ as

$$\theta_{t,l}^i = \vartheta_{m_i} + \beta_{t,l,m_i}, \tag{5.2.7}$$

where β_{t,l,m_i} corresponds to the off-grid gap.

Note that with the off-grid basis, the model can significantly alleviate the location and direction mismatch because there always exist some $\boldsymbol{\kappa}_{t,s_t}$ and β_{t,l,m_i} making (5.2.6) and (5.2.7) hold exactly. The received signal $\mathbf{y}_{t,l}$ in (5.2.1) can be rewritten using the off-grid basis as

$$\mathbf{y}_{t,l} = \mathbf{A}_{t,l}\left(\boldsymbol{\kappa}_t\right)\mathbf{x}_{t,l} + \mathbf{B}_{t,l}\left(\boldsymbol{\beta}_{t,l}\right)\mathbf{z}_{t,l} + \mathbf{n}_{t,l}, \tag{5.2.8}$$

where $\boldsymbol{\kappa}_t = [\boldsymbol{\kappa}_{t,1}; \cdots; \boldsymbol{\kappa}_{t,Q}]$, $\mathbf{A}_{t,l}\left(\boldsymbol{\kappa}_t\right) = \left[\mathbf{a}_l\left(\theta_{t,l}\left(\boldsymbol{\phi}_1 + \boldsymbol{\kappa}_{t,1}\right)\right), \cdots, \mathbf{a}_l\left(\theta_{t,l}\left(\boldsymbol{\phi}_Q + \boldsymbol{\kappa}_{t,Q}\right)\right)\right]$, and

$$\boldsymbol{\kappa}_{t,q} = \begin{cases} \mathbf{p}_t - \boldsymbol{\phi}_{s_t}, & q = s_t \\ [0,0]^T, & \text{otherwise} \end{cases}, \tag{5.2.9}$$

$\boldsymbol{\beta}_{t,l} = \left[\beta_{t,l,1}, \cdots, \beta_{t,l,M_l}\right]^T$, $\mathbf{B}_{t,l}\left(\boldsymbol{\beta}_{t,l}\right) = \left[\mathbf{a}_l\left(\vartheta_1 + \beta_{t,l,1}\right), \cdots, \mathbf{a}_l\left(\vartheta_{M_l} + \beta_{t,l,M_l}\right)\right]$, and

$$\beta_{t,l,m_i} = \begin{cases} \theta_{t,l}^i - \vartheta_{m_i}, & i = 1, \cdots, P_l \\ 0, & \text{otherwise} \end{cases}. \tag{5.2.10}$$

$\mathbf{x}_{t,l} = \left[x_{t,l,1}, \cdots, x_{t,l,Q}\right]^T \in \mathbb{C}^{Q\times 1}$ is called the sparse LOS channel vector, whose q-th element $x_{t,l,q}$ denotes the complex gain of the LOS path from grid location $\boldsymbol{\phi}_q + \boldsymbol{\kappa}_{t,q}$ to BS l at time slot t. $\mathbf{z}_{t,l} = \left[z_{t,l,1}, \cdots, z_{t,l,M_l}\right]^T \in \mathbb{C}^{M_l\times 1}$ is called the NLOS channel vector, whose m-th element $z_{t,l,m}$ denotes the complex gain of the NLOS path arriving at BS l with angle

$\vartheta_m + \beta_{t,l,m}$ at time slot t. By definition, there is only one nonzero element in $\mathbf{x}_{t,l}, \forall l$, which corresponds to $\alpha_{t,l}u_t$ in (5.2.1). The index of the nonzero element is identical for all $\mathbf{x}_{t,l}, \forall l$, which corresponds to the coarse position of the user. Similarly, there are P_l nonzero elements in $\mathbf{z}_{t,l}$, which correspond to $\alpha_{t,l}^i u_t, i = 1, \cdots, P_l$ in (5.2.1). The non-zero indices of $\mathbf{z}_{t,l}$ correspond to the AoAs of the NLOS paths arriving at BS l.

The aggregate received signal from L BSs at time slot t can be written as

$$\mathbf{y}_t = \mathbf{A}_t(\boldsymbol{\kappa}_t)\mathbf{x}_t + \mathbf{B}_t(\boldsymbol{\beta}_t)\mathbf{z}_t + \mathbf{n}_t, \tag{5.2.11}$$

where $\mathbf{y}_t = [\mathbf{y}_{t,1}; \cdots; \mathbf{y}_{t,L}] \in \mathbb{C}^N$, $N = \sum_{l=1}^L N_l$, $\mathbf{x}_t = [\mathbf{x}_{t,1}; \cdots; \mathbf{x}_{t,L}] \in \mathbb{C}^{LQ}$, $\mathbf{z}_t = [\mathbf{z}_{t,1}; \cdots; \mathbf{z}_{t,L}] \in \mathbb{C}^M$, $M = \sum_{l=1}^L M_l$, $\mathbf{n}_t = [\mathbf{n}_{t,1}; \cdots; \mathbf{n}_{t,L}] \in \mathbb{C}^N$, $\boldsymbol{\beta}_t = [\boldsymbol{\beta}_{t,1}; \cdots; \boldsymbol{\beta}_{t,L}] \in \mathbb{R}^M$, $\mathbf{A}_t(\boldsymbol{\kappa}_t) = \mathrm{Diag}(\mathbf{A}_{t,1}(\boldsymbol{\kappa}_t), \cdots, \mathbf{A}_{t,L}(\boldsymbol{\kappa}_t)) \in \mathbb{C}^{N\times LQ}$ and
$\mathbf{B}_t(\boldsymbol{\beta}_t) = \mathrm{Diag}(\mathbf{B}_{t,1}(\boldsymbol{\beta}_{t,1}), \cdots, \mathbf{B}_{t,L}(\boldsymbol{\beta}_{t,L})) \in \mathbb{C}^{N\times M}$[3].

In the above sparse representation, the coarse position $\boldsymbol{\phi}_{s_t}$ of the user is determined by the index s_t of the nonzero element of $\mathbf{x}_{t,l}, \forall l$. The user's true position $\mathbf{p}_t$ is jointly determined by the coarse position $\boldsymbol{\phi}_{s_t}$ and the corresponding position offset $\boldsymbol{\kappa}_{t,s_t}$. We call $s_t \in \{1, \cdots, Q\}$ the user's *location state*. We denote the support vector of $\mathbf{z}_t$ as $\mathbf{v}_t = [v_{t,1,1}, \cdots, v_{t,L,M_L}]^T \in \{0,1\}^{M\times 1}$, in which, $v_{t,l,m} = 1$, if $z_{t,l,m} \neq 0$; otherwise, $v_{t,l,m} = 0$. In Section 5.4, we will discuss how to jointly recover the user's coarse position and refine the off-grid parameters to learn the user's exact location.

5.2.3 Remarks on the Major Assumptions

For analytical purpose, we have imposed two major assumptions as below.

Assumption 1 (LOS Observability)**.** *We should make sure that the LOS paths between the user and the chosen BSs are not blocked or attenuated during the tracking process, and can be received by the chosen BSs (Or in other words, there exist LOS paths between the user and a few nearby BSs). Chosen BSs means the BSs selected to cooperatively track user's locations. This is usually true in practice for omni antennas and in outdoor transmission. This also provides a BS selection criteria to apply the proposed tracking algorithm, i.e., the BSs which are more likely in the LOS region of the user during tracking process will be chosen to do the cooperative localization.*

[3]In this chapter, we user $\mathbf{A}_t(\boldsymbol{\kappa}_t)$ and $\mathbf{B}_t(\boldsymbol{\beta}_t)$ as the notation for measurement matrix.

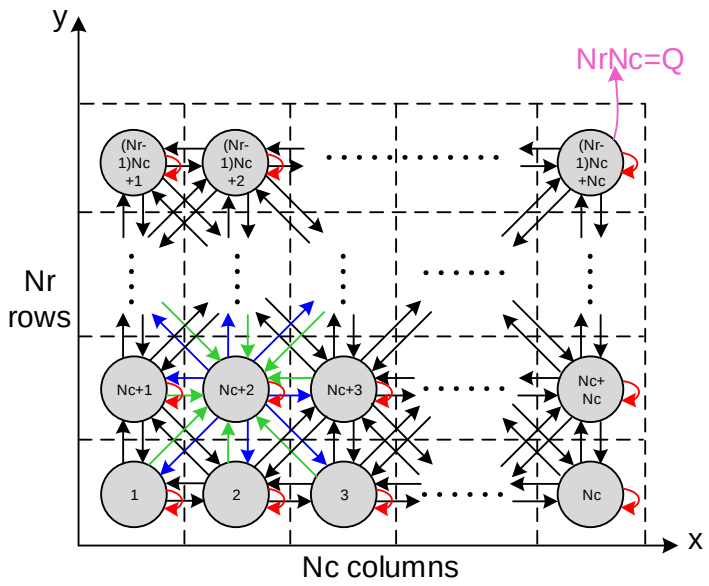

Figure 5.2: Markov chain representation of the geographical area. In the next time slot, the user either stays in the current grid cell or moves to one of the neighboring grid cells.

Assumption 2 (User Movement between Neighboring Cells)**.** *In the next time slot, we assume the user either stays in the current grid cell, or moves to one of the neighboring grid cells, as shown in Fig. 5.2. This is always satisfied in practice when the user's speed is upper bounded and the time slot duration is sufficiently small.*

About these two assumptions, we would like to add several remarks.

Remark 5.1. For Assumption 1, even if its condition is not completely satisfied, the proposed algorithm still can work well as verified in the simulations. In particular, the algorithm works as long as there exist active LOS paths between the user and a few nearby BSs (e.g., 3 BSs with active LOS paths are sufficient to achieve a good localization accuracy as illustrated in Fig. 5.6 of the simulation). Hence, the algorithm is quite robust to the LOS blocking scenario.

Remark 5.2. For Assumption 2, if we consider $5\text{m}\times5\text{m}$ grid size, the duration of one time slot is $\bar{\tau}$ seconds, then this assumption can be satisfied when the user's velocity is upbounded as $|v_x| \leq 10/\bar{\tau}, |v_y| \leq 10/\bar{\tau}$, where v_x (v_y) is the velocity component at x (y) direction. When $\bar{\tau} = 1ms$, the velocity upper bound is $10km/s$, which can be met in most cases.

Remark 5.3. These two assumptions are quite general and can be satisfied in many practical scenarios. In this chapter, we concentrate on equal-polarized and omnidirectional antennas for clarity. However, the proposed algorithm can be applied to more general scenarios. For example, as long as the LOS paths are within the receive beamwidth of the antenna array of the chosen BSs, the proposed algorithm can be readily applied to the directional antennas. For

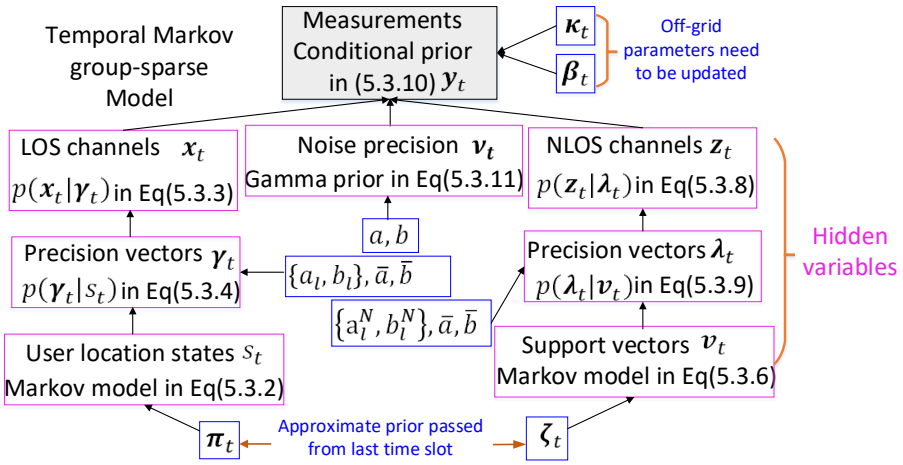

Figure 5.3: Temporal Markov group-sparse model for the LOS and NLOS channels.

non-equal-polarized antennas, the channel model is still a function of AoAs corresponding to the LOS and NLOS paths [103]. By introducing the location grid and angle grid, the whole problem can still be transformed into a compressive sensing problem, but with different measurement matrices. We omit the details for conciseness. The proposed algorithm can be directly used to solve it. Therefore, the proposed algorithm works for more general scenarios.

5.3 D-VBI Problem Formulation

The sparse channel representation in (5.2.11) lacks a probability model for $\boldsymbol{x}_t$ and $\mathbf{z}_t$. Such a probability model provides the foundation for efficient user location tracking in massive MIMO systems. The Markov user mobility model will induce temporal correlation in the support of massive MIMO channels. In addition, the cooperative localization of multiple BSs based on the location grid induces a GS structure of the LOS channels at each time slot. In order to jointly capture the PTC and GS of massive MIMO channels, we propose a TMGS model in this section. Besides capturing the first-order sparsity structure of the massive MIMO channels, another motivation for TMGS probability model is that it enables closed-form update solution for the proposed D-VBI algorithm.

Fig. 5.3 illustrates the high level structures of the TMGS for the LOS channels $(\mathbf{x}_1, \cdots, \mathbf{x}_T)$ and NLOS channels $(\mathbf{z}_1, \cdots, \mathbf{z}_T)$.

5.3.1 Temporal-Markov-Group-Sparse Mobility Model for the LOS Channel

The TMGS prior of the LOS channels is a three-layer hierarchical prior. The first layer of random variable is the user location state $s_t, \forall t$, which represents the coarse position of the user among the Q location grids. The second layer of random variable is precision vector $\boldsymbol{\gamma}_t = [\boldsymbol{\gamma}_{t,1}; \cdots; \boldsymbol{\gamma}_{t,L}] \in \mathbb{R}^{LQ}, \forall t$ with $\boldsymbol{\gamma}_{t,l} = [\gamma_{t,l,1}, \cdots, \gamma_{t,l,Q}]^T \in \mathbb{R}^Q$, where $\gamma_{t,l,q}$ represents the precision (inverse of the variance) of $x_{t,l,q}$. The third layer of random variable is the LOS channel vector $\mathbf{x}_t, \forall t$. For convenience, denote a time series of vectors $\{\mathbf{x}_t\}_{t=1}^T$ as $\mathbf{x}_{1:T}$ (same for $\boldsymbol{\gamma}_{1:T}$, $s_{1:T}$, $\mathbf{z}_{1:T}$, $\boldsymbol{\lambda}_{1:T}$,$\mathbf{v}_{1:T}$). Then the TMGS prior distribution (joint distribution of $\mathbf{x}_{1:T}$, $\boldsymbol{\gamma}_{1:T}$ and $s_{1:T}$) is given by

$$p\left(\mathbf{x}_{1:T}, \boldsymbol{\gamma}_{1:T}, s_{1:T}\right) = \prod_{t=1}^{T} \underbrace{p\left(\mathbf{x}_t \left| \boldsymbol{\gamma}_t \right.\right)}_{\text{LOS Channels}} \underbrace{p\left(\boldsymbol{\gamma}_t \left| s_t \right.\right)}_{\text{Precisions}} \underbrace{p\left(s_{1:T}\right)}_{\text{Location States}}, \tag{5.3.1}$$

where the location states $s_{1:T}$ form a Markov chain, as detailed in Subsection 5.3.1.1, and the conditional priors of the LOS channel precisions and LOS channels form a group sparse model, as detailed in Subsection 5.3.1.2.

5.3.1.1 Mobility Model for User Location States

The Markov model (MM) is considered for user movement prediction in [93]. Based on the Assumption 2, we can use a Markov chain to model the user's mobility, given by

$$p\left(s_{1:T}\right) = p(s_1) \prod_{t=2}^{T} p(s_t \left| s_{t-1} \right.), \tag{5.3.2}$$

where $p(s_t \left| s_{t-1} \right.)$ is characterized by the transition probability matrix (TPM) $\mathbf{G}_t = \{g_{t,j,i}\} \in \mathbb{R}^{Q \times Q}$, in which $g_{t,j,i} = P\left(s_{t+1} = j \left| s_t = i \right.\right)$ is the transition probability between state i and state j, $1 \leq i, j \leq Q$. Since the movements are only between neighboring states, only the neighbor and self transition probabilities are non-zero. Assume the Q grid cells could be divided into N_r rows and N_c columns, we index the grid cells along the rows, as shown in Fig. 5.2. For transitions from state i, i.e., $i = Nc + 2$ in Fig. 5.2, the indices of (potentially) nonzero elements of $\mathbf{G}_t[:, i]$ are given by $\{1, 2, 3, N_c + 1, N_c + 2, N_c + 3, 2N_c + 1, 2N_c + 2, 2N_c + 3\}$. In practice, the prior distribution $p(s_1)$ can be obtained via the other localization technologies (such as GPS), and $\mathbf{G}_t$ could be dynamically updated according to the training

data to improve the robustness w.r.t. the inaccurate prior information on $\mathbf{G}_t$.

5.3.1.2 Group Sparse Model for LOS Channels

To allow the flexibility to model local characteristics of the signal, a non-stationary Gaussian prior distribution with a distinct precision $\gamma_{t,l,q}$ for each element $x_{t,l,q}$ of $\mathbf{x}_t$ is considered, i.e.,

$$p\left(\mathbf{x}_t|\boldsymbol{\gamma}_t\right) = \prod_{q=1}^{Q}\prod_{l=1}^{L}\mathcal{CN}\left(x_{t,l,q};0,\gamma_{t,l,q}^{-1}\right). \tag{5.3.3}$$

The precisions $\gamma_{t,l,q}$ are further constrained by treating them as random variables and imposing a Gamma prior distribution to them. The Gamma prior is selected because it is conjugate to the Gaussian, hence the associated Bayesian inference can be performed in closed form [28, 99]. The conditional prior of precision vector $\boldsymbol{\gamma}_t$ is given by

$$p\left(\boldsymbol{\gamma}_t|s_t\right) = \prod_{l=1}^{L}\prod_{q=1}^{Q}\Gamma\left(\gamma_{t,l,q};a_l,b_l\right)^{1(s_t=q)}\Gamma\left(\gamma_{t,l,q};\bar{a},\bar{b}\right)^{1(s_t\neq q)}, \tag{5.3.4}$$

where $\Gamma\left(\rho;a,b\right)$ is a Gamma hyper-prior with shape parameter a and rate parameter b.

