Linear Dynamical Systems with Sparsity Constraints: Theory and Algorithms

A thesis

submitted in partial fulfilment of the

requirements for the degree of

Doctor of Philosophy

in the Faculty of Engineering

by

Geethu Joseph

Electrical Communication Engineering
Indian Institute of Science, Bangalore
Bangalore – 560 012 (INDIA)

September 2019

To Daniel and Hannah

Acknowledgments

My Ph.D journey has been an extremely inspiring and enjoyable experience, a time of fantastic intellectual and personal growth. I feel very fortunate to have received adequate support and encouragement from my advisor, my family, and friends. They helped me to achieve a healthy work-life balance.

Firstly, I would like to express my sincere gratitude to my advisor *Prof. Chandra R. Murthy* for his wholehearted technical and moral support, patience, and valuable guidance. He initiated me into the world of research, and gave me enormous freedom to work on my own research problems. He taught me how to pursue research and present results elegantly. Our weekly progress meetings helped me to keep motivated when I was stumped about the direction of my research. I also appreciate his assistance in developing my writing skill throughout, and especially in writing papers, the Intel India PhD fellowship application, and this thesis.

I thank my collaborators for all the fruitful interactions: *Ranjitha Prasad*, who is currently at Tata Consultancy Services Innovation Labs, Delhi; *Prof. Bhaskar D. Rao*, who is at the University of California, San Diego, USA; *Ahmad Zoubi*, who is currently at Qualcomm, Oregon, USA; and *Prof. V. John Mathews*, who is at Oregon State University, Corvallis, USA. I am indebted to the faculty members of ECE, EE, CSA and Math departments, where I have attended many interesting courses. I thank the staff members of ECE department, especially, *Ms. Suma T. C.* and *Mr. Srinivasa Murthy* for all the administrative help. I acknowledge with gratitude Intel India for awarding me a fellowship which supported my Ph.D. and conference travels.

I have been blessed with a very loving and supportive family. My heart-felt gratitude goes to my wonderful parents *Sherly* and *Joseph* who have been a source of constant inspiration and unconditional love. They have always supported me in my decisions and encouraged

me to pursue my dreams. I thank with love my husband *Jomon* with whom I shared the most part of this amazing journey. I appreciate his patient listening, timely advice, and sacrifices to make my career a priority in our lives. I am grateful to my little bundles of joy and laughter: *Daniel* and *Hannah*. Their bright smiles put it all into perspective for me. A special thanks go to my kind and helpful brother *Jitthu* who happily ran a hundred errands in IISc for me. I owe a debt of gratitude to my enthusiastic grandmother, *Mary*. Despite her old age, she stayed with me and gave invaluable support during my pregnancy and early days of parenthood. I also extend my deepest gratitude to my mother-in-law *Gracykutty* for taking care of my babies very well.

Finally, I would like to convey sincere gratitude to all my IISc friends. I am privileged to have shared the lab space with *Saurabh, Partha, Mohit, Shilpa, Jobin, Thirupathaiah, Suma, Praveen, Ribhu, Bala, Nandan, Chirag* and *Rubin*. I will cherish the stimulating discussions, lunch and ice cream treats, and all the fun we had all my life. I also thank *Saurabh, Mohit, Ranjitha, Abhay, Venu* and other members of SPC lab for the invaluable technical exchanges. I offer gratitude to all my hostel mates, particularly, *Arathi, Indu, Jeenu, Nazreen, Vandhana, Anu, Priti, Chenju* and *Seena*. Our long chats in the C-mess hall, birthday celebrations and photo sessions made my hostel life colorful. Also, my friend *Rahul* deserves a special mention for all the interesting times together.

Thanks for all for your encouragement!

Abstract

This thesis develops new mathematical theory and presents novel recovery algorithms for discrete linear dynamical systems (LDS) with sparsity constraints on either control inputs or initial state. The recovery problems in this framework manifest as the problem of reconstructing one or more sparse signals from a set of noisy underdetermined linear measurements. The goal of our work is to design algorithms for sparse signal recovery which can exploit the underlying structure in the measurement matrix and the unknown sparse vectors, and to analyze the impact of these structures on the efficacy of the recovery.

We answer three fundamental and interconnected questions on sparse signal recovery problems that arise in the context of LDS. First, what are necessary and sufficient conditions for the existence of a sparse solution? Second, given that a sparse solution exists, what are good low-complexity algorithms that exploit the underlying signal structure? Third, when are these algorithms guaranteed to succeed? These questions are considered in the context of three different sparsity models, as described below.