The priors in (5.3.3) and (5.3.4) can be used to capture the group-sparsity of LOS channels, as explained below. When $s_t = q$, there is an active LOS path from the q-th grid cell to each BS. In this case, the shape and rate parameter a_l, b_l of the precision $\gamma_{t,l,q}$ should be chosen such that $\frac{a_l}{b_l} = \mathbb{E}\left[\gamma_{t,l,q}\right] = \Theta(1), \forall q, l$, since the variance $\gamma_{t,l,q}^{-1}$ of $x_{t,l,q}$ is $\Theta(1)$ when it is active. When $s_t \neq q$, there is no active LOS path from the q-th grid cell to each BS. In this case, the shape and rate parameter $\bar{a}$, $\bar{b}$ of the precision $\gamma_{t,l,q}$ should be chosen such that $\frac{\bar{a}}{\bar{b}} = \mathbb{E}\left[\gamma_{t,l,q}\right] \gg 1, \forall q, l$, since the variance $\gamma_{t,l,q}^{-1}$ of $x_{t,l,q}$ is close to zero when it is inactive. Note that the channel energies of the LOS channels $\{\mathbf{x}_{t,l}\}_{l=1}^{L}$ associated with different BSs all concentrate on the same index s_t. Therefore, the LOS channel vectors $\mathbf{x}_t$ can be separated into Q blocks with block size L, i.e., the q-th block is constituted by $[x_{t,1,q},\cdots,x_{t,L,q}]$. For blocks satisfying $s_t = q$, the conditional priors in (5.3.4) assign smaller precisions to make the block elements deviate from zero. For blocks satisfying $s_t \neq q$, the conditional priors in (5.3.4) assign larger precisions to make the block elements concentrate on zero. Meanwhile, we use different precisions within each block to capture the different path gains for different BSs.

5.3.2 Temporal-Markov-Group-Sparse Mobility Model for the NLOS Channel

The TMGS prior of NLOS channels is a three-layer hierarchical prior. The first layer of random variable is support vector $\mathbf{v}_t = [\mathbf{v}_{t,1}; \cdots; \mathbf{v}_{t,L}] \in \mathbb{R}^M, \forall t$ with $\mathbf{v}_{t,l} = [v_{t,l,1}, \cdots, v_{t,l,M_l}]^T \in \mathbb{R}^{M_l}$, where $v_{t,l,m} \in \{0, 1\}$ represents whether there is an active NLOS path arriving at the BS l from the m-th angle grid. The second layer of random variable is precision vector $\boldsymbol{\lambda}_t = [\boldsymbol{\lambda}_{t,1}; \cdots; \boldsymbol{\lambda}_{t,L}] \in \mathbb{R}^M, \forall t$ with $\boldsymbol{\lambda}_{t,l} = [\lambda_{t,l,1}, \cdots, \lambda_{t,l,M_l}]^T \in \mathbb{R}^{M_l}$, where $\lambda_{t,l,m}$ represents the precision (inverse of the variance) of $z_{t,l,m}$. The third layer of random variable is the NLOS channel vectors $\mathbf{z}_t, \forall t$. Then the TMGS prior distribution (joint distribution of $\mathbf{z}_{1:T}$, $\boldsymbol{\lambda}_{1:T}$ and $\mathbf{v}_{1:T}$) is given by

$$p\left(\mathbf{z}_{1:T}, \boldsymbol{\lambda}_{1:T}, \mathbf{v}_{1:T}\right) = \prod_{t=1}^{T} \underbrace{p\left(\mathbf{z}_t \left| \boldsymbol{\lambda}_t \right.\right)}_{\text{NLOS Channels}} \underbrace{p\left(\boldsymbol{\lambda}_t \left| \mathbf{v}_t \right.\right)}_{\text{Precisions}} \underbrace{p\left(\mathbf{v}_{1:T}\right)}_{\text{SupportVectors}}, \tag{5.3.5}$$

where the elements of $\mathbf{v}_t$ across time form independent Markov chains, which will be elaborated in Subsection 5.3.2.1, and the conditional priors of the NLOS channel precisions and the NLOS channels form a sparse model, which will be elaborated in Subsection 5.3.2.2.

5.3.2.1 Markov Model for Support Vector

Due to the MM user mobility model and the slowly changing propagation environment, the support vector $\mathbf{v}_t$ changes slowly over time, i.e., if $v_{t,l,m} = 1$, there is a high probability that $v_{t+1,l,m} = 1$. Such temporal correlation of support vector $\mathbf{v}_t$ could be modeled by an MM process [66] as follows:

$$p\left(\mathbf{v}_{1:T}\right) = \prod_{l=1}^{L} \prod_{m=1}^{M_l} \left(p\left(v_{1,l,m}\right) \prod_{t=2}^{T} p\left(v_{t,l,m} \left| v_{t-1,l,m} \right.\right) \right), \tag{5.3.6}$$

with the transition probability given by

$$p\left(v_{t,l,m} \left| v_{t-1,l,m} \right.\right) = \begin{cases} \rho_{01}^{v_{t,l,m}} \left(1 - \rho_{01}\right)^{1 - v_{t,l,m}} & v_{t-1,l,m} = 0 \\ \rho_{10}^{1 - v_{t,l,m}} \left(1 - \rho_{10}\right)^{v_{t,l,m}} & v_{t-1,l,m} = 1 \end{cases}, \tag{5.3.7}$$

where $\rho_{01} = p\left(v_{t,l,m} = 1 \left| v_{t-1,l,m} = 0 \right.\right)$ and $\rho_{10} = p\left(v_{t,l,m} = 0 \left| v_{t-1,l,m} = 1 \right.\right)$. The initial distribution $p\left(v_{1,l,m}\right)$ is set to be the steady state distribution, i.e., $p\left(v_{1,l,m} = 1\right) = \frac{\rho_{01}}{\rho_{01} + \rho_{10}}, \forall l, m$.

Note that the Markov parameters $\{\rho_{01}, \rho_{10}\}$ characterize the degree of temporal correlation of the NLOS paths. Specifically, smaller ρ_{01} and ρ_{10} lead to highly correlated supports across time, which means the propagation environment between the user and BSs is changing slowly. Larger ρ_{01} and ρ_{10} can allow support to change substantially across time, which means the propagation environment is changing significantly. As such, the MM for the NLOS support vectors can be used to model various channel realizations in practice. We consider the steadily changing propagation environment between user and BSs. Therefore, the transition probabilities ρ_{01} and ρ_{10} are (almost) static during the tracking process that we are interested in. The statistic parameters $\{\rho_{01}, \rho_{10}\}$ could be automatically learned based on the EM framework during the recovery process [66].

5.3.2.2 Sparse Model for NLOS Channels

Similar to LOS channels, we assign a non-stationary Gaussian prior distribution with a distinct precision $\lambda_{t,l,m}$ for each entry of $\mathbf{z}_t$ as follows:

$$p\left(\mathbf{z}_t \left| \boldsymbol{\lambda}_t \right.\right) = \prod_{l=1}^{L} \prod_{m=1}^{M_l} \mathcal{CN}\left(z_{t,l,m}; 0, \lambda_{t,l,m}^{-1}\right). \tag{5.3.8}$$

The conditional prior of precision vector $\boldsymbol{\lambda}_t$ is given by

$$p\left(\boldsymbol{\lambda}_t \left| \mathbf{v}_t \right.\right) = \prod_{l,m} \Gamma\left(\lambda_{t,l,m}; a_l^N, b_l^N\right)^{v_{t,l,m}} \Gamma\left(\lambda_{t,l,m}; \bar{a}, \bar{b}\right)^{1-v_{t,l,m}}. \tag{5.3.9}$$

The priors in (5.3.8) and (5.3.9) can be used to capture the sparsity of NLOS channels, as explained below. When $v_{t,l,m} = 1$, there is an active NLOS path from the m-th angle grid arriving at BS l. In this case, $a_l^N / b_l^N = \mathbb{E}\left[\lambda_{t,l,m}\right] = \Theta(1), \forall l, m$, since the variance $\lambda_{t,l,m}^{-1}$ of $z_{t,l,m}$ is $\Theta(1)$ if it is active. When $v_{t,l,m} = 0$, there is no active NLOS path from the m-th angle grid arriving at BS l. In this case, $\frac{\bar{a}}{\bar{b}} = \mathbb{E}\left[\lambda_{t,l,m}\right] \gg 1, \forall l, m$, since the variance $\lambda_{t,l,m}^{-1}$ of $z_{t,l,m}$ is close to zero if it is inactive.

Remark 5.4. Note that the Gaussian assumption on the path coefficients in (5.3.3) and (5.3.8) is just for the convenience of inducing sparsity through the hierarchical model specified in Bayesian-inference-based algorithm, and this approximation doesn't sacrifice the performance much as validated in the simulations.

5.3.3 D-VBI Formulation with TMGS Prior

Under the assumption of the complex Gaussian noise, we have

$$p\left(\mathbf{y}_{t,l}\left|\mathbf{x}_{t,l},\mathbf{z}_{t,l},\nu_{t,l};\boldsymbol{\kappa}_t,\boldsymbol{\beta}_{t,l}\right.\right)=\mathcal{CN}\left(\mathbf{y}_{t,l};\mathbf{A}_{t,l}\left(\boldsymbol{\kappa}_t\right)\mathbf{x}_{t,l}+\mathbf{B}_{t,l}\left(\boldsymbol{\beta}_{t,l}\right)\mathbf{z}_{t,l},\nu_{t,l}^{-1}\boldsymbol{I}\right), \tag{5.3.10}$$

where $\nu_{t,l}=\sigma_{t,l}^{-2}$ represents the noise precision for BS l. Since $\nu_{t,l}$ is usually unknown, we model it as a Gamma hyper-prior

$$p(\nu_{t,l})=\Gamma(\nu_{t,l};a,b), \tag{5.3.11}$$

where we set $a,b\rightarrow 0$ as in [27]. Then the joint prior of $\boldsymbol{\nu}_t=[\nu_{t,1},\cdots,\nu_{t,L}]$ is given by $p\left(\boldsymbol{\nu}_t\right)=\prod_{l=1}^{L}p(\nu_{t,l})$.

Denote the complete hidden variables as $\boldsymbol{h}_t=\{\mathbf{x}_t,\mathbf{z}_t,\boldsymbol{\gamma}_t,\boldsymbol{\lambda}_t,s_t,\mathbf{v}_t,\boldsymbol{\nu}_t\}$. For convenience, we use $\boldsymbol{h}_{t,n}$ to denote an individual variable in $\boldsymbol{h}_t$. Let $\mathcal{H}=\{n|\forall\boldsymbol{h}_{t,n}\in\boldsymbol{h}_t\}$. In order to recursively track the user's location $\mathbf{p}_t$ ($1\leq t\leq T$), at each time slot, our primary goal is to estimate the user location state (coarse location) s_t and the location grid offset $\boldsymbol{\kappa}_t$ given the observations up to the current time slot, $\mathbf{y}_{1:t}$ in model (5.2.11). In particular, the offset parameters $\boldsymbol{\kappa}_{1:t},\boldsymbol{\beta}_{1:t}$ are obtained by maximizing the likelihood function as follows:

$$\begin{aligned}\left[\hat{\boldsymbol{\kappa}}_{1:t},\hat{\boldsymbol{\beta}}_{1:t}\right]&=\arg\max_{\boldsymbol{\kappa}_{1:t},\boldsymbol{\beta}_{1:t}}\ln p(\mathbf{y}_{1:t};\boldsymbol{\kappa}_{1:t},\boldsymbol{\beta}_{1:t})\\&=\arg\max_{\boldsymbol{\kappa}_{1:t},\boldsymbol{\beta}_{1:t}}\ln\int p\left(\boldsymbol{h}_{1:t},\mathbf{y}_{1:t};\boldsymbol{\kappa}_{1:t},\boldsymbol{\beta}_{1:t}\right)d\boldsymbol{h}_{1:t}.\end{aligned} \tag{5.3.12}$$

Then, for given estimates of offset parameters $\boldsymbol{\kappa}_{1:t}=\hat{\boldsymbol{\kappa}}_{1:t},\boldsymbol{\beta}_{1:t}=\hat{\boldsymbol{\beta}}_{1:t}$, we aim at calculating the marginal posteriors $p\left(s_t\left|\mathbf{y}_{1:t};\boldsymbol{\kappa}_{1:t},\boldsymbol{\beta}_{1:t}\right.\right)$ by performing the Bayesian inference for s_t, then the estimation $\hat{s}_t$ of s_t could be given by the MAP probability estimate as follows:

$$\hat{s}_t=\arg\max_{s_t\in\{1,\cdots,Q\}}p\left(s_t\left|\mathbf{y}_{1:t};\boldsymbol{\kappa}_{1:t},\boldsymbol{\beta}_{1:t}\right.\right). \tag{5.3.13}$$

$p\left(s_t\left|\mathbf{y}_{1:t};\boldsymbol{\kappa}_{1:t},\boldsymbol{\beta}_{1:t}\right.\right)$ can be calculated through

$$\begin{aligned} p\left(s_t\left|\mathbf{y}_{1:t};\boldsymbol{\kappa}_{1:t},\boldsymbol{\beta}_{1:t}\right.\right) &\propto \int p\left(\boldsymbol{h}_{1:t},\mathbf{y}_{1:t};\boldsymbol{\kappa}_{1:t},\boldsymbol{\beta}_{1:t}\right) d\left(\boldsymbol{h}_{1:t}\right)_{-s_t} \\ \propto \int \prod_{\tau=1}^{t} & p\left(\mathbf{y}_\tau\left|\mathbf{x}_\tau,\mathbf{z}_\tau,\boldsymbol{\nu}_\tau;\boldsymbol{\kappa}_\tau,\boldsymbol{\beta}_\tau\right.\right) p\left(\boldsymbol{\nu}_\tau\right) p\left(\mathbf{x}_\tau\left|\boldsymbol{\gamma}_\tau\right.\right) p\left(\boldsymbol{\gamma}_\tau\left|s_\tau\right.\right) p\left(s_\tau\left|s_{\tau-1}\right.\right) \\ & \times p\left(\mathbf{z}_\tau\left|\boldsymbol{\lambda}_\tau\right.\right) p\left(\boldsymbol{\lambda}_\tau\left|\mathbf{v}_\tau\right.\right) p\left(\mathbf{v}_\tau\left|\mathbf{v}_{\tau-1}\right.\right) d\left(\boldsymbol{h}_{1:t}\right)_{-s_t}, \end{aligned} \tag{5.3.14}$$

where $p(s_1\left|s_0\right.) = p(s_1)$ and $p\left(\mathbf{v}_1\left|\mathbf{v}_0\right.\right) = p\left(\mathbf{v}_1\right)$, $\propto$ denotes the left is proportional to the right, and $\left(\boldsymbol{h}_{1:t}\right)_{-s_t}$ denotes the vector collections $\{\boldsymbol{h}_\tau\}_{\tau=1}^{t}$ except for the element s_t. Finally, the user's location estimation $\hat{\mathbf{p}}_t$ at time slot t can be given by

$$\hat{\mathbf{p}}_t = \boldsymbol{\phi}_{\hat{s}_t} + \hat{\boldsymbol{\kappa}}_{t,\hat{s}_t}. \tag{5.3.15}$$

Challenge 1: It is very challenging to calculate the exact posterior in (5.3.13) and the likelihood function in (5.3.12), because it is hard to calculate the closed form of the integration in (5.3.14) due to the complicated priors of the hidden variables and the correlations between hidden variables.

Challenge 2: Moreover, the objective in (5.3.12) is a high-dimensional non-convex function. It is very difficult to directly use the optimization method (i.e., gradient method) to solve problem (5.3.12).

To overcome these challenges, we propose a D-VBI algorithm to find the approximation of the posterior distribution in (5.3.13) and the approximate stationary point of problem (5.3.12).

5.4 D-VBI Algorithm for User Location Tracking

In the proposed D-VBI algorithm, the BSs first send the received signals $\mathbf{y}_{t,l}, \forall l$ to the cloud. Then the cloud runs the D-VBI algorithm to track s_t along with the off-grid parameters $\boldsymbol{\kappa}_t$ recursively, based on the noisy measurement and the messages passed from the last time slot.

5.4.1 Outline of Dynamic Variational Bayesian Inference

In (5.3.12), we need to jointly optimize $\boldsymbol{\kappa}_{1:t}, \boldsymbol{\beta}_{1:t}$ based on all the observations $\mathbf{y}_{1:t}$. One possible solution is to store all the available observations $\mathbf{y}_{1:t}$ and perform a joint optimization

of $\boldsymbol{\kappa}_{1:t}, \boldsymbol{\beta}_{1:t}$ at each time t. However, the memory cost and computational complexity of such a brute-force solution would become unacceptable for large t. To address this challenge, we propose a D-VBI algorithm, which is based on problem decomposition and approximation. Specifically,

1. **Problem Decomposition:** We first decompose the joint optimization problem (5.3.12) into subproblems based on Lemma 5.1, such that the subproblem at time t only involves the optimization of $\boldsymbol{\kappa}_t, \boldsymbol{\beta}_t$, and $q\left(\boldsymbol{h}_t\right)$ ($q(\boldsymbol{h}_t)$ is a PDF of hidden variables $\boldsymbol{h}_t$) based on the current observation $\mathbf{y}_t$ and the posterior $p\left(s_{t-1}, \mathbf{v}_{t-1} \left| \mathbf{y}_{1:t-1}; \boldsymbol{\kappa}^{\star}_{1:t-1}, \boldsymbol{\beta}^{\star}_{1:t-1} \right.\right)$, where $\boldsymbol{\kappa}^{\star}_{1:t}$ and $\boldsymbol{\beta}^{\star}_{1:t}$ denote the optimal solution of (5.3.12).

2. **Problem Approximation:** Then we obtain an approximate subproblem for time slot t by adding an additional factorized constraint on $q\left(\boldsymbol{h}_t\right)$ and replacing the exact posterior $p\left(s_{t-1}, \mathbf{v}_{t-1} \left| \mathbf{y}_{1:t-1}; \boldsymbol{\kappa}^{\star}_{1:t-1}, \boldsymbol{\beta}^{\star}_{1:t-1} \right.\right)$ with an approximate posterior that can be obtained from the messages passed from the previous time slot.

3. **inexact Block Coordinate Descent:** Finally, we propose an inexact block coordinate descent (BCD) algorithm to find a stationary solution $\left(\boldsymbol{\kappa}^{*}_{t}, \boldsymbol{\beta}^{*}_{t}, q^{*}\left(\boldsymbol{h}_t\right)\right)$ of the approximate subproblem for time slot t. Then $\boldsymbol{\kappa}^{*}_{t}, \boldsymbol{\beta}^{*}_{t}$ is an approximation of $\boldsymbol{\kappa}^{\star}_{t}, \boldsymbol{\beta}^{\star}_{t}$ and $q^{*}\left(\boldsymbol{h}_t\right)$ is the approximation of the exact posterior $p\left(\boldsymbol{h}_t \left| \mathbf{y}_{1:t}; \boldsymbol{\kappa}^{\star}_{1:t}, \boldsymbol{\beta}^{\star}_{1:t} \right.\right)$. At last, the messages $\left\{q^{*}\left(s_t\right), q^{*}\left(\mathbf{v}_t\right)\right\}$ are passed to the next time slot $t+1$.

In the following, we will elaborate the problem decomposition and approximation in Section 5.4.2, and the inexact BCD algorithm in Section 5.4.3.

5.4.2 Problem Decomposition and Approximation

The problem decomposition is based on the following Lemma.

Lemma 5.1 (Problem Decomposition)**.** *Let $\boldsymbol{\kappa}^{\star}_{1:t}, \boldsymbol{\beta}^{\star}_{1:t}$ denote the optimal solution of (5.3.12). Consider the following optimization problem at time t*

$$\max_{q(\boldsymbol{h}_t), \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t} \int q\left(\boldsymbol{h}_t\right) \ln \frac{p\left(\boldsymbol{h}_t, \mathbf{y}_{1:t}; \boldsymbol{\kappa}^{\star}_{1:t-1}, \boldsymbol{\beta}^{\star}_{1:t-1}, \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right)}{q\left(\boldsymbol{h}_t\right)} d\boldsymbol{h}_t, \tag{5.4.1}$$

where

$$
\begin{aligned}
& p\left(\boldsymbol{h}_t, \mathbf{y}_{1:t}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}, \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right) \\
\propto & p\left(\mathbf{y}_t \left|\boldsymbol{x}_t, \mathbf{z}_t, \boldsymbol{\nu}_t; \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right.\right) p\left(\boldsymbol{\nu}_t\right) p\left(\mathbf{x}_t \left|\boldsymbol{\gamma}_t\right.\right) p\left(\boldsymbol{\gamma}_t \left|s_t\right.\right) p\left(\mathbf{z}_t \left|\boldsymbol{\lambda}_t\right.\right) p\left(\boldsymbol{\lambda}_t \left|\mathbf{v}_t\right.\right) \\
& \times \sum_{s_{t-1}, \mathbf{v}_{t-1}} p\left(s_t \left|s_{t-1}\right.\right) p\left(\mathbf{v}_t \left|\mathbf{v}_{t-1}\right.\right) p\left(s_{t-1}, \mathbf{v}_{t-1} \left|\mathbf{y}_{1:t-1}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}\right.\right). \qquad (5.4.2)
\end{aligned}
$$

When $t = 1$, the optimization problem (5.4.1) is reduced to

$$
\max_{q(\boldsymbol{h}_1), \boldsymbol{\kappa}_1, \boldsymbol{\beta}_1} \int q\left(\boldsymbol{h}_1\right) \ln\left(\frac{p\left(\boldsymbol{h}_1, \mathbf{y}_1; \boldsymbol{\kappa}_1, \boldsymbol{\beta}_1\right)}{q\left(\boldsymbol{h}_1\right)}\right) d\boldsymbol{h}_1. \qquad (5.4.3)
$$

Then the optimal solutions of problem (5.4.1), i.e., $\boldsymbol{\kappa}_t^{\star}, \boldsymbol{\beta}_t^{\star}$ and $q^{\star}\left(\boldsymbol{h}_t\right) = p\left(\boldsymbol{h}_t \left|\mathbf{y}_{1:t}; \boldsymbol{\kappa}_{1:t}^{\star}, \boldsymbol{\beta}_{1:t}^{\star}\right.\right)$ are the optimal solutions of original problem (5.3.12).

Please refer to Appendix 5.7.1 for the proof.

According to Lemma 5.1, both the optimal parameter $\boldsymbol{\kappa}_t^{\star}, \boldsymbol{\beta}_t^{\star}$ for the original problem (5.3.12), and the associated posterior $p\left(\boldsymbol{h}_t \left|\mathbf{y}_{1:t}; \boldsymbol{\kappa}_{1:t}^{\star}, \boldsymbol{\beta}_{1:t}^{\star}\right.\right)$ can be obtained by solving Subproblem (5.4.1). Note that Subproblem (5.4.1) only depends on the previous observations $\mathbf{y}_{1:t-1}$ and previous parameters $\boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}$ via the posterior $p\left(s_{t-1}, \mathbf{v}_{t-1} \left|\mathbf{y}_{1:t-1}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}\right.\right)$, as shown in (5.4.2). Therefore, if we can find a good approximation for the posterior $p\left(s_{t-1}, \mathbf{v}_{t-1} \left|\mathbf{y}_{1:t-1}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}\right.\right)$ based on the messages from the previous time slot, we can get rid of the intractable joint optimization of $\boldsymbol{\kappa}_{1:t}, \boldsymbol{\beta}_{1:t}$. Specifically, we have the following approximation

$$
p\left(s_{t-1}, \mathbf{v}_{t-1} \left|\mathbf{y}_{1:t-1}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}\right.\right) \approx q^{*}\left(s_{t-1}\right) q^{*}\left(\mathbf{v}_{t-1}\right), \qquad (5.4.4)
$$

where $q^{*}\left(s_{t-1}\right) \approx p\left(s_{t-1} \left|\mathbf{y}_{1:t-1}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}\right.\right)$ and $q^{*}\left(\mathbf{v}_{t-1}\right) \approx p\left(\mathbf{v}_{t-1} \left|\mathbf{y}_{1:t-1}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}\right.\right)$ are the approximate posteriors obtained from the previous time slot $t-1$ by solving the approximate subproblem $\mathscr{A}$ in (5.4.5) for time $t-1$. By replacing $p\left(s_{t-1}, \mathbf{v}_{t-1} \left|\mathbf{y}_{1:t-1}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}\right.\right)$ with its approximation $q^{*}\left(s_{t-1}\right) q^{*}\left(\mathbf{v}_{t-1}\right)$ and adding an additional factorized constraint on

$q(\boldsymbol{h}_t)$, the optimization problem in (5.4.1) could be simplified as

$$\mathscr{A}: \max_{q(\boldsymbol{h}_t),\boldsymbol{\kappa}_t,\boldsymbol{\beta}_t} \underbrace{\int q(\boldsymbol{h}_t) \ln\left(\frac{\hat{p}(\mathbf{y}_t, \boldsymbol{h}_t; \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t)}{q(\boldsymbol{h}_t)}\right) d\boldsymbol{h}_t}_{\mathcal{U}_t\left(q_{t,1:|\mathcal{H}|}, \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right)}, \tag{5.4.5}$$

$$\text{s.t.} \quad q(\boldsymbol{h}_t) = \prod_{n\in\mathcal{H}} q(\boldsymbol{h}_{t,n}). \tag{5.4.6}$$

We denote $q(\boldsymbol{h}_{t,n})$ as $q_{t,n}, \forall n \in \mathcal{H}$, and $q_{t,1:|\mathcal{H}|} = \left(q_{t,1}, \cdots, q_{t,|\mathcal{H}|}\right)$. In (5.4.5), $\hat{p}(\mathbf{y}_t, \boldsymbol{h}_t; \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t)$ is given by

$$\begin{aligned}
&\hat{p}(\mathbf{y}_t, \boldsymbol{h}_t; \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t) \\
=&p(\mathbf{y}_t|\boldsymbol{x}_t, \mathbf{z}_t, \boldsymbol{\nu}_t; \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t)\, p(\boldsymbol{\nu}_t)\, p(\mathbf{x}_t|\boldsymbol{\gamma}_t)\, p(\boldsymbol{\gamma}_t|s_t)\, p(\mathbf{z}_t|\boldsymbol{\lambda}_t)\, p(\boldsymbol{\lambda}_t|\mathbf{v}_t)\, \hat{p}(s_t)\, \hat{p}(\mathbf{v}_t),
\end{aligned} \tag{5.4.7}$$

in which the approximate priors of s_t and $\mathbf{v}_t$ are given by

$$\hat{p}(s_t) = \sum_{s_{t-1}} p(s_t|s_{t-1})\, q^*(s_{t-1}), \tag{5.4.8}$$

$$\hat{p}(\mathbf{v}_t) = \sum_{\mathbf{v}_{t-1}} p(\mathbf{v}_t|\mathbf{v}_{t-1})\, q^*(\mathbf{v}_{t-1}). \tag{5.4.9}$$

Since the objective $\mathcal{U}_t$ in Problem $\mathscr{A}$ involves both functions $q_{t,n}$'s and variables $\boldsymbol{\kappa}_t, \boldsymbol{\beta}_t$, it is difficult to find the optimal solution. We adopt an inexact BCD algorithm to find a stationary solution instead. Specifically, a stationary solution for Problem $\mathscr{A}$ is defined as follows.

Definition 5.1 (Stationary Solution)**.** $(q^*(\boldsymbol{h}_t), \boldsymbol{\kappa}_t^*, \boldsymbol{\beta}_t^*)$ is called a stationary solution of Problem $\mathscr{A}$ if

$$q_{t,n}^* = \arg\max_{q_{t,n}} \mathcal{U}_t\left(q_{t,n}, q_{t,-n}^*, \boldsymbol{\kappa}_t^*, \boldsymbol{\beta}_t^*\right),\ n \in \mathcal{H}$$

and $(\boldsymbol{\kappa}_t^*, \boldsymbol{\beta}_t^*)$ is a stationary point of $\max_{\boldsymbol{\kappa}_t,\boldsymbol{\beta}_t} \mathcal{U}_t\left(q_{t,1:|\mathcal{H}|}^*, \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right)$, where $q_{t,-n}^* = \left(q_{t,1:n-1}^*, q_{t,n+1:|\mathcal{H}|}^*\right)$, and $q^*(\boldsymbol{h}_t) = \prod_{n\in\mathcal{H}} q_{t,n}^*$.

5.4.3 Inexact Block Coordinated Descent

The inexact BCD optimizes each function/variable in an alternating way. Specifically, in the i-th iteration, we update $q_{t,n}$'s as

$$q_{t,n}^{(i+1)} = \arg\max_{q_{t,n}} \mathcal{U}_t \left(q_{t,n}, q_{t,-n}^{(i)}, \boldsymbol{\kappa}_t^{(i)}, \boldsymbol{\beta}_t^{(i)} \right), n \in \mathcal{H}, \tag{5.4.10}$$

where $(\cdot)^{(i)}$ stands for the i-th iteration, and $q_{t,-n}^{(i)} = \left(q_{t,1:n-1}^{(i+1)}, q_{t,n+1:|\mathcal{H}|}^{(i)} \right)$. The update formulas for $q_{t,n}^{(i+1)}$'s are summarized in the following lemma.

Lemma 5.2 (Solution of (5.4.10))**.** *Problem (5.4.10) has a unique solution given by [28] :*

$$\ln q_{t,n}^{(i+1)} \propto \begin{cases} \left\langle \ln p \left(\mathbf{y}_t, \boldsymbol{h}_t; \boldsymbol{\kappa}_t^{(i)}, \boldsymbol{\beta}_t^{(i)} \right) \right\rangle_{\prod q_{t,-n}^{(i)}}, & t = 1 \\ \left\langle \ln \hat{p} \left(\mathbf{y}_t, \boldsymbol{h}_t; \boldsymbol{\kappa}_t^{(i)}, \boldsymbol{\beta}_t^{(i)} \right) \right\rangle_{\prod q_{t,-n}^{(i)}}, & t > 1 \end{cases} \tag{5.4.11}$$

for $n \in \mathcal{H}$, where $\langle f(x) \rangle_{q(x)} = \int f(x) q(x) dx$.

On the other hand, we use the gradient method to update $\boldsymbol{\kappa}_t, \boldsymbol{\beta}_t$ as

$$\boldsymbol{\kappa}_t^{(i+1)} = \boldsymbol{\kappa}_t^{(i)} + \Delta_{\kappa_t} \cdot \boldsymbol{\xi}_{\kappa_t^{(i)}}^{(i+1)}, \tag{5.4.12}$$

$$\boldsymbol{\beta}_t^{(i+1)} = \boldsymbol{\beta}_t^{(i)} + \Delta_{\beta_t} \cdot \boldsymbol{\xi}_{\beta_t^{(i)}}^{(i+1)}, \tag{5.4.13}$$

where $\boldsymbol{\xi}_{\kappa_t^{(i)}}^{(i+1)}$ and $\boldsymbol{\xi}_{\beta_t^{(i)}}^{(i+1)}$ are the derivatives of the objective function $\mathcal{U}_t$ w.r.t. $\boldsymbol{\kappa}_t$ and $\boldsymbol{\beta}_t$ at point $\left(q^{(i+1)}(\boldsymbol{h}_t), \boldsymbol{\kappa}_t^{(i)}, \boldsymbol{\beta}_t^{(i)} \right)$ and $\left(q^{(i+1)}(\boldsymbol{h}_t), \boldsymbol{\kappa}_t^{(i+1)}, \boldsymbol{\beta}_t^{(i)} \right)$, respectively, and Δ_{κ_t} and Δ_{β_t} are the stepsizes that can be determined by the Armijo rule [104]. Alternatively, we may use a fixed stepsize, as mentioned in [27], to reduce the computational complexity. The detailed expressions of $\boldsymbol{\xi}_{\kappa_t^{(i)}}^{(i+1)}$ and $\boldsymbol{\xi}_{\beta_t^{(i)}}^{(i+1)}$ are given in Appendix 5.7.2.

The above update rules guarantee that the objective function $\mathcal{U}_t$ is non-decreasing and the inexact BCD algorithm will converge to stationary points.

Lemma 5.3 (Convergence of Inexact BCD)**.** *The update rules in (5.4.11), (5.4.12) and (5.4.13) give non-decreasing sequences $\mathcal{U}_t \left(q_{t,1:|\mathcal{H}|}^{(i)}, \boldsymbol{\kappa}_t^{(i)}, \boldsymbol{\beta}_t^{(i)} \right)$ for $i = 1, 2, 3, \cdots$. Every limiting point of the iterates $\left\{ q_{t,1:|\mathcal{H}|}^{(i)}, \boldsymbol{\kappa}_t^{(i)}, \boldsymbol{\beta}_t^{(i)} \right\}$ generated by the inexact BCD algorithm is a stationary solution of Problem $\mathscr{A}$ in (5.4.5).*

Please refer to Appendix 5.7.3 for the proof.

5.4.4 Closed-Form Update for the Posterioris $q_{t,n}$'s

Based on Lemma 5.2, the update equations of $q\left(\boldsymbol{h}_{t,n}\right), \forall n \in \mathcal{H}$ are given in the following. For conciseness, we omit the iteration index i. The detailed derivations can be found in Appendix 5.7.4. Note that $\langle\cdot\rangle_{\boldsymbol{h}_{t,n}}$ is equivalent to $\langle\cdot\rangle_{q_{t,n}}$, $\langle f\left(\boldsymbol{h}_{t,n}\right)\rangle$ is equivalent to $\langle f\left(\boldsymbol{h}_{t,n}\right)\rangle_{q_{t,n}}$.

5.4.4.1 Approximate prior of s_t and $\mathbf{v}_t$

Let $q^*\left(s_{t-1}\right) = \sum_{q=1}^{Q} \tilde{\pi}_{t-1,q}\delta\left(s_{t-1}-q\right)$ and

$$q^*\left(v_{t-1,l,m}\right) = \tilde{\zeta}_{t-1,l,m}\delta\left(v_{t-1,l,m}-1\right) + \left(1-\tilde{\zeta}_{t-1,l,m}\right)\delta\left(v_{t-1,l,m}\right), \forall l,m$$

denote the messages passed from the previous time slot $t-1$, where the posterior probabilities $\tilde{\pi}_{t-1,q}, \forall q$ and $\tilde{\zeta}_{t-1,l,m}, \forall l,m$ can be calculated through (5.4.22) and (5.4.24) at time slot $t-1$. Let $\pi_{t,q}$ and $\zeta_{t,l,m}$ denote the approximate prior probability of $p(s_t = q)$ and $p(v_{t,l,m} = 1)$. According to (5.4.8) and (5.4.9), we have

$$\hat{p}\left(s_t\right) = \sum_{q=1}^{Q} \pi_{t,q}\delta\left(s_t - q\right), \tag{5.4.14}$$

$$\hat{p}\left(\mathbf{v}_t\right) = \prod_{l,m} \zeta_{t,l,m}\delta\left(v_{t,l,m}-1\right) + \left(1-\zeta_{t,l,m}\right)\delta\left(v_{t,l,m}\right), \tag{5.4.15}$$

where $\pi_{t,q} = \mathbf{G}_{t-1}[q,:]\tilde{\boldsymbol{\pi}}_{t-1}$ and $\zeta_{t,l,m} = \tilde{\zeta}_{t-1,l,m}\left(1-\rho_{10}\right) + \left(1-\tilde{\zeta}_{t-1,l,m}\right)\rho_{01}$.

5.4.4.2 Update for $\boldsymbol{\nu}_t$

For given $q(\mathbf{x}_t)$ and $q(\mathbf{z}_t)$, $q\left(\boldsymbol{\nu}_t\right)$ can be derived as

$$q\left(\boldsymbol{\nu}_t\right) = \prod_{l=1}^{L} q(\nu_{t,l}) = \prod_{l=1}^{L} \Gamma\left(\nu_{t,l}; c_{t,l}, d_{t,l}\right), \tag{5.4.16}$$

where $c_{t,l} = N_l + a$, $d_{t,l} = \left\langle \left\| \mathbf{y}_{t,l} - \mathbf{A}_{t,l}\left(\boldsymbol{\kappa}_t\right)\mathbf{x}_{t,l} - \mathbf{B}_{t,l}\left(\boldsymbol{\beta}_{t,l}\right)\mathbf{z}_{t,l} \right\|^2 \right\rangle_{\mathbf{x}_{t,l},\mathbf{z}_{t,l}} + b$.