Within the LDS framework, we first consider the simplest sparsity model of a single unknown sparse initial state vector with no additional structure. This problem is known as the observability problem in the control theory literature, and the initial state can be recovered using standard compressed sensing (CS) algorithms. However, the recovery guarantees for this case are different from the classical sparse recovery guarantees because the measurement matrix that arises in LDS is fundamentally different from the matrices that are typically considered in the CS literature. We seek to obtain the conditions for observability of LDS when the initial state is sparse and the observation matrix is random. Taking advantage of randomness in the measurements, we use concentration inequalities to derive an upper bound on the minimum number of measurements that can ensure faithful recovery of the sparse initial state.

Next, we move to a more complicated sparsity model, which is concerned with the recovery of a set of sparse control input vectors. In this setting, we first derive necessary and sufficient conditions for the existence of a sparse solution for any given pair of initial and final states in the LDS. These conditions enable us to develop a simple procedure to test the controllability of LDS using sparse inputs, which is non-combinatorial in nature, unlike the existing sparse-controllability tests.

Following the existence test, we address the second question, namely that of devising low-complexity recovery algorithms. We develop online non-iterative algorithms for the same sparsity model. Motivated by the wideband wireless channel estimation problem, we assume that the control inputs are jointly sparse, and the system transfer matrix is diagonal. We devise two online algorithms based on the sparse Bayesian learning framework. The algorithms are implemented using the sequential expectation-maximization procedure, combined with Kalman smoothing. Consequently, they require minimal computational and memory resources, and have bounded delays. Further, we rigorously examine the properties of the algorithm to answer the third question on recovery guarantees. The analysis involves elegant use of tools from stochastic approximation theory.

Finally, we present the most sophisticated sparsity model considered in the thesis, where both the control inputs and observation matrix are assumed to be unknown. This problem is referred to as the dictionary learning problem in the CS literature. Here, we focus on algorithm development and establishing its guarantees. We adopt a Bayesian approach for the recovery, and solve the resulting optimization problem using the alternating minimization procedure and the Armijo line search procedure. We then provide recovery guarantees by characterizing the properties of the algorithm using Kurdyka-Łojasiewicz-based analysis. We also show that the algorithm is likely to converge to a sparse representation.

Apart from the above set of algorithms and theoretical results, we also apply the sparse signal recovery framework to anomaly imaging for structural health monitoring. The goal here is to recover the anomaly map of a structure using multi-sensor measurements. We develop an algorithm that exploits the inherit clustered sparsity in the map, and benchmark its performance against two state-of-the-art algorithms using real-world damage measurements.

Overall, the thesis presents rigorous theoretical analysis and accurate yet low complexity algorithms for sparse recovery problems that arise in the context of LDS.

Glossary

AM	:	Alternating Minimization
a.s.	:	almost surely
ALS	:	Armijo Line Search
AR	:	Auto-Regressive
CoSaMP	:	Compressive Sampling Matching Pursuit
CS	:	Compressive Sensing
EM	:	Expectation Maximization
i.i.d.	:	Independent and Identically Distributed
ITH	:	Iterative Hard Thresholding
KL	:	Kullback-Leibler
KM-SBL	:	Kalman Multiple Sparse Bayesian Learning
LASSO	:	Least Angle Absolute Shrinkage and Selection Operator
LHS	:	Left Hand Side
LDS	:	Linear Dynamical Systems
MAP	:	Maximum a Posteriori
ML	:	Maximum Likelihood
MSE	:	Mean Squared Error
MMV	:	Multiple Measurement Vector
M-SBL	:	Multiple Sparse Bayesian Learning
PBH	:	Popov-Belevitch-Hautus
OFDM	:	Orthogonal Frequency Division Multiplexing
OMP	:	Orthogonal Matching Pursuit
RHS	:	Right Hand Side
RMSE	:	Relative Mean Square Error
RIC	:	Restricted Isometry Constant
RIP	:	Restricted Isometry Property
SBL	:	Sparse Bayesian Learning
SNR	:	Signal-to-Noise Ratio
SRR	:	Support Recovery Rate

Notation

Vectors and matrices are denoted by boldface small letters and boldface capital letters, respectively. Sets are denoted by calligraphy letters. Rest of the notation is listed below.