5.4.4.3 Update for $\mathbf{x}_t$

For given $q(\boldsymbol{\nu}_t)$, $q(\mathbf{z}_t)$ and $q(\boldsymbol{\gamma}_t)$, $q\left(\mathbf{x}_t\right)$ can be derived as

$$q(\mathbf{x}_t) = \prod_{l=1}^{L} \mathcal{CN}\left(\mathbf{x}_{t,l}; \boldsymbol{\mu}_{t,l}^{x}, \boldsymbol{\Sigma}_{t,l}^{x}\right), \tag{5.4.17}$$

where

$$\boldsymbol{\Sigma}_{t,l}^{x} = \left(\langle \nu_{t,l} \rangle \, \mathbf{A}_{t,l} \left(\boldsymbol{\kappa}_t \right)^H \mathbf{A}_{t,l} \left(\boldsymbol{\kappa}_t \right) + \operatorname{diag} \langle \boldsymbol{\gamma}_{t,l} \rangle \right)^{-1},$$
$$\boldsymbol{\mu}_{t,l}^{x} = \langle \nu_{t,l} \rangle \, \boldsymbol{\Sigma}_{t,l}^{x} \mathbf{A}_{t,l} \left(\boldsymbol{\kappa}_t \right)^H \left(\mathbf{y}_{t,l} - \mathbf{B}_{t,l} \left(\boldsymbol{\beta}_{t,l} \right) \langle \mathbf{z}_{t,l} \rangle \right).$$

5.4.4.4 Update for $\mathbf{z}_t$

For given $q(\boldsymbol{\nu}_t)$, $q(\mathbf{x}_t)$ and $q(\boldsymbol{\lambda}_t)$, $q(\mathbf{z}_t)$ can be derived as

$$q(\mathbf{z}_t) = \prod_{l=1}^{L} \mathcal{CN} \left(\mathbf{z}_{t,l}; \boldsymbol{\mu}_{t,l}^{z}, \boldsymbol{\Sigma}_{t,l}^{z} \right), \tag{5.4.18}$$

where

$$\boldsymbol{\Sigma}_{t,l}^{z} = \left(\langle \nu_{t,l} \rangle \, \mathbf{B}_{t,l} \left(\boldsymbol{\beta}_{t,l} \right)^H \mathbf{B}_{t,l} \left(\boldsymbol{\beta}_{t,l} \right) + \operatorname{diag} \langle \boldsymbol{\lambda}_{t,l} \rangle \right)^{-1},$$
$$\boldsymbol{\mu}_{t,l}^{z} = \langle \nu_{t,l} \rangle \, \boldsymbol{\Sigma}_{t,l}^{z} \mathbf{B}_{t,l} \left(\boldsymbol{\beta}_{t,l} \right)^H \left(\mathbf{y}_{t,l} - \mathbf{A}_{t,l} \left(\boldsymbol{\kappa}_t \right) \langle \mathbf{x}_{t,l} \rangle \right).$$

5.4.4.5 Update for $\boldsymbol{\gamma}_t$ and $\boldsymbol{\lambda}_t$

For given $q(s_t)$ and $q(\mathbf{x}_t)$, $q(\boldsymbol{\gamma}_t)$ can be derived as

$$q\left(\boldsymbol{\gamma}_t \right) = \prod_{l=1}^{L} \prod_{q=1}^{Q} \Gamma \left(\gamma_{t,l,q}; \tilde{a}_{t,l,q}, \tilde{b}_{t,l,q} \right), \tag{5.4.19}$$

where $\tilde{a}_{t,l,q} = \langle 1 \left(s_t = q \right) \rangle \, a_l + \langle 1 \left(s_t \neq q \right) \rangle \, \bar{a} + 1$, $\tilde{b}_{t,l,q} = \langle 1 \left(s_t = q \right) \rangle \, b_l + \langle 1 \left(s_t \neq q \right) \rangle \, \bar{b} + \left\langle \left| x_{t,l,q} \right|^2 \right\rangle$.

For given $q(\mathbf{v}_t)$ and $q(\mathbf{z}_t)$, $q(\boldsymbol{\lambda}_t)$ can be derived as

$$q\left(\boldsymbol{\lambda}_t \right) = \prod_{l=1}^{L} \prod_{m=1}^{M_l} \Gamma \left(\lambda_{t,l,m}; \hat{a}_{t,l,m}, \hat{b}_{t,l,m} \right), \tag{5.4.20}$$

where $\hat{a}_{t,l,m} = \langle v_{t,l,m} \rangle \, a_l^N + \langle 1 - v_{t,l,m} \rangle \, \bar{a} + 1$, $\hat{b}_{t,l,m} = \langle v_{t,l,m} \rangle \, b_l^N + \langle 1 - v_{t,l,m} \rangle \, \bar{b} + \left\langle \left| z_{t,l,m} \right|^2 \right\rangle$.

5.4.4.6 Update for s_t and $\mathbf{v}_t$

For given $q(\boldsymbol{\gamma}_t)$, $q(s_t)$ can be derived as

$$q\left(s_t \right) = \sum_{q=1}^{Q} \tilde{\pi}_{t,q} \delta \left(s_t - q \right). \tag{5.4.21}$$

$\tilde{\pi}_{t,q}$ is given by

$$\tilde{\pi}_{t,q} = \frac{1}{C_1}\pi_{t,q}e^{\sum_{l=1}^{L}\chi_{t,l,q}}, \tag{5.4.22}$$

where $\chi_{t,l,q} = (a_l - 1)\langle \ln \gamma_{t,l,q}\rangle - b_l \langle \gamma_{t,l,q}\rangle + (\bar{a} - 1)\sum_{q' \neq q}\langle \ln \gamma_{t,l,q'}\rangle - \bar{b}\sum_{q' \neq q}\langle \gamma_{t,l,q'}\rangle$, C_1 is the normalization constant to make $\sum_{q=1}^{Q}\tilde{\pi}_{t,q} = 1$, given by $C_1 = \sum_{q=1}^{Q}\pi_{t,q}e^{\sum_{l=1}^{L}\chi_{t,l,q}}$.

For given $q(\boldsymbol{\lambda}_t)$, $q(\mathbf{v}_t)$ can be derived as

$$q\left(\mathbf{v}_t\right) = \prod_{l,m}\tilde{\zeta}_{t,l,m}\delta\left(v_{t,l,m} - 1\right) + \left(1 - \tilde{\zeta}_{t,l,m}\right)\delta\left(v_{t,l,m}\right), \tag{5.4.23}$$

where

$$\tilde{\zeta}_{t,l,m} = \frac{1}{C_2}\zeta_{t,l,m}\frac{\left(b_l^N\right)^{a_l^N}}{\Gamma\left(a_l^N\right)}e^{\left(a_l^N - 1\right)\left\langle \ln \lambda_{t,l,m}\right\rangle - b_l^N\left\langle \lambda_{t,l,m}\right\rangle}, \tag{5.4.24}$$

and C_2 is the normalization constant, given by $C_2 = \zeta_{t,l,m}\frac{\left(b_l^N\right)^{a_l^N}}{\Gamma\left(a_l^N\right)}e^{\left(a_l^N - 1\right)\left\langle \ln \lambda_{t,l,m}\right\rangle - b_l^N\left\langle \lambda_{t,l,m}\right\rangle} + \left(1 - \zeta_{t,l,m}\right)\frac{\left(\bar{b}\right)^{\bar{a}}}{\Gamma(\bar{a})}e^{(\bar{a}-1)\left\langle \ln \lambda_{t,l,m}\right\rangle - \bar{b}\left\langle \lambda_{t,l,m}\right\rangle}$.

The involved expectations are given as follows for $\forall q, l, m$:

$$\begin{aligned}
&\left\langle \left\|\mathbf{y}_{t,l} - \mathbf{A}_{t,l}\left(\boldsymbol{\kappa}_t\right)\mathbf{x}_{t,l} - \mathbf{B}_{t,l}\left(\boldsymbol{\beta}_{t,l}\right)\mathbf{z}_{t,l}\right\|^2\right\rangle_{\mathbf{x}_{t,l},\mathbf{z}_{t,l}} \\
&= \left\|\mathbf{y}_{t,l} - \mathbf{A}_{t,l}\left(\boldsymbol{\kappa}_t\right)\boldsymbol{\mu}_{t,l}^x - \mathbf{B}_{t,l}\left(\boldsymbol{\beta}_{t,l}\right)\boldsymbol{\mu}_{t,l}^z\right\|^2 \\
&+ \operatorname{tr}\left(\mathbf{A}_{t,l}\left(\boldsymbol{\kappa}_t\right)\boldsymbol{\Sigma}_{t,l}^x\mathbf{A}_{t,l}\left(\boldsymbol{\kappa}_t\right)^H\right) + \operatorname{tr}\left(\mathbf{B}_{t,l}\left(\boldsymbol{\beta}_{t,l}\right)\boldsymbol{\Sigma}_{t,l}^z\mathbf{B}_{t,l}\left(\boldsymbol{\beta}_{t,l}\right)^H\right),
\end{aligned} \tag{5.4.25}$$

$$\left\langle \nu_{t,l}\right\rangle = \frac{c_{t,l}}{d_{t,l}}, \left\langle \gamma_{t,l,q}\right\rangle = \frac{\tilde{a}_{t,l,q}}{\tilde{b}_{t,l,q}}, \left\langle \lambda_{t,l,m}\right\rangle = \frac{\hat{a}_{t,l,m}}{\hat{b}_{t,l,m}}, \tag{5.4.26}$$

$$\left\langle 1\left(s_t = q\right)\right\rangle = \tilde{\pi}_{t,q}, \left\langle 1\left(s_t \neq q\right)\right\rangle = 1 - \tilde{\pi}_{t,q}, \tag{5.4.27}$$

$$\left\langle v_{t,l,m}\right\rangle = \tilde{\zeta}_{t,l,m}, \left\langle 1 - v_{t,l,m}\right\rangle = 1 - \tilde{\zeta}_{t,l,m}, \tag{5.4.28}$$

$$\left\langle \left|x_{t,l,q}\right|^2\right\rangle = \left|\mu_{t,l,q}^x\right|^2 + \boldsymbol{\Sigma}_{t,l,q,q}^x, \left\langle \mathbf{x}_{t,l}\right\rangle = \boldsymbol{\mu}_{t,l}^x, \tag{5.4.29}$$

$$\left\langle \left|z_{t,l,m}\right|^2\right\rangle = \left|\mu_{t,l,m}^z\right|^2 + \boldsymbol{\Sigma}_{t,l,m,m}^z, \left\langle \mathbf{z}_{t,l}\right\rangle = \boldsymbol{\mu}_{t,l}^z, \tag{5.4.30}$$

$$\left\langle \ln \gamma_{t,l,q}\right\rangle = \psi\left(\tilde{a}_{t,l,q}\right) - \ln\left(\tilde{b}_{t,l,q}\right), \left\langle \ln \lambda_{t,l,m}\right\rangle = \psi\left(\hat{a}_{t,l,m}\right) - \ln\left(\hat{b}_{t,l,m}\right). \tag{5.4.31}$$

where $\psi\left(x\right) = \frac{d}{dx}\ln\left(\Gamma\left(x\right)\right)$ is the digamma function, $\mu_{t,l,q}^x$ ($\mu_{t,l,m}^z$) is the q-th (m-th) element of $\boldsymbol{\mu}_{t,l}^x$ ($\boldsymbol{\mu}_{t,l}^z$), and $\boldsymbol{\Sigma}_{t,l,q,q}^x$ ($\boldsymbol{\Sigma}_{t,l,m,m}^z$) is the q-th (m-th) diagonal element of $\boldsymbol{\Sigma}_{t,l}^x$ ($\boldsymbol{\Sigma}_{t,l}^z$).

Algorithm 5.1 D-VBI for user location tracking

Input: $\{\mathbf{y}_1, \cdots, \mathbf{y}_T\}$, $\mathbf{A}_t(\mathbf{0})$, $\mathbf{B}_t(\mathbf{0}), \forall t$, and $\{a_l, b_l, a_l^N, b_l^N\}$,$\bar{a}, \bar{b}$,$a, b$.
Output: $\{\hat{\mathbf{p}}_1, \cdots, \hat{\mathbf{p}}_T\}$.
for $t = 1, \cdots, T$ **do**
  Update $\hat{p}(s_t)$ and $\hat{p}(\mathbf{v}_t)$ using (5.4.14)-(5.4.15).
  Initialize: $i = 0$, $\boldsymbol{\kappa}_t^{(0)} = \mathbf{0}$, $\boldsymbol{\beta}_t^{(0)} = \mathbf{0}$, $\tilde{a}_{t,l,q}^{(0)} = \pi_{t,q} a_l + (1 - \pi_{t,q})\bar{a}$, $\tilde{b}_{t,l,q}^{(0)} = \pi_{t,q} b_l + (1 - \pi_{t,q})\bar{b}$, $\hat{a}_{t,l,m}^{(0)} = \zeta_{t,l,m} a_l^N + (1 - \zeta_{t,l,m})\bar{a}$, $\hat{b}_{t,l,m}^{(0)} = \zeta_{t,l,m} b_l^N + (1 - \zeta_{t,l,m})\bar{b}$, $\tilde{\pi}_{t,q}^{(0)} = \pi_{t,q}$, $\tilde{\zeta}_{t,l,m}^{(0)} = \zeta_{t,l,m}$, $\widetilde{\boldsymbol{\Sigma}}_{t,l} = \left(\widetilde{\mathbf{A}}_{t,l}^H \widetilde{\mathbf{A}}_{t,l} + \mathrm{diag}\,\langle \Lambda_{t,l} \rangle^{(0)}\right)^{-1}$, $\left[\boldsymbol{\Sigma}_{t,l}^{x(0)}, \boldsymbol{\Sigma}_{t,l}^{z(0)}\right] = \mathrm{diagblock}(\widetilde{\boldsymbol{\Sigma}}_{t,l})$, $\left[\boldsymbol{\mu}_{t,l}^{x(0)}; \boldsymbol{\mu}_{t,l}^{z(0)}\right] = \widetilde{\boldsymbol{\Sigma}}_{t,l} \widetilde{\mathbf{A}}_{t,l}^H \mathbf{y}_{t,l}$, in which $\widetilde{\mathbf{A}}_{t,l} = [\mathbf{A}_{t,l}(\mathbf{0}), \mathbf{B}_{t,l}(\mathbf{0})]$, $\Lambda_{t,l} = [\gamma_{t,l}; \boldsymbol{\lambda}_{t,l}], \forall l, q, m$.
  Calculate the expectations about $\mathbf{x}_t$, $\mathbf{z}_t$, s_t, $\mathbf{v}_t$, $\boldsymbol{\gamma}_t$, $\boldsymbol{\lambda}_t$ using (5.4.25)-(5.4.31).
  while not converge **do**
    %Approximate Posterior Distributions:
    Update $q^{(i+1)}(\boldsymbol{\nu}_t)$ using (5.4.16) and the related expectation using (5.4.26).
    Update $q^{(i+1)}(\mathbf{x}_t)$ using (5.4.17) and the related expectations using (5.4.29).
    Update $q^{(i+1)}(\mathbf{z}_t)$ using (5.4.18) and the related expectations using (5.4.25) and (5.4.30).
    Update $q^{(i+1)}(\boldsymbol{\gamma}_t)$, $q^{(i+1)}(\boldsymbol{\lambda}_t)$ using (5.4.19)-(5.4.20) and the related expectations using (5.4.26) and (5.4.31).
    Update $q^{(i+1)}(s_t)$, $q^{(i+1)}(\mathbf{v}_t)$ using (5.4.21)-(5.4.24) and the related expectations using (5.4.27) and (5.4.28).
    %Off-grid Parameters Update:
    Update $\boldsymbol{\kappa}_t^{(i+1)}$ and $\boldsymbol{\beta}_t^{(i+1)}$ according to (5.4.12) and (5.4.13).
    $i = i + 1$.
  end while
  Set $q^*(s_t) = q^{(i)}(s_t)$, $q^*(\mathbf{v}_t) = q^i(\mathbf{v}_t)$, $\boldsymbol{\kappa}_t^* = \boldsymbol{\kappa}_t^{(i)}$. Pass $q^*(s_t)$ and $q^*(\mathbf{v}_t)$ to $t + 1$.
  Estimate $\hat{s}_t$ according to $\hat{s}_t = \arg\max_{s_t} q^*(s_t)$, and $\hat{\boldsymbol{\kappa}}_t = \boldsymbol{\kappa}_t^*$, then the position estimation $\hat{\mathbf{p}}_t$ is given by (5.3.15).
end for

5.4.5 D-VBI Algorithm Realization

The proposed overall D-VBI algorithm can be summarized as in Algorithm 5.1 and Fig. 5.4. It is shown in [27] and [28] that the Bayesian inference algorithm is not sensitive to the prior parameters, such as $\{a_l, b_l, a_l^N, b_l^N\}$ and $\bar{a}, \bar{b}$. When we have prior information about the approximate path gain between the user and BSs, we could set b_l/a_l and b_l^N/a_l^N to be the approximate path gain. Otherwise, we could simply set $a_l = b_l = a_l^N = b_l^N = \bar{a} = 1$, $\bar{b} = 0.001$.

5.4.6 Algorithm Complexity

We discuss the computational complexity of proposed algorithm. Following the overall flow of the proposed D-VBI algorithm in Fig. 5.4, the number of mathematical operations involved at each step is summarized in the Table 5.1. We focus on the complicated mathematical operations, such as multiplications and divisions. Note that the updates of $q(\mathbf{x}_t)$ and $q(\mathbf{z}_t)$ require

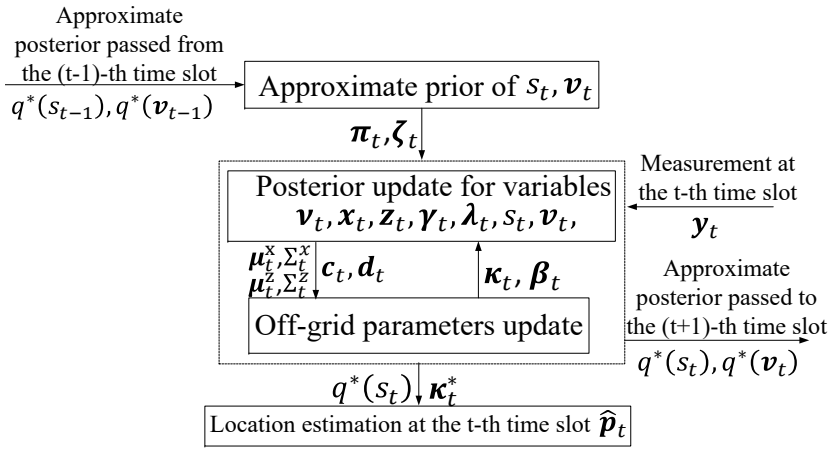

Figure 5.4: The overall flow of the proposed D-VBI algorithm

Number of mathematical operations	Multiplication	Division
$\hat{p}(s_t)$	Q^2	
$\hat{p}(\mathbf{v}_t)$	$2M$	
$q(\boldsymbol{\nu}_t)$	$\sum_l [2N_lQ + N_lQ^2 + 2N_lM_l + N_lM_l^2] + N$	
$q(\mathbf{x}_t)$	$2QL + 2NQ + 2\sum_l N_l^2Q + 2Q^2N + \sum_l N_lM_l$	$QL +$ $L+\mathcal{O}\left(\sum_l N_l^3\right)$
$q(\mathbf{z}_t)$	$2M + 2\sum_l (N_lM_l + N_l^2M_l + M_l^2N_l) + NQ$	$M + \mathcal{O}\left(\sum_l N_l^3\right)$
$q(\boldsymbol{\gamma}_t)$	$5LQ$	
$q(\boldsymbol{\lambda}_t)$	$5M$	
$q(s_t)$	$5Q$	$LQ + Q$
$q(\mathbf{v}_t)$	$8M$	$2M + L + 1$
$\boldsymbol{\kappa}_t$	$2Q + 5LQ + 5QN$	$2LQ$
$\boldsymbol{\beta}_t$	$2M + \sum_l 5N_lM_l$	

Table 5.1: Number of mathematical operations involved in the D-VBI algorithm

the matrix inversion. Assuming the arithmetic with individual elements has complexity $\mathcal{O}(1)$, the computational complexity of matrix inversion for $n \times n$ matrix is $\mathcal{O}(n^3)$. We use matrix inversion lemma to reduce the complexity of computing $\boldsymbol{\Sigma}_{t,l}^x$ and $\boldsymbol{\Sigma}_{t,l}^z$. From Table 5.1, it can be seen that the main computational burden is updating $q(\boldsymbol{\nu}_t), q(\mathbf{x}_t)$ and $q(\mathbf{z}_t)$. Because their updates require the most number of the mathematical operations (cubic order) compared to the updates of other steps.

5.5 Simulation Results

In this section, we verify the location tracking performance of the proposed algorithm in massive MIMO systems. The proposed algorithm is compared with the following baselines:

- **Baseline 1** (VBI (i.i.d.) [28]): The VBI algorithm assumes i.i.d sparse channel prior at each time slot.

- **Baseline 2** (VBI (Group-Sparse) [28]): The VBI algorithm assumes group sparse LOS channel prior and i.i.d sparse NLOS channel prior at each time slot.

- **Baseline 3** (DiSouL [90]): The LOS and NLOS channels are recovered through $\ell_{2,1}$ norm minimization at each time slot independently and the grid is refined around the estimated location and angles adaptively.

- **Baseline 4** (ML Classifier [86]): This is a fingerprinting-based algorithm, which exploits changes in statistics of the sparse beamspace channel matrix as a function of the user position. BS1 in Fig. 5.5 is chosen as the operation BS. In each grid cell, we uniformly pick $N_{sp} = 1000$ sample user positions and compute the average covariance matrix for this cell.

- **Baseline 5** (D-VBI (without off-grid)): The D-VBI algorithm proposed in this paper is used to track user's location by exploiting the TMGS prior without considering the off-grid effect, i.e., the off-grid parameters $\boldsymbol{\kappa}_t$ and $\boldsymbol{\beta}_t$ are set to be zero.

We consider the user is moving within an area $\mathcal{X}$ with size 50×50 m, which is split into $Q = 100$ grid cells, each with size 5×5 m in Section 5.5.1 and 5.5.2. Assume the origin of the coordinate system is in the middle of the area, four BSs are located at $[-50\,\mathrm{m}, 50\,\mathrm{m}]$,$[50\,\mathrm{m}, 50\,\mathrm{m}]$,$[-50\,\mathrm{m}, -50\,\mathrm{m}]$,$[-50\,\mathrm{m}, 50\,\mathrm{m}]$ respectively. We consider directional and non-directional user movements, as shown in Fig. 5.5. Two scatterers uniformly located are considered. The uniform linear array (ULA) and uniform circular array (UCA) with $\frac{\lambda}{2}$ inter antenna spacing are considered. The transmit power is $P_T = |u_t|^2$. The channel gain for LOS path is computed as $\alpha_{t,l} = 10^{-L\left(d_{t,l}\right)/20} e^{\frac{j2\pi f_c d_{t,l}}{c}}$, where $L(d_{t,l}) = 20\log_{10}\left(\frac{4\pi f_c}{c}\right) + 10n\log_{10}\left(\frac{d_{t,l}}{1m}\right)$ is the pathloss at distance $d_{t,l}$ in meters, f_c is the carrier frequency, c is the light speed, n is the pathloss exponent. The channel gain for NLOS path is computed as $\alpha^i_{t,l} = 10^{-L(d_{t,l})/20}\sqrt{X^i_{t,l}} e^{j\phi^i_{t,l}}$, where $X^i_{t,l}$ is the shadowing coefficient and the phase $\phi^i_{t,l}$ is modeled as $\phi^i_{t,l} \sim \mathcal{U}(0, 2\pi)$. We adopt the standard cellular channel parameters in 3GPP [102] as shown in Table 5.2.

We use root-mean-square error (RMSE) as a performance metric for the tracking schemes, which is defined as $\mathrm{RMSE} = \sqrt{\frac{1}{KT}\sum_{k=1}^{K}\sum_{t=1}^{T}\left\|\hat{\mathbf{p}}^k_t - \mathbf{p}_t\right\|^2}$, where $\hat{\mathbf{p}}^k_t$ denotes the position

Parameters	Value
carrier frequency f_c	7 GHz
Path loss exponent (PLE) n in $L(d_{t,l})$	$n = 2$ for LOS path, $n = 3$ for NLOS path
Standard deviation of log-norm shadowing σ_X	$\sigma_X = 6.8$ dB
Noise power $\sigma^2_{t,l}, \forall l$	-92 dBm

Table 5.2: Simulation parameters

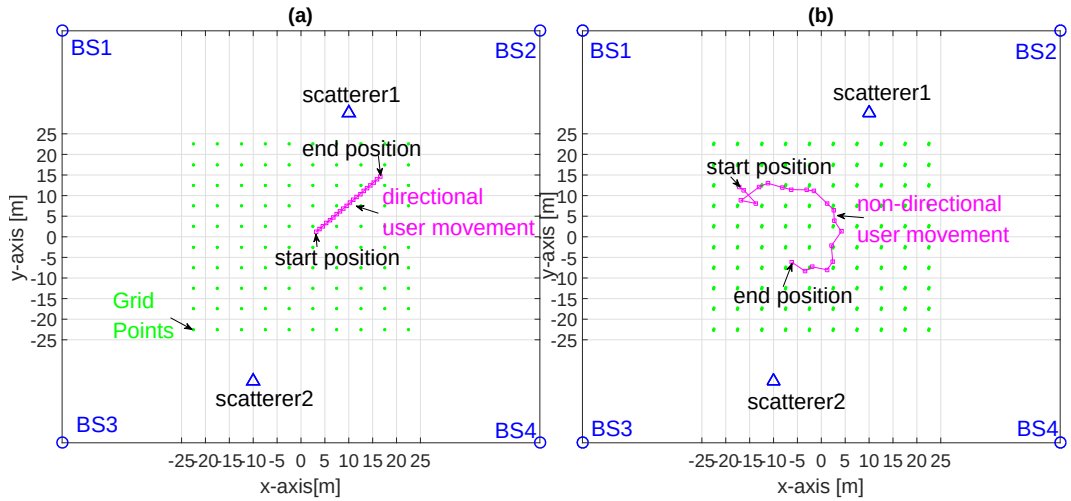

Figure 5.5: User's movement trajectory considered in the simulations. We consider $T = 20$, $L = 4$, $Q = 100$, and the grid resolution is 5×5 m. (a) the user moves in a directional manner; (b) the user moves without a directional trend.

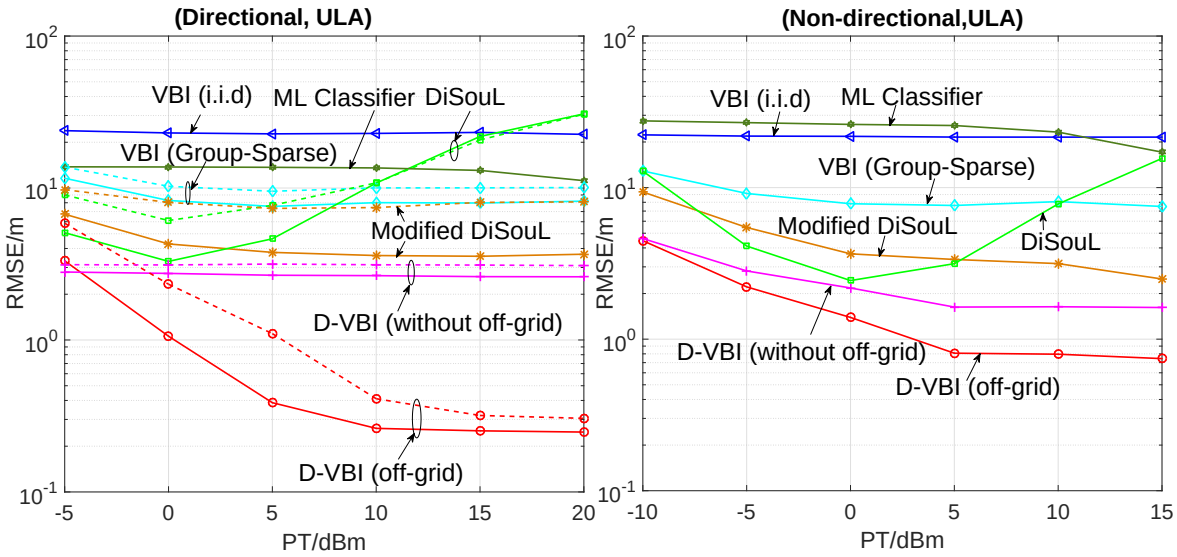

Figure 5.6: RMSE performance versus the transmit power P_T when ULA is used. Set $N_l = 32, \forall l$, $Q = 100$. Left: directional user movement; Right: non-directional user movement.

estimation at the t-th time slot in the k-th simulation run, and $K = 500, T = 20$ for each scheme. We set $M_l = N_l, \forall l$, $\rho_{01} = \rho_{10} = 0.2$. The TPM $\mathbf{G}_t$ is learned by the previous location estimates for the directional user movement and is set to have equal probability in the potential nonzero indices for the non-directional user movement. Note that the simulation results for the message-passing-based algorithms [24, 25] are not shown because these algorithms will totally diverge under the ill-conditioned sensing matrix in the location tracking problem.

5.5.1 Impact of Transmit Power and User Movement Direction

The RMSE performance of different algorithms versus the transmit power P_T is shown in Fig. 5.6 for ULA and Fig. 5.7 for UCA. Under each antenna configuration, we consider different user movement trajectories. From the simulation, we found that the RMSE performance of the original DiSouL [90] becomes worse for higher transmit powers because as P_T increases, the effective noise power (i.e., the mismatch between the measurements and the signal model) caused by the position/angle offset is enlarged, but the constraint bound ϵ (which is supposed to reflect this mismatch) in [90, (17b)] only depends on the AWGN power and is not adjusted according to the increased effective noise power. On the other hand, the VBI-based methods can automatically learn the effective noise power and thus is less sensitive to the position/angle offset. The modified DiSouL baseline is added in this subsection

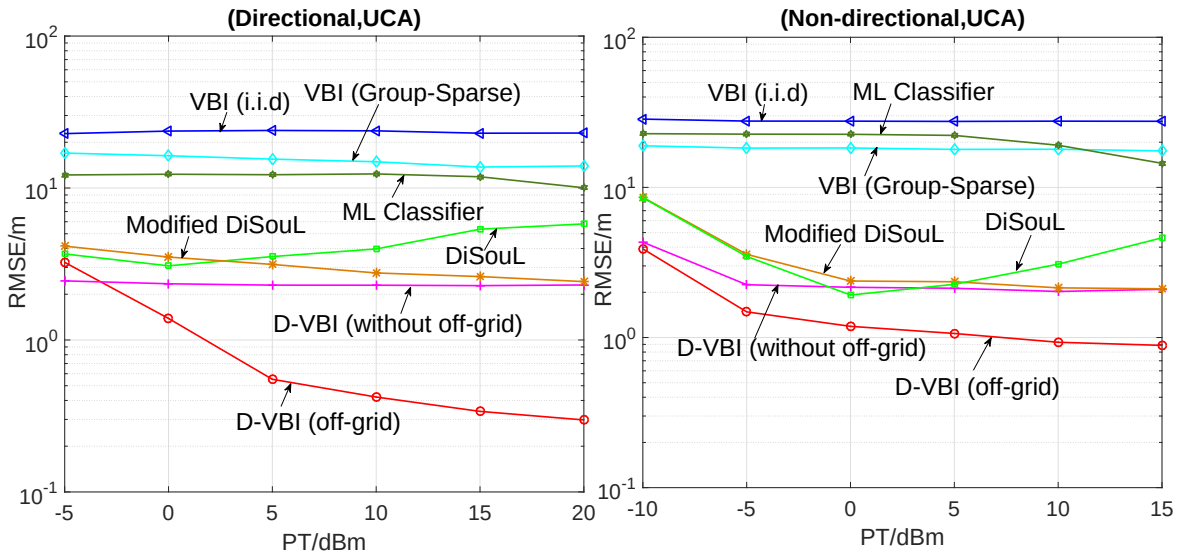

Figure 5.7: RMSE performance versus the transmit power P_T when UCA is used. Set $N_l = 32, \forall l$, $Q = 100$. Left: directional user movement; Right: non-directional user movement.

to deal with the inappropriate ϵ for higher P_T. Specifically, ϵ is set to be a larger value (calculated assuming the largest off-grid effect) at the first iteration, then is gradually decreased in the grid refinement procedure. Fig. 5.6 and 5.7 show that the modified DiSouL performs better than the original DiSouL for higher P_T.

From Fig. 5.6 and Fig. 5.7, we can see that the proposed algorithm with off-grid refinement outperforms the other baselines under different antenna arrays for both directional and non-directional user movement. In the directional case, the previous location estimates could provide more accurate prior for the transition matrix $\mathbf{G}_t$, therefore the proposed algorithm achieves better performance compared to the non-directional case. This further confirms that: Firstly, the proposed TMGS probability model captures the first-order structure of the massive MIMO channels in location tracking problem; Secondly, the proposed algorithm is not sensitive to the true distribution of the channel and can work well under the typical simulation setup.

In order to show the robustness of our algorithm to Assumption 1, in the left side of Fig. 5.6, we also simulate the case when the LOS path between user and BS3 (as shown in Fig. 5.5) is blocked during the tracking process. The simulation results are presented by the dash lines. It shows that even though BS3 cannot receive LOS path from the moving user, the proposed algorithm still can work quite well. In particular, we do not need every BS to have a LOS path to the user. We just need a few (e.g. 3 BSs with active LOS path) and the algorithm can achieve good performance. Hence, the algorithm is quite robust to the LOS blocking

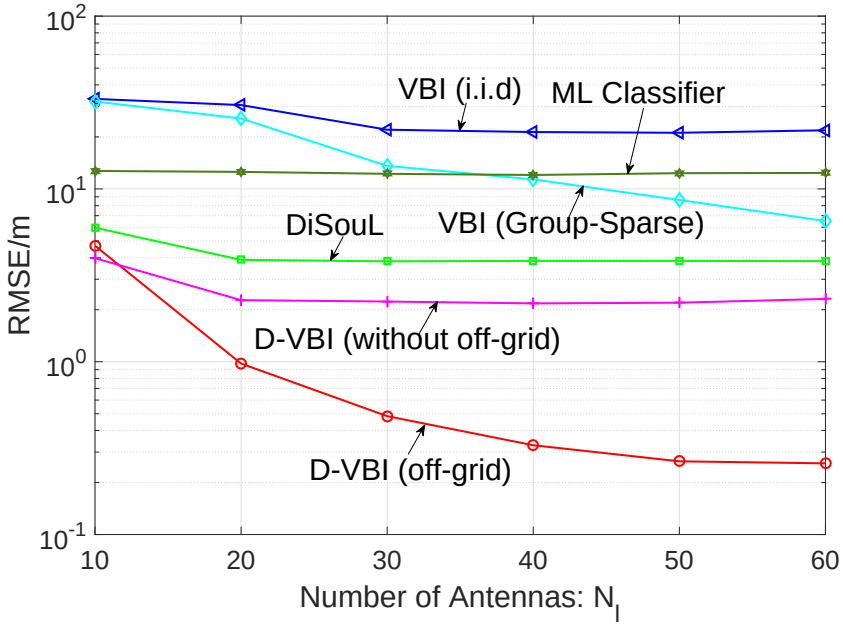

Figure 5.8: RMSE performance versus the number of antennas N_l for directional user movement. Consider equal transmit antenna numbers at all BSs, and UCA is used. Set $P_T = 8$ dBm, $Q = 100$.

scenario.

5.5.2 Impact of Antenna Numbers

Fig. 5.8 illustrates the RMSE performance of different algorithms versus the number of antennas N_l. It shows that the proposed D-VBI outperforms all other baselines for massive antenna arrays. When the antenna number is increasing, the angular resolution is improved, which can improve the localization accuracy.