Vector

$\boldsymbol{a}_i$	:	i^{th} element of vector $\boldsymbol{a}$
$\|\cdot\|$	:	Euclidean norm of a vector
$\|\cdot\|_0$	:	Number of nonzero entries of a vector
$\|\cdot\|_1$	:	ℓ_1-norm of a vector
$\|\cdot\|_\infty$	:	Infinity norm of a vector
$\mathsf{Diag}\{\cdot\}$	:	Diagonal matrix of with entries of a vector on the diagonal
$\mathsf{Supp}\{\cdot\}$	:	Support set of a vector

Matrix

$\boldsymbol{A}_{ij}$	:	$(i,j)^{\mathsf{th}}$ entry of matrix $\boldsymbol{A}$
$\boldsymbol{A}_i$	:	i^{th} column of matrix $\boldsymbol{A}$
$\left(\boldsymbol{A}^{\mathsf{T}}\right)_i$	:	i^{th} row of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{\mathcal{S}}$	:	Set of columns of matrix $\boldsymbol{A}$ indexed by the set $\mathcal{S}$
$(\cdot)^{\mathsf{T}}$	:	Transpose of a matrix
$\|\cdot\|$	:	Determinant of a matrix
$(\cdot)^\dagger$	:	Pseudo-inverse of a matrix
$\mathsf{Tr}\{\cdot\}$	:	Trace of a matrix
$\mathsf{Rank}\{\cdot\}$	:	Rank of a matrix
$\|\cdot\|_F$	:	Frobenius norm of a matrix
$\|\cdot\|_2$	:	Spectral norm of a matrix
$\mathcal{D}\{\cdot\}$	:	Diagonal matrix with same diagonal entries as the argument matrix
$\mathsf{vec}\{\cdot\}$	:	Vectorized version of a matrix
$\mathcal{CS}\{\cdot\}$	:	Column space of a matrix
$\odot$	:	Khatri-Rao product of matrices

Field

$\mathbb{R}$	:	Field of real numbers
$\mathbb{R}_+$	:	Field of non-negative real numbers
$\mathbb{C}$	:	Field of complex numbers

Probability

$\mathbb{P}\{\cdot\}$	:	Probability of an event
$\mathbb{E}\{\cdot\}$	:	Expectation operator
$\mathcal{N}$	:	Normal distribution

Set

$\lvert\cdot\rvert$	:	Cardinality of a set
$(\cdot)^{\mathrm{c}}$	:	Complement of a set
$\cup$	:	Union of two sets
$\cap$	:	Intersection of two sets

Miscellaneous

$\mathbf{0}$	:	All zero vector or matrix
$\mathbf{1}$	:	All ones vector
$\boldsymbol{I}$	:	Identity matrix
$\mathbb{1}_{\{\cdot\}}$	:	Indicator function

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Connecting sparsity and state space models

Linear dynamical systems (LDS) are well studied and widely accepted mathematical models for describing and analyzing a variety of physical systems that evolve in time. The model serves as the core engine in diverse fields such as automatic control systems [1], signal processing [2], communications [3], economics [4], mechanical and civil engineering [5, 6], health care [7, 8], etc. The canonical representation of the linear dynamics is the discrete-time state space model as given below:

$$\boldsymbol{x}_k = \boldsymbol{D}\boldsymbol{x}_{k-1} + \boldsymbol{H}\boldsymbol{h}_k \tag{1.1}$$

$$\boldsymbol{y}_k = \boldsymbol{A}_{(k)}\boldsymbol{x}_k + \boldsymbol{w}_k, \tag{1.2}$$

for time instants $k = 0, 1, \ldots$. Here, $\boldsymbol{x}_k \in \mathbb{R}^N$ denotes the state vector that characterizes the system. The state is influenced by the input $\boldsymbol{h}_k \in \mathbb{R}^L$; and the output $\boldsymbol{y}_k \in \mathbb{R}^m$ represents the measured response of the system. The output is corrupted by the noise term denoted by $\boldsymbol{w}_k \in \mathbb{R}^m$. Also, $\boldsymbol{D} \in \mathbb{R}^{N \times N}$ is the system transfer matrix, $\boldsymbol{H} \in \mathbb{R}^{N \times L}$ is the

input matrix, and $\boldsymbol{A}_{(k)} \in \mathbb{R}^{m \times N}$ is the observation matrix of the system at time instant k. For example, in an orthogonal frequency division multiplexing (OFDM) wireless system, the state denotes the successive instantiations of a time-varying wireless channel. The temporal correlation of the channel, modeled using a first-order auto-regressive process, is captured by (1.1). Also, (1.2) denotes the linear relation between the received signal, pilot signal, and the channel instantiation.