5.5.3 Impact of Grid Resolution

Fig. 5.9 plots the cumulative density function (CDF) of the root-temporal-mean-square error (RTMSE), defined as $\text{RTMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\|\hat{\mathbf{p}}_t - \mathbf{p}_t\|^2}$, for different location grid resolutions. It shows that the proposed algorithm achieves higher location tracking accuracy with higher probability compared to the baselines for different grid resolutions.

5.5.4 Impact of the Number of NLOS Paths

In Fig. 5.10, we plot the RMSE performance versus the number of NLOS paths. It shows that the proposed algorithm is robust to the number of NLOS paths (i.e., the number of

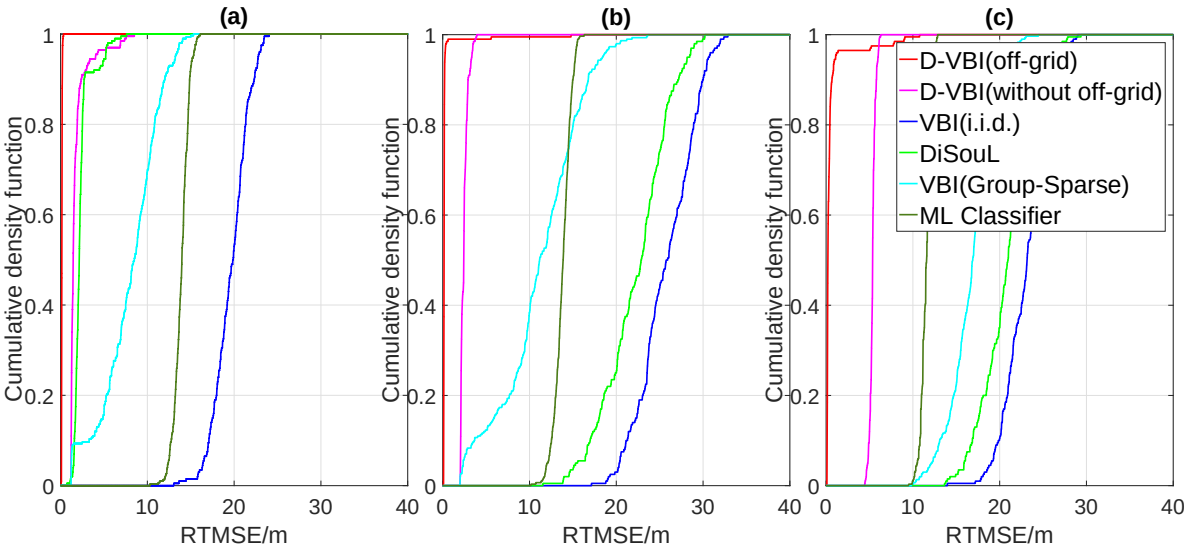

Figure 5.9: CDF of the RTMSE for different grid resolutions for directional user movement when ULA is used. Set $N_l = 32, \forall l$, $P_T = 8$ dBm. (a) Grid resolution is 3×3 m, $Q = 256$, $N_r = N_c = 16$, $\mathcal{X} = 48 \times 48$ m; (b) Grid resolution is 5×5 m, $Q = 100$, $N_r = N_c = 10$, $\mathcal{X} = 50 \times 50$ m; (c) Grid resolution is 10×10 m, $Q = 25$, $N_r = N_c = 5$, $\mathcal{X} = 50 \times 50$ m.

scatterers in the environment), and the proposed algorithm achieves significant performance gains compared to the baselines even when there are more NLOS paths.

5.6 Summary

We propose a novel user location tracking algorithm in massive MIMO systems. To capture the PTC of massive MIMO channels across time and the GS resulting from the cooperative localization, we propose a TMGS model. In order to exploit the TMGS prior and deal with the ill-conditioned measurement matrix in location tracking problem, we propose a D-VBI algorithm. Specifically, we first decompose the joint optimization problem into subproblems which only involve the optimization variables at the current time slot and the posteriors from the last time slot. Then, we obtain an approximate subproblem by substituting the exact posterior with its approximation passed from the last time slot. Next, an inexact BCD is proposed to find a stationary solution of the approximate subproblem. At last, we obtained the MAP estimation of the user's location. The simulations show the superior location tracking performance of the proposed algorithm.

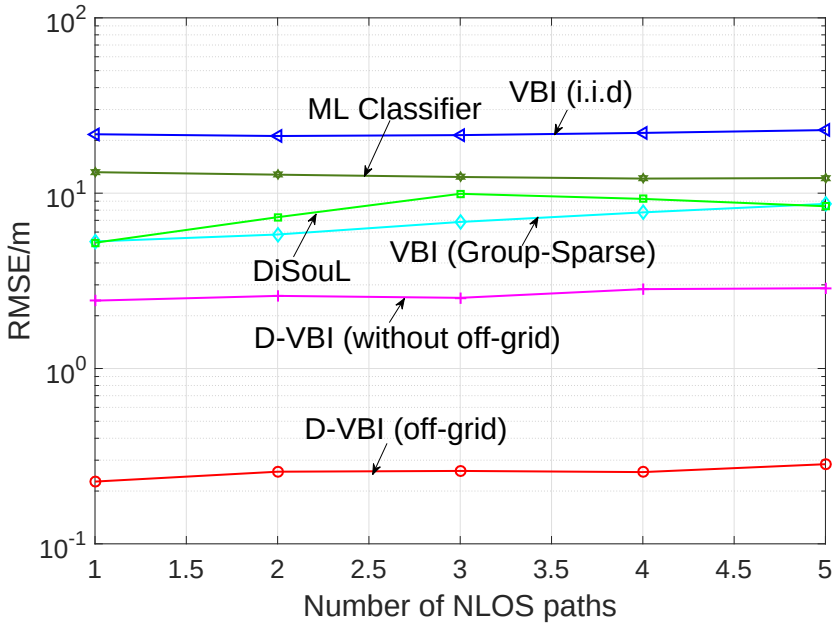

Figure 5.10: RMSE performance versus the number of NLOS paths for directional user movement when ULA is used. Set $N_l = 32, \forall l, Q = 100, P_T = 8$dBm.

5.7 Appendix

5.7.1 Proof of Lemma 5.1

We first illustrate how to calculate $p\left(\boldsymbol{h}_t, \mathbf{y}_{1:t}; \boldsymbol{\kappa}^{\star}_{1:t-1}, \boldsymbol{\beta}^{\star}_{1:t-1}, \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right)$ in the problem (5.4.1), then we will prove the statement.

Because $p\left(\boldsymbol{h}_t, \mathbf{y}_{1:t}; \boldsymbol{\kappa}^{\star}_{1:t-1}, \boldsymbol{\beta}^{\star}_{1:t-1}, \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right) \propto p\left(\mathbf{y}_t \left| \boldsymbol{h}_t; \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right.\right) p\left(\boldsymbol{h}_t \left| \mathbf{y}_{1:t-1}; \boldsymbol{\kappa}^{\star}_{1:t-1}, \boldsymbol{\beta}^{\star}_{1:t-1}\right.\right)$. The first term is given by (5.3.10). The second term can be calculated as

$$p\left(\boldsymbol{h}_t \left| \mathbf{y}_{1:t-1}; \boldsymbol{\kappa}^{\star}_{1:t-1}, \boldsymbol{\beta}^{\star}_{1:t-1}\right.\right) = \int p\left(\boldsymbol{h}_t \left| \boldsymbol{h}_{t-1}\right.\right) p\left(\boldsymbol{h}_{t-1} \left| \mathbf{y}_{1:t-1}; \boldsymbol{\kappa}^{\star}_{1:t-1}, \boldsymbol{\beta}^{\star}_{1:t-1}\right.\right) d\boldsymbol{h}_{t-1},$$

in which $p\left(\boldsymbol{h}_t \left| \boldsymbol{h}_{t-1}\right.\right) = p\left(\boldsymbol{\nu}_t, \mathbf{x}_t, \boldsymbol{\gamma}_t, \mathbf{z}_t, \boldsymbol{\lambda}_t \left| s_t, \mathbf{v}_t\right.\right) p\left(s_t \left| s_{t-1}\right.\right) p\left(\mathbf{v}_t \left| \mathbf{v}_{t-1}\right.\right)$. Then we have

$$\begin{aligned} &p\left(\boldsymbol{h}_t \left| \mathbf{y}_{1:t-1}; \boldsymbol{\kappa}^{\star}_{1:t-1}, \boldsymbol{\beta}^{\star}_{1:t-1}\right.\right) \\ &= p\left(\boldsymbol{\nu}_t, \mathbf{x}_t, \boldsymbol{\gamma}_t, \mathbf{z}_t, \boldsymbol{\lambda}_t \left| s_t, \mathbf{v}_t\right.\right) \sum_{s_{t-1}, \mathbf{v}_{t-1}} p\left(s_t \left| s_{t-1}\right.\right) p\left(\mathbf{v}_t \left| \mathbf{v}_{t-1}\right.\right) p\left(s_{t-1}, \mathbf{v}_{t-1} \left| \mathbf{y}_{1:t-1}; \boldsymbol{\kappa}^{\star}_{1:t-1}, \boldsymbol{\beta}^{\star}_{1:t-1}\right.\right). \end{aligned}$$

Then we can get the final expression of $p\left(\boldsymbol{h}_t, \mathbf{y}_{1:t}; \boldsymbol{\kappa}^{\star}_{1:t-1}, \boldsymbol{\beta}^{\star}_{1:t-1}, \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right)$, as given by (5.4.2).

The optimization problem (5.4.1) can be calculated as follows:

$$\begin{aligned}
&\int q\left(\boldsymbol{h}_t\right) \ln\left(\frac{p\left(\boldsymbol{h}_t, \mathbf{y}_{1:t}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}, \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right)}{q\left(\boldsymbol{h}_t\right)}\right) d\boldsymbol{h}_t \\
\leq & \ln \int q\left(\boldsymbol{h}_t\right) \frac{p\left(\boldsymbol{h}_t, \mathbf{y}_{1:t}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}, \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right)}{q\left(\boldsymbol{h}_t\right)} d\boldsymbol{h}_t \\
= & \ln p\left(\mathbf{y}_{1:t}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}, \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right),
\end{aligned} \tag{5.7.1}$$

where Jensen's inequality is applied to (5.7.1). Therefore, (5.4.1) is maximized w.r.t. $q\left(\boldsymbol{h}_t\right)$ when $q\left(\boldsymbol{h}_t\right)$ has the following form:

$$q^{\star}\left(\boldsymbol{h}_t\right) = p\left(\boldsymbol{h}_t \left| \mathbf{y}_{1:t}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}, \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right.\right), \tag{5.7.2}$$

and the optimization problem (5.4.1) is reduced to

$$\max_{\boldsymbol{\kappa}_t, \boldsymbol{\beta}_t} \ln p\left(\mathbf{y}_{1:t}; \boldsymbol{\kappa}_{1:t-1}^{\star}, \boldsymbol{\beta}_{1:t-1}^{\star}, \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t\right). \tag{5.7.3}$$

Therefore, the optimal solutions of (5.7.3), i.e., $\boldsymbol{\kappa}_t^{\star}$, $\boldsymbol{\beta}_t^{\star}$ are the optimal solutions of the original problem (5.3.12).

5.7.2 Gradient Update for Off-grid Parameters

The derivative $\boldsymbol{\xi}_{\kappa_t}$ can be calculated as $\boldsymbol{\xi}_{\kappa_t} = \left[\xi\left(\boldsymbol{\kappa}_{t,1}\right); \cdots; \xi\left(\boldsymbol{\kappa}_{t,Q}\right)\right]$, with

$$\xi\left(\boldsymbol{\kappa}_{t,q}\right) = \begin{bmatrix} \sum_{l=1}^{L} 2\mathrm{Re}\left(\widetilde{\mathbf{a}}_{t,l,q}^{\prime H}\left(\widetilde{\mathbf{a}}_{t,l,q} c_1 + \mathbf{c}_2\right)\right) c_3 \\ \sum_{l=1}^{L} 2\mathrm{Re}\left(\widetilde{\mathbf{a}}_{t,l,q}^{\prime H}\left(\widetilde{\mathbf{a}}_{t,l,q} c_1 + \mathbf{c}_2\right)\right) c_4 \end{bmatrix},$$

where $c_0 = \left\langle \nu_{t,l} \right\rangle$, $c_1 = -c_0\left(\left|\mu_{t,l,q}^{x}\right|^2 + \boldsymbol{\Sigma}_{t,l,q,q}^{x}\right)$, $\mathbf{c}_2 = -c_0\left(\sum_{q' \neq q} \boldsymbol{\Sigma}_{t,l,q',q}^{x} \widetilde{\mathbf{a}}_{t,l,q'} - \left(\mu_{t,l,q}^{x}\right)^{*} \mathbf{y}_{t,l,-q}\right)$,
$c_3 = -\left(\phi_q^y + \kappa_{t,q}^y - \tilde{p}_l^y\right) / \left\|\boldsymbol{\phi}_q + \boldsymbol{\kappa}_{t,q} - \tilde{\mathbf{p}}_l\right\|^2$, $c_4 = \left(\phi_q^x + \kappa_{t,q}^x - \tilde{p}_l^x\right) / \left\|\boldsymbol{\phi}_q + \boldsymbol{\kappa}_{t,q} - \tilde{\mathbf{p}}_l\right\|^2$,
$\mathbf{y}_{t,l,-q} = \mathbf{y}_{t,l} - \mathbf{B}_{t,l}\left(\boldsymbol{\beta}_{t,l}\right) \boldsymbol{\mu}_{t,l}^{z} - \sum_{q' \neq q} \widetilde{\mathbf{a}}_{t,l,q'} \mu_{t,l,q'}^{x}$, $\widetilde{\mathbf{a}}_{t,l,q} = \mathbf{a}_l\left(\theta_{t,l}\left(\boldsymbol{\phi}_q + \boldsymbol{\kappa}_{t,q}\right)\right)$ and
$\widetilde{\mathbf{a}}_{t,l,q}^{\prime} = d\mathbf{a}_l\left(\theta_{t,l}\left(\boldsymbol{\phi}_q + \boldsymbol{\kappa}_{t,q}\right)\right) / d\theta_{t,l}\left(\boldsymbol{\phi}_q + \boldsymbol{\kappa}_{t,q}\right)$.

The derivative $\boldsymbol{\xi}_{\beta_{t,l}}$ can be calculated as $\boldsymbol{\xi}_{\beta_{t,l}} = \left[\xi\left(\beta_{t,l,1}\right), \cdots, \xi\left(\beta_{t,l,M_l}\right)\right]^T$, with

$$\xi\left(\beta_{t,l,m}\right) = 2\mathrm{Re}\left(\bar{\mathbf{a}}_{t,l,m}^{\prime H}\left(\bar{\mathbf{a}}_{t,l,m} c_5 + \mathbf{c}_6\right)\right),$$

where $c_5 = -c_0\left(\left|\mu^z_{t,l,m}\right|^2 + \mathbf{\Sigma}^z_{t,l,m,m}\right)$, $\mathbf{c}_6 = -c_0\sum_{m'\neq m}\mathbf{\Sigma}^z_{t,l,m',m}\bar{\mathbf{a}}_{t,l,m'} + c_0\left(\mu^z_{t,l,m}\right)^*\mathbf{y}_{t,l,-m}$, $\mathbf{y}_{t,l,-m} = \mathbf{y}_{t,l} - \mathbf{A}_{t,l}\left(\boldsymbol{\kappa}_t\right)\boldsymbol{\mu}^x_{t,l} - \sum_{m'\neq m}\bar{\mathbf{a}}_{t,l,m'}\mu^z_{t,l,m'}$, $\bar{\mathbf{a}}_{t,l,m} = \mathbf{a}_l\left(\vartheta_m + \beta_{t,l,m}\right)$ and $\bar{\mathbf{a}}'_{t,l,m} = d\mathbf{a}_l\left(\vartheta_m + \beta_{t,l,m}\right)/d\beta_{t,l,m}$.

5.7.3 Proof of Lemma 5.3

For conciseness, we omit the time index t. The non-decreasing property can be achieved by

$$\begin{aligned}
\mathcal{U}\left(q^{(i)}_{1:|\mathcal{H}|}, \boldsymbol{\kappa}^{(i)}, \boldsymbol{\beta}^{(i)}\right) &\leq \mathcal{U}\left(q^{(i+1)}_{1}, q^{(i)}_{2:|\mathcal{H}|}, \boldsymbol{\kappa}^{(i)}, \boldsymbol{\beta}^{(i)}\right) \\
&\leq \cdots \\
&\leq \mathcal{U}\left(q^{(i+1)}_{1:|\mathcal{H}|}, \boldsymbol{\kappa}^{(i)}, \boldsymbol{\beta}^{(i)}\right) \\
&\leq \mathcal{U}_t\left(q^{(i+1)}_{1:|\mathcal{H}|}, \boldsymbol{\kappa}^{(i+1)}, \boldsymbol{\beta}^{(i)}\right) \\
&\leq \mathcal{U}\left(q^{(i+1)}_{1:|\mathcal{H}|}, \boldsymbol{\kappa}^{(i+1)}, \boldsymbol{\beta}^{(i+1)}\right).
\end{aligned}$$

The objective function $\mathcal{U}$ is upper bounded by 1, therefore the non-decreasing sequences $\mathcal{U}\left(q^{(i)}_{1:|\mathcal{H}|}, \boldsymbol{\kappa}^{(i)}, \boldsymbol{\beta}^{(i)}\right)$ converge to a limit.

From Section 5.4.4, the updates of $q\left(\boldsymbol{h}\right)$ can be considered as some parameterized functions. For example, a Gamma distribution with parameters $\left\{\tilde{a}_{l,q}, \tilde{b}_{l,q}\right\}$ for $q\left(\boldsymbol{\gamma}\right)$,or a Gaussian distribution with parameters $\{\boldsymbol{\mu}^x_l, \mathbf{\Sigma}^x_l\}$ for $q\left(\mathbf{x}\right)$. Therefore, the optimization problem (5.4.5), which is optimized over the functional space for $q\left(\boldsymbol{h}\right)$, can be considered as a conventional parameter optimization problem. Denote all the updated parameters related to $q\left(\boldsymbol{h}\right)$ and $\{\boldsymbol{\kappa}, \boldsymbol{\beta}\}$ as $\boldsymbol{r}$, which includes $H = |\mathcal{H}| + 2$ blocks of parameters. The first $|\mathcal{H}|$ blocks are updated alternatively by

$$\boldsymbol{r}^i_n = \arg\max_{\boldsymbol{r}_n} u_n(\boldsymbol{r}_n, \boldsymbol{r}^{i-1}), n = 1, \cdots, |\mathcal{H}|, \tag{5.7.4}$$

where $u_n(\boldsymbol{r}_n, \boldsymbol{r}^{i-1}) \triangleq \mathcal{U}(\boldsymbol{r}_n, \boldsymbol{r}^{i-1}_{-n})$, and (5.7.4) has unique solution for any point $\boldsymbol{r}^{i-1}$, as shown by Lemma 5.2. For the last two blocks, the gradient update in (5.4.12) and (5.4.13) are equivalent to solving the convex approximation of the original function for the n-th block, which is

$$\begin{aligned}
\boldsymbol{y}^i_n &= \arg\max_{\boldsymbol{r}_n} u_n(\boldsymbol{r}_n, \boldsymbol{r}^{i-1}), n = |\mathcal{H}| + 1, |\mathcal{H}| + 2 \\
\boldsymbol{r}^i_n &= \boldsymbol{r}^{i-1}_n + \Delta^i_n \cdot \boldsymbol{d}^{i-1}_n,
\end{aligned} \tag{5.7.5}$$

where $\boldsymbol{d}_n^{i-1} = \boldsymbol{y}_n^i - \boldsymbol{r}_n^{i-1}$, and $u_n(\boldsymbol{r}_n, \boldsymbol{r}^{i-1}) \triangleq \mathcal{U}(\boldsymbol{r}^{i-1}) + \langle \mathcal{U}_n'\left(\boldsymbol{r}^{i-1}\right), \boldsymbol{r}_n - \boldsymbol{r}_n^{i-1}\rangle - \frac{1}{2}\left\|\boldsymbol{r}_n - \boldsymbol{r}_n^{i-1}\right\|^2$, where $\mathcal{U}_n'\left(\boldsymbol{r}^{i-1}\right)$ is the block partial gradient of $\mathcal{U}$ at $\boldsymbol{r}_n^{i-1}$, and Δ_n^i is the stepsize determined by the Armijo rule. It can be seen that

$$\begin{aligned} u_n(\boldsymbol{r}_n^{i-1}, \boldsymbol{r}^{i-1}) &= \mathcal{U}(\boldsymbol{r}^{i-1}), \\ u_n'(\boldsymbol{r}_n, \boldsymbol{r}^{i-1})\Big|_{\boldsymbol{r}_n = \boldsymbol{r}_n^{i-1}} &= \mathcal{U}_n'\left(\boldsymbol{r}^{i-1}\right). \end{aligned} \tag{5.7.6}$$

Consider a limit point $\boldsymbol{w}$ and a sub-sequence $\{\boldsymbol{r}^{i_j}\}_j$ converging to $\boldsymbol{w}$. For the first $|\mathcal{H}|$ blocks with a unique solution in (5.7.4), following the same proof for Theorem 2-b in [104], we can get

$$u_n(\boldsymbol{w}_n, \boldsymbol{w}) \geq u_n(\boldsymbol{r}_n, \boldsymbol{w}), \forall \boldsymbol{r}_n, n = 1, \cdots, |\mathcal{H}|. \tag{5.7.7}$$

For the last two blocks updated by (5.7.5), due to the use of the Armijo step size, following similar contradiction proof as (A.27)-(A.29) in [104], we have $\lim_{j\to\infty} \boldsymbol{d}_n^{r_j} = 0$. Because $u_n(\boldsymbol{y}_n^{i_j+1}, \boldsymbol{r}^{r_j}) \geq u_n(\boldsymbol{r}_n, \boldsymbol{r}^{r_j})$, $\forall \boldsymbol{r}_n$ and $\boldsymbol{y}_n^{i_j+1} = \boldsymbol{d}_n^{i_j} + \boldsymbol{r}_n^{i_j}$, letting $j \to \infty$ yields

$$u_n(\boldsymbol{w}_n, \boldsymbol{w}) \geq u_n(\boldsymbol{r}_n, \boldsymbol{w}), \forall \boldsymbol{r}_n, n = |\mathcal{H}| + 1, |\mathcal{H}| + 2. \tag{5.7.8}$$

The inequalities (5.7.7) and (5.7.8) imply that the $u_n'(\boldsymbol{r}_n, \boldsymbol{w}; \boldsymbol{d}_n)\,|_{\boldsymbol{r}_n = \boldsymbol{w}_n} \leq 0$, $\forall \boldsymbol{d}_n \in \mathbb{R}^{|r_n|}$, which is the directional derivative of u_n at point $\boldsymbol{w}_n$ in direction $\boldsymbol{d}_n$. Combining this with (5.7.6) yields

$$\mathcal{U}'(\boldsymbol{w}; \boldsymbol{d}) \leq 0, \forall \boldsymbol{d} = (0, \cdots, \boldsymbol{d}_n, \cdots, 0),$$

which is the directional derivative of $\mathcal{U}$ at point $\boldsymbol{w}$ in direction $\boldsymbol{d}$. This implies that point $\boldsymbol{w}$ is the stationary point of $\mathcal{U}$.

5.7.4 Derivation of Eq.(5.4.16)-(5.4.24)

$q(\boldsymbol{\nu}_t)$ in (5.4.16) can be derived as

$$\begin{aligned} &\ln q(\boldsymbol{\nu}_t) \propto \langle \ln p(\mathbf{y}_t | \boldsymbol{x}_t, \mathbf{z}_t, \boldsymbol{\nu}_t; \boldsymbol{\kappa}_t, \boldsymbol{\beta}_t) \rangle_{\mathbf{x}_t, \mathbf{z}_t} + \ln p(\boldsymbol{\nu}_t) \\ \propto &\sum_l -\nu_{t,l} \left\langle \left\| \mathbf{y}_{t,l} - \mathbf{A}_{t,l}(\boldsymbol{\kappa}_t)\, \mathbf{x}_{t,l} - \mathbf{B}_{t,l}(\boldsymbol{\beta}_{t,l})\, \mathbf{z}_{t,l} \right\|^2 \right\rangle_{\mathbf{x}_t, \mathbf{z}_t} + N_l \ln \nu_{t,l} + (a-1)\ln \nu_{t,l} - b\nu_{t,l} \\ \propto &\sum_l (c_{t,l} - 1)\ln \nu_{t,l} - d_{t,l}\nu_{t,l}. \end{aligned}$$

$q(\mathbf{x}_{t,l})$ in (5.4.17) can be obtained as

$$\begin{aligned}
&\ln q(\mathbf{x}_{t,l}) \propto \left\langle \ln p\left(\mathbf{y}_{t,l} \left| \mathbf{x}_{t,l}, \mathbf{z}_{t,l}, \nu_{t,l}; \boldsymbol{\kappa}_t, \boldsymbol{\beta}_{t,l}\right.\right)\right\rangle_{\mathbf{z}_{t,l},\nu_{t,l}} + \left\langle \ln p\left(\mathbf{x}_{t,l} \left| \boldsymbol{\gamma}_{t,l}\right.\right)\right\rangle_{\gamma_{t,l}} \\
\propto &- \left(\mathbf{x}_{t,l} - \boldsymbol{\mu}_{t,l}^{x}\right)^{H} \left(\boldsymbol{\Sigma}_{t,l}^{x}\right)^{-1} \left(\mathbf{x}_{t,l} - \boldsymbol{\mu}_{t,l}^{x}\right).
\end{aligned}$$

$q(\mathbf{z}_{t,l})$ in (5.4.18) can be obtained similarly. $q(\boldsymbol{\gamma}_t)$ in (5.4.19) can be obtained as

$$\begin{aligned}
&\ln q\left(\boldsymbol{\gamma}_t\right) \propto \left\langle \ln p\left(\mathbf{x}_t \left| \boldsymbol{\gamma}_t\right.\right)\right\rangle_{\mathbf{x}_t} + \left\langle \ln p\left(\boldsymbol{\gamma}_t \left| s_t\right.\right)\right\rangle_{s_t} \\
\propto &\sum_{l=1}^{L}\sum_{q=1}^{Q}\left(\tilde{a}_{t,l,q} - 1\right)\ln\gamma_{t,l,q} - \tilde{b}_{t,l,q}\gamma_{t,l,q}.
\end{aligned}$$

$q(\boldsymbol{\lambda}_t)$ in (5.4.20) can be obtained similarly. $q\left(s_t\right)$ in (5.4.21) can be obtained as

$$\begin{aligned}
&\ln q\left(s_t\right) \propto \left\langle \ln p\left(\boldsymbol{\gamma}_t \left| s_t\right.\right)\right\rangle_{\boldsymbol{\gamma}_t} + \ln\hat{p}\left(s_t\right) \\
\propto &\sum_{q,l} 1\left(s_t = q\right)\left(\ln\frac{\left(b_l\right)^{a_l}}{\Gamma(a_l)} + \left(a_l - 1\right)\left\langle\ln\gamma_{t,l,q}\right\rangle - b_l\left\langle\gamma_{t,l,q}\right\rangle\right) \\
&+ 1\left(s_t \neq q\right)\left(\ln\frac{\left(\bar{b}\right)^{\bar{a}}}{\Gamma(\bar{a})} + \left(\bar{a} - 1\right)\left\langle\ln\gamma_{t,l,q}\right\rangle - \bar{b}\left\langle\gamma_{t,l,q}\right\rangle\right) + \ln\sum_{q=1}^{Q}\pi_{t,q}\delta\left(s_t - q\right) \\
\propto &\ln\sum_{q=1}^{Q}\tilde{\pi}_{t,q}\delta\left(s_t - q\right).
\end{aligned}$$

$q\left(\mathbf{v}_t\right)$ in (5.4.23) can be obtained similarly.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The central theme of this thesis has been the CS algorithm designs with applications to massive MIMO systems, such as the channel estimation in massive MIMO systems, channel tracking in massive MIMO systems and user location tracking in massive MIMO systems. Different application scenarios will pose different challenges for the CS algorithm design. Therefore, it's necessary to design efficient and robust CS algorithm to address the challenges under different applications and exploit the additional beneficial information to enhance the CS recovery performance. We summarize the contributions as follows.

6.1.1 Weighted LASSO for Massive MIMO Channel Estimation

In practice, it's possible to obtain statistical PSI about the sparse signal to be recovered and it's critical to optimally incorporate such statistical PSI to enhance the recovery performance. We propose a weighted LASSO algorithm to fully exploit the statistical PSI and optimize the recovery performance. We also derive the closed-form accurate expression for the minimum aNSE and analyze the minimum number of measurement required for stable recovery.

In massive MIMO system, BS can obtain certain channel support side information (CSSI), which can be exploited to enhance the CE performance and reduce the pilot overhead. Therefore, the weighted LASSO algorithm can be applied for CE problem in massive MIMO system to utilize the CSSI in an optimal way. Based on the accuracy of the CSSI, the optimal LASSO weights which minimize the aNSE can be obtained. Moreover, we can characterize the minimum number of pilots required to achieve stable channel recovery, which provides

valuable instructions for practical CE problem.

6.1.2 Dynamic Turbo-OAMP for Massive MIMO Channel Tracking

CS has been applied to exploit the structured sparsity of massive MIMO channels to reduce the pilot overheads in massive MIMO downlink channel estimation and channel tracking. However, the existing structured sparse channel estimation algorithms are designed based on oversimplified channel models with restrictive assumptions, and thus perform poorly under realistic channels. We propose a new statistical model, i.e., 2D-MM, to capture the 2D dynamic sparsity (i.e., structured sparsity in the spatial domain and probabilistic temporal dependency of channel in the temporal domain) of massive MIMO channels, which has the flexibility to model different propagation environment in practice. By combining the turbo approach and the OAMP, we derive an efficient message passing algorithm called D-TOAMP to recursively track a dynamic massive MIMO channel with 2D-MM sparsity prior. The proposed D-TOAMP can achieve a better performance than AMP-based algorithm due to the usage of orthogonal measurement matrix and the exploitation of the structured sparsity. Moreover, the proposed D-TOAMP provides a systematic framework for the tracking of the dynamic sparse signals.

6.1.3 Turbo-VBI for Robust Recovery of Structured Sparse Signals with Uncertain Measurement Matrix

In many practical applications in wireless communications, we need to solve the problem of recovering a structured sparse signal from a linear measurement model with uncertain measurement matrix. There are two challenges of designing a general algorithm framework for this problem. The first is how to design a flexible and tractable sparse prior to capture different structured sparsities in specific application. The second is how to handle a general measurement matrix that may contain uncertain parameters and may be ill-conditioned with correlated columns. Due to the restrictions of the existing CS recovery methods, we need to propose an efficient and robust CS algorithm to overcome these challenges. Specifically, we first propose a 3LHS sparse prior to capture various sophisticated structured sparsities that may occur in practice. Then, by combing the message passing and VBI approaches via turbo framework, we propose a Turbo-VBI algorithm to fully exploit the 3LHS structured

sparsity prior under an uncertain measurement matrix to achieve robust and efficient recovery performance.

6.1.4 D-VBI for User Location Tracking in Massive MIMO Systems

In 5G network, accurate user location tracking is the key to enable location-based services and assist communications. To improve the location tracking accuracy, user mobility model and temporal correlation of massive MIMO channels can be utilized. Moreover, the cooperative localization of multiple BSs based on the location grid induces a group-sparsity structure of the LOS channels. We propose a 3LHS sparse prior called TMGS to jointly capture the temporal correlation and group sparsity of massive MIMO channels in location tracking problem. Then, a variant of Turbo-VBI algorithm, i.e., D-VBI, is proposed to handle the TMGS prior under ill-conditioned measurement matrix which contains off-grid uncertain parameters. Due to the probabilistic temporal dependencies of massive MIMO channels, the D-VBI algorithm can provide prior information about the support of massive MIMO channels in the next time slot to improve the location tracking accuracy

6.2 Future Work

CS plays a key role in many engineering and scientific applications, such as image signal processing [105], wireless communications [7,10], autonomous driving [106], etc. Therefore, developing novel CS algorithms to accommodate to different application requirements still has great potential.

6.2.1 Efficient Robust CS Algorithm Design for Large Dimensional Problem

When we need to reconstruct a large dimensional (in the order of 10^4-10^6) sparse signal from a noisy measurement, it's necessary to design a computational efficient CS algorithm to reduce the computation cost. SBL/VBI has some appealing properties, such as good recovery performance compared to greedy algorithms, robust to the measurement matrix compared to AMP-based algorithm. However, one of the primary deficiencies of the Bayesian methods is their high computational complexity due to the covariance matrix inversion in Gaussian

models. Even though the AMP-based approach has low complexity (linear complexity), the fact that it only performs well under i.i.d. or partially orthogonal measurement matrix will hinder its usage in many practical applications. Therefore, for a future research direction, it's interesting to develop an efficient robust CS algorithm for large dimensional problem, which has low computational complexity and is robust to different types of measurement matrices.

One possible application of efficient CS algorithm is massive connectivity, which is a key requirement for future wireless cellular networks. In a massive device connectivity scenario, a BS may be required to connect a large number of devices (in the order 10^4 to 10^6) and the device activity patterns are typically sporadic so that at any given time, only a small fraction of devices are active [107]. The BS needs to identify the active users and estimate their channels before the data transmission takes place. Due to the large number of potential devices and the massive antennas with arbitrary geometry employed at the BS, the resulting CS problem is high dimensional, and an efficient robust CS algorithm design would be necessary to solve the large dimensional problem with low complexity and robust performance.

6.2.2 Robust Bilinear CS Algorithm Design

The existing bilinear CS algorithms include massage-passing-based algorithms, such as BiGAMP [108], PBiGAMP [109], bilinear adaptive VAMP (BAd-VAMP) [110] and SBL-based algorithm in [111]. The massage-passing-based bilinear CS algorithms have i.i.d. assumptions on the measurement matrices or the matrices needed to be recovered, which will make them unusable for some applications. The SBL-based bilinear CS algorithm has high computational complexity and it's difficult to incorporate the complicated prior for the sparse matrix. Thus, it's necessary to develop robust bilinear CS algorithm to jointly recover structured sparse matrix and dense matrix from linear measurements with uncertain measurement matrices.

One possible application of robust bilinear CS algorithm is joint channel estimation and data detection in massive MIMO system. The joint channel-data estimation can improve the system performance since the partially detected data symbols can be used as soft pilots to enhance the quality of channel estimation in an iterative way. The resulting problem is a bilinear CS recovery problem with the problem formulation given by $\mathbf{Y} = \mathbf{A}(\boldsymbol{\vartheta})\mathbf{S}\mathbf{B}(\boldsymbol{\beta})\mathbf{X} + \mathbf{Z}$, where $\mathbf{S}$ is the sparse angular domain channel, which exhibits spatial structured sparsity; $\mathbf{X}$ is the transmitted data symbol, which should satisfy the finite-alphabet constraint; $\mathbf{A}$ and $\mathbf{B}$ are the array response matrices at BS side and user side, which could have general forms

based on the antenna geometry; $\boldsymbol{\vartheta}$ and $\boldsymbol{\beta}$ are the off-grid parameters to eliminate the power leakage due to the angular discretization at BS side and user side. The robust bilinear CS algorithm should be designed to recover the structured sparse matrix $\mathbf{S}$ and the dense matrix $\mathbf{X}$ from the measurement $\mathbf{Y}$ simultaneously and automatically learn the off-grid parameters $\boldsymbol{\vartheta}$ and $\boldsymbol{\beta}$ in the recovery process.