An important problem associated with a linear dynamical system is the estimation of the system state vectors $\boldsymbol{x}_k$, for $k = 0, 1, \ldots$ using the corresponding measurements $\boldsymbol{y}_k$. This problem is equivalent to solving a system of linear equations. For example, in the context of wireless channel model explained above, this problem translates to the time-varying channel estimation and tracking problem. We recall that, in general, for solving a linear system, the number of measurements should not be less than the number of unknowns. However, if the solution is known to admit a sparse representation in a suitable basis, the number of measurements required can be potentially reduced by exploiting this additional information. The reduction in the number of measurements is advantageous in many real world systems. For instance, in wireless channel estimation, the channel vector is known to be sparse in the lag domain, and thus exploiting sparsity help to reduce the pilot overhead. Hence, the whole thesis is devoted to the mathematics underlying the state estimation of LDS when the state vectors admit sparse representation. Our work is motivated by the results from the area of sparse signal recovery and compressive sensing (CS) literature. These results serve as the point of departure for our work, and we discuss them in the next section.

1.1 Review of Compressed Sensing

The CS framework studies the theory and algorithmic development for finding sparse solutions to underdetermined systems of linear equations [9–11]. The standard CS problem is to reconstruct an $s-$sparse vector $\boldsymbol{x} \in \mathbb{R}^N$ from linear measurements:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^m, m < N. \tag{1.3}$$

There are two aspects to the CS problem: one, design of the linear measurement process, and second, design of a suitable recovery algorithm. These two problems are equally important and delicate, and they form the foundation of the thesis. In the next subsection, we discuss some of the popular and well-understood algorithms and the associated recovery guarantees available in the literature.

1.1.1 Sparse Recovery Algorithms

Some of the existing algorithms for the recovery of a sparse vector are as follows:

- *Basis pursuit*: It is a convex optimization method with polynomial time complexity [9, 12]. It solves the following optimization problem:

$$\arg\min_{\boldsymbol{x}\in\mathbb{R}^N} \|\boldsymbol{x}\|_1 \text{ subject to } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \tag{1.4}$$

  There are other variants of the algorithm such as LASSO, Dantzig selector, ℓ_p norm minimization, etc [13, 14].

- *Thresholding algorithms*: These are iterative algorithms based on carefully designed thresholding functions. Basic thresholding, iterative hard thresholding (IHT) and

hard thresholding pursuit (HTP) are some of the algorithms that belong to this class of algorithms [15].

- *Greedy algorithms*: These algorithms are designed based on making a locally optimal choice of the support of the vector at each stage. Orthogonal matching pursuit (OMP) and compressive sampling matching pursuit (CoSaMP) are examples of greedy algorithms [16–21].

- *Bayesian methods:* These algorithms impose a fictitious Bayesian prior on the unknown vector and solve the sparse recovery problem using probabilistic estimation. Sparse Bayesian learning (SBL) and approximate message passing are some well-received Bayesian methods in literature [22–26].

Clearly, the recovery performance of these algorithms depend on the properties of the measurement matrices. For example, if the number of rows m of the measurement matrix is less than the sparsity s, the recovery is information theoretically impossible. The desired properties of the measurement matrices are discussed next.

1.1.2 Recovery Guarantees

One can look for two types of guarantees for a CS problem:

- *Existence and uniqueness:* When is a solution to a compressed sensing problem guaranteed to exist?

- *Recovery guarantee:* When is the compressed sensing algorithm guaranteed to be recover the sparsest solution?