6.2.3 Non-linear CS Algorithm Design

To incorporate the physical impairments in wireless communication data path, such as the non-linearity of power amplifier (PA), quantization effect or low resolution analog to digital converter (ADC), it's necessary to design a general non-linear CS algorithm. Consider the following model: $\mathbf{y} = f(\mathbf{A}\mathbf{x} + \mathbf{n})$, where $\mathbf{A} \in \mathbb{C}^{M \times N}$ is a general sensing matrix, $\mathbf{x} \in \mathbb{C}^N$ is a sparse signal, $\mathbf{n} \in \mathbb{C}^M$ is a general noise vector (could be non-Gaussian), $f : \mathbb{C}^M \to \mathbb{C}^M$ is a non-linear function. One possible research direction is to figure out the requirement on the nonlinear function f and the system parameters such as the sparsity ratio $\|\mathbf{x}\|_0 / N$ and the sampling rate M/N, such that the stable recovery of $\mathbf{x}$ from $\mathbf{y}$ is possible. It's also interesting to develop efficient non-linear CS algorithm to achieve robust recovery of sparse signal from a nonlinear measurement model.

Bibliography

[1] M. K. Samimi and T. S. Rappaport, "3-D millimeter-wave statistical channel model for 5G wireless system design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 7, pp. 2207–2225, 2016.

[2] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge University Press, 2005.

[3] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2477–2488, 2005.

[4] W. Lu and N. Vaswani, "Modified compressive sensing for real-time dynamic mr imaging," in *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 3045–3048.

[5] W. Li and J. C. Preisig, "Estimation of rapidly time-varying sparse channels," *IEEE Journal of Oceanic Engineering*, vol. 32, no. 4, pp. 927–939, 2007.

[6] S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.

[7] A. Liu, V. K. Lau, and W. Dai, "Exploiting burst-sparsity in massive MIMO with partial channel support information," *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7820–7830, 2016.

[8] L. Chen, A. Liu, and X. Yuan, "Structured turbo compressed sensing for massive MIMO channel estimation using a Markov prior," *IEEE Transactions on Vehicular Technology*, 2017.

[9] Z. Gao, L. Dai, W. Dai, B. Shim, and Z. Wang, "Structured compressive sensing-based spatio-temporal joint channel estimation for FDD massive MIMO," *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 601–617, 2016.

[10] Z. Gao, L. Dai, Z. Wang, and S. Chen, "Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO," *IEEE transactions on signal processing*, vol. 63, no. 23, pp. 6169–6183, 2015.

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

[12] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[13] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and computational harmonic analysis*, vol. 26, no. 3, pp. 301–321, 2009.

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

[15] X. Rao and V. K. Lau, "Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3261–3271, 2014.

[16] M. F. Duarte and Y. C. Eldar, "Structured compressed sensing: From theory to applications," *IEEE Transactions on signal processing*, vol. 59, no. 9, pp. 4053–4085, 2011.

[17] C. Thrampoulidis, A. Panahi, and B. Hassibi, "Asymptotically exact error analysis for the generalized ℓ_2^2-lasso," 2015.

[18] M. Stojnic, F. Parvaresh, and B. Hassibi, "On the reconstruction of block-sparse signals with an optimal number of measurements," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 3075–3085, 2009.

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

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

[21] J. Ma, X. Yuan, and L. Ping, "Turbo compressed sensing with partial DFT sensing matrix," *IEEE Signal Processing Letters*, vol. 22, no. 2, pp. 158–161, 2015.

[22] ——, "On the performance of turbo signal recovery with partial DFT sensing matrices," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1580–1584, 2015.

[23] J. Ma and L. Ping, "Orthogonal AMP," *IEEE Access*, vol. 5, pp. 2020–2033, 2017.

[24] P. Schniter, "Turbo reconstruction of structured sparse signals," in *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, March 2010, pp. 1–6.

[25] L. Chen, A. Liu, and X. Yuan, "Structured turbo compressed sensing for massive MIMO channel estimation using a Markov prior," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4635–4639, 2018.

[26] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a Markov-tree prior," *IEEE transactions on signal processing*, vol. 60, no. 7, pp. 3439–3448, 2012.

[27] J. Dai, A. Liu, and V. K. Lau, "FDD massive MIMO channel estimation with arbitrary 2D-array geometry," *IEEE Transactions on Signal Processing*, vol. 66, no. 10, 2018.

[28] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 131–146, 2008.

[29] C. W. Fox and S. J. Roberts, "A tutorial on variational Bayesian inference," *Artificial intelligence review*, vol. 38, no. 2, pp. 85–95, 2012.

[30] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[31] A. Liu, L. Lian, V. K. Lau, and X. Yuan, "Downlink channel estimation in multiuser massive mimo with hidden markovian sparsity," *IEEE Transactions on Signal Processing*, vol. 66, no. 18, pp. 4796–4810.

[32] E. Larsson, O. Edfors, F. Tufvesson, and T. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, 2014.

[33] E. Telatar, "Capacity of multi-antenna Gaussian channels," *European transactions on telecommunications*, vol. 10, no. 6, pp. 585–595, 1999.

[34] X. Zhu, L. Dai, W. Dai, Z. Wang, and M. Moonen, "Tracking a dynamic sparse channel via differential orthogonal matching pursuit," in *MILCOM 2015 - 2015 IEEE Military Communications Conference*, Oct 2015, pp. 792–797.

[35] X. Zhu, L. Dai, G. Gui, W. Dai, Z. Wang, and F. Adachi, "Structured matching pursuit for reconstruction of dynamic sparse channels," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–5.

[36] X. Rao and V. K. Lau, "Compressive sensing with prior support quality information and application to massive MIMO channel estimation with temporal correlation," *IEEE Transactions on Signal Processing*, vol. 63, no. 18, pp. 4914–4924, 2015.

[37] L. Dai and X. Gao, "Priori-aided channel tracking for millimeter-wave beamspace massive MIMO systems," in *2016 URSI Asia-Pacific Radio Science Conference (URSI AP-RASC)*, Aug 2016, pp. 1493–1496.

[38] A. Guerra, F. Guidi, and D. Dardari, "Position and orientation error bound for wideband massive antenna arrays," in *2015 IEEE International Conference on Communication Workshop (ICCW)*. IEEE, 2015, pp. 853–858.

[39] A. Shahmansoori, G. E. Garcia, G. Destino *et al.*, "5G position and orientation estimation through millimeter wave MIMO," in *2015 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2015, pp. 1–6.

[40] K. Huang, R. W. Heath Jr, and J. G. Andrews, "Limited feedback beamforming over temporally-correlated channels," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 1959–1975, 2009.

[41] 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, 2011.

[42] N. Vaswani and W. Lu, "Modified-CS: Modifying compressive sensing for problems with partially known support," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4595–4607, 2010.

[43] H. Mansour and R. Saab, "Recovery analysis for weighted ℓ_1-minimization using the null space property," *Applied and Computational Harmonic Analysis*, 2015.

[44] M. A. Khajehnejad, W. Xu, A. S. Avestimehr, and B. Hassibi, "Weighted â 1 minimization for sparse recovery with prior information," in *IEEE International Symposium on Information Theory, 2009. ISIT 2009*. IEEE, 2009, pp. 483–487.

[45] S. Oymak, M. A. Khajehnejad, and B. Hassibi, "Recovery threshold for optimal weight ℓ_1 minimization," in *2012 IEEE International Symposium on Information Theory Proceedings*, July 2012, pp. 2032–2036.

[46] A. Flinth, "Optimal choice of weights for sparse recovery with prior information," *IEEE Transactions on Information Theory*, vol. 62, no. 7, pp. 4276–4284, 2016.

[47] J. Zhan and N. Vaswani, "Time invariant error bounds for modified-cs-based sparse signal sequence recovery," *IEEE Transactions on Information Theory*, vol. 61, no. 3, pp. 1389–1409, 2015.

[48] W. Chen and Y. Li, "Recovery of signals under the high order rip condition via prior support information," *arXiv preprint arXiv:1603.03464*, 2016.

[49] H. Mansour and Ö. Yilmaz, "Weighted-ℓ_1 minimization with multiple weighting sets," in *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 2011, pp. 813 809–813 809.

[50] D. Needell, R. Saab, and T. Woolf, "Weighted-minimization for sparse recovery under arbitrary prior information," *Information and Inference: A Journal of the IMA*, p. iaw023, 2017.

[51] W. Lu and N. Vaswani, "Modified basis pursuit denoising (modified-bpdn) for noisy compressive sensing with partially known support," in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 3926–3929.

[52] C. Thrampoulidis, A. Panahi, and B. Hassibi, "Asymptotically exact error analysis for the generalized equation-lasso," in *2015 IEEE International Symposium on Information Theory (ISIT)*, June 2015, pp. 2021–2025.

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

[54] L. Jacques, "A short note on compressed sensing with partially known signal support," *Signal Processing*, vol. 90, no. 12, pp. 3308–3312, 2010.

[55] L. Bottou and N. Murata, "Stochastic approximations and efficient learning," *The Handbook of Brain Theory and Neural Networks, Second edition,. The MIT Press, Cambridge, MA*, 2002.

[56] C. Thrampoulidis, S. Oymak, and B. Hassibi, "The Gaussian min-max theorem in the presence of convexity," *arXiv preprint arXiv:1408.4837*, 2014.

[57] S. Oymak, C. Thrampoulidis, and B. Hassibi, "The squared-error of generalized lasso: A precise analysis," in *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2013, pp. 1002–1009.

[58] 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.

[59] Y. Wu and S. Verdú, "Optimal phase transitions in compressed sensing," *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6241–6263, 2012.

[60] H. Yin, D. Gesbert, M. Filippou, and Y. Liu, "A coordinated approach to channel estimation in large-scale multiple-antenna systems," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 2, pp. 264–273, 2013.

[61] C. R. Berger, Z. Wang, J. Huang, and S. Zhou, "Application of compressive sensing to sparse channel estimation," *IEEE Communications Magazine*, vol. 48, no. 11, pp. 164–174, 2010.

[62] O. Mehanna and N. D. Sidiropoulos, "Channel tracking and transmit beamforming with frugal feedback," *IEEE Transactions on Signal Processing*, vol. 62, no. 24, pp. 6402–6413, 2014.

[63] X. Wang, J. Wang, L. He, C. Pan, and J. Song, "Basis expansion model based spectral efficient channel recovery scheme for spatial-temporal correlated massive MIMO systems," *IET Communications*, vol. 11, no. 17, pp. 2621–2629, 2017.

[64] X. Liu, Y. Shi, J. Zhang, and K. B. Letaief, "Massive CSI acquisition in dense cloud-RAN with spatial and temporal prior information," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[65] J. Zhao, F. Gao, W. Jia, J. Zhao, and W. Zhang, "Channel tracking for massive MIMO systems with spatial-temporal basis expansion model," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–5.

[66] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *IEEE transactions on signal processing*, vol. 61, no. 21, pp. 5270–5284, 2013.

[67] E. Fornasini, "2D Markov chains," *Linear Algebra and its Applications*, vol. 140, pp. 101–127, 1990.

[68] ——, "2D Markov chains," *Linear Algebra and its Applications*, vol. 140, pp. 101–127, 1990.

[69] J. Nocedal and S. J. Wright, "Numerical optimization 2nd," 2006.

[70] C. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of statistics*, pp. 95–103, 1983.

[71] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Transactions on communications*, vol. 44, no. 10, pp. 1261–1271, 1996.

[72] J. Salo, G. D. Galdo, J. Salmi, P. Kyosti, M. Milojevic, D. Laselva, and C. Schneider, "MATLAB implementation of the 3GPP spatial channel model (3GPP TR 25.996)," 2005, Jan. [Online]. Available: http://www.tkk.fi/Units/Radio/scm/

[73] L. Lian, A. Liu, and V. K. N. Lau, "Weighted LASSO for sparse recovery with statistical prior support information," *IEEE Transactions on Signal Processing*, vol. 66, no. 6, pp. 1607–1618, March 2018.

[74] Z. Gao, L. Dai, S. Han, I. Chih-Lin, Z. Wang, and L. Hanzo, "Compressive sensing techniques for next-generation wireless communications," *IEEE Wireless Communications*, 2018.

[75] J. Fang, Y. Shen, H. Li, and P. Wang, "Pattern-coupled sparse bayesian learning for recovery of block-sparse signals," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 360–372, Jan 2015.

[76] S. Ji, Y. Xue, L. Carin *et al.*, "Bayesian compressive sensing," *IEEE Transactions on signal processing*, vol. 56, no. 6, p. 2346, 2008.

[77] D. P. Wipf and B. D. Rao, "Sparse bayesian learning for basis selection," *IEEE Transactions on Signal processing*, vol. 52, no. 8, pp. 2153–2164, 2004.

[78] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[79] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Info. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[80] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear gauss-seidel method under convex constraints," *Operat. Res. Lett.*, vol. 26, pp. 127–136, 2000.

[81] A. Cupper, G. Treu, and C. Linnhoff-Popien, "TraX: A device-centric middleware framework for location-based services," *IEEE Communications Magazine*, vol. 44, no. 9, pp. 114–120, 2006.

[82] Y. Liu, X. Shi, S. He, and Z. Shi, "Prospective positioning architecture and technologies in 5G networks," *IEEE Network*, vol. 31, no. 6, pp. 115–121, November 2017.

[83] U. Bareth and A. Kupper, "Energy-efficient position tracking in proactive location-based services for smartphone environments," in *2011 IEEE 35th Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, 2011, pp. 516–521.

[84] V. Savic and E. G. Larsson, "Fingerprinting-based positioning in distributed massive MIMO systems," in *2015 IEEE 82nd Vehicular Technology Conference (VTC Fall)*. IEEE, 2015, pp. 1–5.

[85] K. N. R. S. V. Prasad, E. Hossain, and V. K. Bhargava, "Machine learning methods for RSS-based user positioning in distributed massive MIMO," *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 8402–8417, Dec 2018.

[86] H. Deng and A. Sayeed, "Mm-wave MIMO channel modeling and user localization using sparse beamspace signatures," in *2014 IEEE 15th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2014, pp. 130–134.

[87] A. Hu, T. Lv, H. Gao, Z. Zhang, and S. Yang, "An ESPRIT-based approach for 2-D localization of incoherently distributed sources in massive MIMO systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 996–1011, 2014.

[88] A. Shahmansoori, G. E. Garcia, G. Destino *et al.*, "Position and orientation estimation through millimeter-wave MIMO in 5G systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1822–1835, 2018.

[89] Z. Lin, T. Lv, and P. T. Mathiopoulos, "3-D indoor positioning for millimeter-wave massive MIMO systems," *IEEE Transactions on Communications*, pp. 1–1, 2018.

[90] N. Garcia, H. Wymeersch, E. G. Larsson, A. M. Haimovich, and M. Coulon, "Direct localization for massive MIMO," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2475–2487, 2017.

[91] A. J. Weiss, "Direct position determination of narrowband radio frequency transmitters," *IEEE signal processing letters*, vol. 11, no. 5, pp. 513–516, 2004.

[92] O. Bialer, D. Raphaeli, and A. J. Weiss, "Maximum-likelihood direct position estimation in dense multipath," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2069–2079, 2013.

[93] T. Laursen, N. B. Pedersen, J. J. Nielsen, and T. K. Madsen, "Hidden Markov model based mobility learning for improving indoor tracking of mobile users," in *2012 9th Workshop on Positioning Navigation and Communication (WPNC)*. IEEE, 2012, pp. 100–104.

[94] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 912–926, 2011.

[95] C. Kurisummoottil Thomas and D. T. Slock, "Gaussian variational Bayes Kalman filtering for dynamic sparse Bayesian learning," in *ITISE 2018, 5th International Conference on Time Series and Forecasting, 19-21 September 2018, Granada, Spain*, Granada, SPAIN, 09 2018.

[96] L. Lian, A. Liu, and V. K. Lau, "Exploiting dynamic sparsity for downlink FDD-massive MIMO channel tracking," *IEEE Transactions on Signal Processing*, 2019.

[97] B. Zhou, Q. Chen, T. J. Li, and P. Xiao, "Online variational Bayesian filtering-based mobile target tracking in wireless sensor networks," *Sensors*, vol. 14, no. 11, pp. 21 281–21 315, 2014.

[98] S. Farahmand, G. B. Giannakis, G. Leus, and Z. Tian, "Sparsity-aware Kalman tracking of target signal strengths on a grid," in *2011 Proceedings of the 14th International Conference on Information Fusion (FUSION)*. IEEE, 2011, pp. 1–6.

[99] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of machine learning research*, vol. 1, no. Jun, pp. 211–244, 2001.

[100] Q. Spencer, M. Rice, B. Jeffs, and M. Jensen, "Indoor wideband time/angle of arrival multipath propagation results," in *1997 IEEE 47th Vehicular Technology Conference. Technology in Motion*, vol. 3. IEEE, 1997, pp. 1410–1414.

[101] Y. Zhou, M. Herdin, A. M. Sayeed, and E. Bonek, "Experimental study of MIMO channel statistics and capacity via the virtual channel representation," *Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep*, vol. 5, pp. 10–15, 2007.

[102] K. Haneda *et al.*, "5G 3GPP-like channel models for outdoor urban microcellular and macrocellular environments," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, May 2016, pp. 1–7.

[103] L. Jiang, L. Thiele, and V. Jungnickel, "On the modelling of polarized MIMO channel," *Proc. Europ. Wireless*, vol. 2007, pp. 1–4, 2007.

[104] M. Razaviyayn, "Successive convex approximation: Analysis and applications," Ph.D. dissertation, University of Minnesota, 2014.

[105] L. He and L. Carin, "Exploiting structure in wavelet-based bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 57, no. 9, pp. 3488–3497, Sept 2009.

[106] J. A. del Peral-Rosado, J. A. Lopez-Salcedo, S. Kim, and G. Seco-Granados, "Feasibility study of 5G-based localization for assisted driving," in *2016 International Conference on Localization and GNSS (ICL-GNSS)*, June 2016, pp. 1–6.

[107] L. Liu and W. Yu, "Massive device connectivity with massive MIMO," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 1072–1076.

[108] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing-Part I: Derivation," *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 5839–5853, 2014.

[109] J. T. Parker and P. Schniter, "Parametric bilinear generalized approximate message passing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 795–808, 2016.

[110] S. Sarkar, A. K. Fletcher, S. Rangan, and P. Schniter, "Bilinear recovery using adaptive vector-amp," *IEEE Transactions on Signal Processing*, vol. 67, no. 13, pp. 3383–3396, 2019.

[111] R. Prasad, C. R. Murthy, and B. D. Rao, "Joint channel estimation and data detection in MIMO-OFDM systems: A sparse Bayesian learning approach," *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5369–5382, 2015.