The first question is relatively easy to answer, as the solution is guaranteed to exist if the union of subspaces spanned by all sets of s columns of $\boldsymbol{A}$ is $\mathbb{R}^m$. However, union of two subspaces is a subspace if and only if one of them is contained in the other one. Thus, the solution is guaranteed to exist if the column space spanned by the any of the s columns of $\boldsymbol{A}$ is $\mathbb{R}^m$. Further, the solution is unique if the null space of $\boldsymbol{A}$ does not contain any $2s-$sparse vector. However, the second question is by far less trivial and has received a lot of research attention. The most popular approach to establishing guarantees for the exact recovery is through the so-called restricted isometry property (RIP) [27,28], defined as follows. A measurement matrix $\boldsymbol{A}$ is said to satisfy the s-RIP with restricted isometry constant (RIC) δ_s if $\delta_s \in (0,1)$, where

$$\delta_s \triangleq \inf\left\{\delta : 1-\delta \leq \|\boldsymbol{A}\boldsymbol{z}\|^2 \leq 1+\delta,\ \forall \|\boldsymbol{z}\| = 1,\ \text{and}\ \|\boldsymbol{z}\|_0 \leq s\right\}. \tag{1.5}$$

Some examples of RIP based guarantees for exact recovery of sparse vectors with ℓ_0 norm at most s are as follows:

- $\delta_s < 1/3$ and $\delta_{2s} < \sqrt{1/2}$, (more generally, $\delta_{ts} < \sqrt{(t-1)/t}$ for $t \geq 4/3$ and $\delta_{ts} < \sqrt{t/(4-t)}$ for $0 < t < 4/3$) are sharp for recovery using basis pursuit [29–31].
- $\delta_{3s} < 1/8$ is sufficient for recovery via the iterative hard thresholding (IHT) algorithm [32].
- $\delta_{s+1} < \frac{1}{\sqrt{s+1}}$ is sufficient for recovery via the orthogonal matching pursuit (OMP) algorithm [33].

The RIP also ensures that the recovery process is robust to additive noise and is stable when the unknown vector is not precisely sparse.

1.1.3 Extensions

The theory and algorithms we discussed so far are appropriate for the recovery of a single sparse vector with no additional side information or constraint. However, in practice, there are LDS related problems for which one has to recover more than one sparse vector and the sparse vectors exhibit additional structural properties. For example, in the wireless channel estimation problem, one has to recover successive instantiations of a sparse time-varying wireless channel. These instantiations have the same power delay profile, and the nonzero coefficients of these instantiations are temporally correlated. Thus, a recovery algorithm exploiting the common support and temporal correlation yields better recovery performance. We list the different types of sparsity models studied in the literature:

- *Block sparsity:* A sparsity pattern in which the non-zero entries occur in multiple clusters [34, 35].
- *Piecewise sparsity:* A sparsity pattern formed by the concatenation of a set of sparse vectors [36].
- *Joint sparsity:* A sparsity pattern in which a set of sparse vectors share the same support [37–41].
- *Temporally correlated joint sparsity:* A model in which vectors exhibit joint sparsity along with temporal correlation of the nonzero entries [3, 42, 43].

In the light of the above discussion, for a given sparse recovery problem, the first step is to identify any structure in addition to sparsity that exists in the signal. Then, one can explore two facets of the problem: development of efficient recovery algorithms and theoretical guarantees on existence of a solution, and recovery performance. Hence, in this

thesis, we build on the theory of sparse signal recovery to address the state estimation problem in LDS, which deals with the above aspects.

With the above background, we next introduce the sparsity models associated with LDS that are investigated in this thesis.

1.2 Sparsity Models in LDS

We consider three versions (denoted by SM1-SM3) of sparse recovery problems for *state estimation in LDS*. The categorization is based on the information available to the recovery algorithm about the inputs and the observation matrix, as we describe next. Here, our goal is to highlight a selection of LDS-related problems that can be reduced to or can be modeled using the CS framework, without explicitly detailing the mathematical model.

1.2.1 SM1: Known inputs and observation matrix

In *version 1 (SM1)* of the problem, the inputs and observation matrix are known, and the goal is to estimate the sequence of system states. This problem is equivalent to the estimation of the initial state, i.e., the state at time zero. Thus, this version is concerned with the question of how well the initial state of a linear dynamical system can be inferred from its observations and inputs. Here, we assume that the initial state of the system is known to admit a sparse representation in a suitable basis. For example, diffusion processes in complex networks that model phenomena like disease or epidemic spreading in human society [7, 8], air or water pollution [5, 6], virus spreading in computer and mobile phone networks [44,45], information propagation in online social networks [46], etc., are known to have a sparse initialization. Identifying the initial state of these processes accurately is a

critical first step towards their control [47]. Thus, a key problem here is the recoverability of the sparse system state, which simplifies to a single measurement model as in (1.3).

1.2.2 SM2: Unknown inputs and known observation matrix

In the next model, *version 2 (SM2)*, the goal is to estimate inputs and the initial state, which is equivalent to the estimation of the state evolution over time. Thus, this version refers to the estimation of a sequence of vectors, which are assumed to be sparse. In other words, we assume that the initial state and the inputs are sparse. A motivating application for such sparse control is a networked control system. The system is comprised of controllers, plants and sensors, connected over a network medium. Due to the limited bandwidth of the physical communication network, the communication in the network only support low data rates [48,49]. In order to reduce the size of data exchanged between controllers and plants, one can use sparse signals as control inputs, because the sparse signals are known to admit compact representations [9–11]. Another motivating real-world problem that can be modeled using sparse inputs is the wireless channel estimation problem described at the beginning of this chapter. For this problem, the inputs could refer to the difference between the consecutive instantiations of the channel, and the goal is to recover the sequence of sparse channel instantiations.

1.2.3 SM3: Unknown inputs and observation matrix

In this sparse recovery problem, *version 3 (SM3),* one needs to learn both the matrix that characterizes LDS and the external inputs that influence the state. In the CS setup, this problem is known as *dictionary learning.* Learning system specific, adaptive measurement matrices are particularly beneficial when the measurement model is not precisely known,

as in the case of an image. The use of adaptive dictionaries often leads to more compact representations and better performance in signal denoising, inpainting, and restoration. This method is known to outperform the traditional approach of using predefined dictionaries like wavelets or union of orthogonal bases like the Fourier and Dirac.

The above three sparsity patterns that arise in LDS are motivated by different real-world applications, and it is interesting to explore the theoretical and algorithmic aspects of these sparsity models. In the next section, we sketch the territory of research presented in this thesis.

1.3 Scope and Contributions of the Thesis

The central research questions that drive our investigation are as follows:

Q1. When is a sparse solution is guaranteed to exist for a given sparsity model?

Q2. If the solution exists, what are some efficient reconstruction algorithms?

Q3. For a given reconstruction algorithm, when is the solution guaranteed to be faithfully recovered?

These questions are not independent, as the recovery guarantees are algorithm dependent, and the reconstruction algorithm assumes the existence of at least one sparse solution. The thesis addresses these fundamental questions for the three versions of sparsity patterns presented in Section 1.2.

The overall thesis organization follows a path from simple to more complicated versions of the problems, i.e., from SM1-SM3. In the following subsections, we give an overview of CS problems considered and the major findings obtained in the thesis. Here, we adopt an

informal style rather than delving into the technical details. We refine these questions for each case separately, mentioning the special structures associated with the measurement matrices or the sparse signals. This helps us connect the LDS problems with the existing CS literature and recognize the gaps in the literature. We then elaborate on the specific contributions of the thesis.

1.3.1 Known inputs and observation matrix (SM1)

As mentioned in the previous section, this version of the problem reduces to the standard CS problem in Section 1.1. Since we do not assume any special structure on the initial state vector, any conventional recovery algorithm can be applied to this version of the problem. However, the theoretical guarantees which depend on the properties of the measurement model require a fresh look, as the LDS model imposes a special structure on the measurement matrices. Due to this structure, classical CS based recovery guarantees do not apply to this problem. The initial state estimation problem in LDS is called *observability problem* in the control theory terminology. To sum up, under the model SM1, we present an in-depth study of question Q3, and establish recovery guarantees under the measurement matrix that arises in the observability problem of an LDS. We discuss this in Chapter 2.

We show that, if the initial state vector admits a sparse representation, the number of measurements can be significantly reduced by using random projections for obtaining the measurements. Our analysis gives sufficient conditions for the RIP of the observability matrix to hold, which leads to guarantees for the observability of the system. These conditions depend only on the properties of system transfer and observation matrices and are derived using tools from probability theory and compressed sensing. Our results are

stronger than the existing results in the regime where they are comparable. These results appear in [50, 51].

1.3.2 Unknown inputs and known observation matrix (SM2)

For this version of the sparsity model, the goal is to estimate a set of sparse vectors, which makes it different from SM1. We address all three questions Q1, Q2 and Q3 for this problem in Chapters 3,4 and 5, respectively. We start with the question Q1 regarding the existence of a set of sparse control inputs which can drive the system to any desired state. This problem is referred to as the *controllability* problem. We revisit the controllability problem taking the sparsity constraints into account. To this end, we first derive necessary and sufficient conditions for ensuring controllability of an LDS with arbitrary transfer matrices. Our characterizations are in terms of algebraic conditions, which require verifying rank conditions on an appropriately defined set of matrices. The number of conditions to be verified grows with the state dimension, but does not depend on the number of input vectors required to steer the system to the desired state. In this way, the results have a similar flavor as the classical results for unconstrained input system, although the proof technique is completely different. Using these conditions, we design a non-combinatorial test to check the controllability of LDS using sparse inputs. Further, we characterize the minimum number of input vectors required to satisfy the derived conditions for controllability. Finally, we present a generalized Kalman decomposition-like procedure that separates the state-space into subspaces corresponding to sparse-controllable and sparse-uncontrollable parts. Our results form a theoretical foundation for designing networked linear control systems with sparse inputs, by introducing and investigating the notion of controllability under sparsity constraints. These results appear in [52].

Next, we address the question Q2 on the recovery of sparse state vectors of LDS for model SM2 in Chapter 4. In this part of the study, we assume that the system transfer matrix is a diagonal matrix. Hence, the problem is to reconstruct temporally sparse vectors sharing a common support, from noisy underdetermined linear measurements. We devise two Bayesian algorithms that sequentially recover the vectors, without waiting for all the measurements to arrive. The online algorithms are formulated using the SBL framework and are implemented using a sequential expectation-maximization procedure combined with Kalman smoothing. The first set of algorithms are iterative in nature, which are then modified to develop noniterative algorithms. Due to the online nature of the algorithm, it requires less computational and memory resources compared to offline processing. We illustrate the efficacy of the algorithms using sparse orthogonal frequency division multiplexing channel estimation through numerical results. These results are published in [53, 54].

Finally, we present the solution to question Q3 for model SM2 in Chapter 5, in the context of algorithms presented in Chapter 4. We analyze the convergence of the algorithms in the special case when the sparse vectors are uncorrelated, using tools from stochastic approximation theory. We show that the sequence of the covariance estimates converges either to the global minimum of the offline equivalent cost function or to the all-zero vector, regardless of the sparsity level of the signal. These results appear in [54].

1.3.3 Unknown inputs and observation matrix (SM3)

In this version, we need to find a decomposition that can explain our measurements and ensure that the control inputs are sparse. This problem is equivalent to a matrix factorization problem, and it is different from the sparse signal recovery problem of solving a

system of linear equations with some constraints or signal structure. Thus, the theoretical analysis demanded by question Q1 is often hard to carry out for this version of the problem. Hence, we restrict our work to the algorithm design and the related recovery guarantees, i.e., we only seek answers to questions Q2 and Q3, for SM3, and our results are presented in Chapter 6.

For this work, we assume that the system transfer matrix is a zero matrix. The joint recovery of the sparse representation and dictionary is formulated using the sparse Bayesian learning framework by imposing a fictitious prior on the sparse vectors. The parameters of the prior on the sparse vectors and the dictionary are simultaneously learned using the expectation-maximization algorithm. The dictionary update step turns out to be a nonconvex problem which is solved using either an alternating minimization (AM) procedure or the Armijo line search (ALS). Next, to address Q3, we show that the algorithm converges, and further analyze the stability of the solution by characterizing its limit points. We also analyze the minima of the overall cost function of the presented algorithm and prove that the desired sparse representation is likely to be achieved by the algorithm. Through numerical results, we demonstrate the efficacy of the presented algorithm and compare it with existing dictionary learning algorithms for the application of image denoising. These results appear in [55].

1.3.4 Anomaly imaging for structural health monitoring

In Chapter 7, we include a different application of structured sparse signal recovery, namely, anomaly imaging for structural health monitoring. Although this chapter does not discuss an LDS-based sparsity model, the ideas of the chapter are aligned with the main theme of sparse signal recovery. This chapter presents a new tomography-based

anomaly mapping algorithm for composite structures. The system consists of an array of piezoelectric transducers which sequentially excites the structure and collects the resulting waveform at the remaining transducers. Anomaly indices computed from the sensor waveforms are fed as input to the mapping algorithm. The output of the algorithm is a color map indicating the outline of damage on the structure when present. Unlike prior work on this topic, the algorithm of this chapter explicitly accounts for both sparsity and cluster pattern structures that are typical of structural anomalies. Hence, our algorithm provides excellent reconstruction accuracy by incorporating the available prior information on the anomaly map. Experimental results on a unidirectional composite plate confirm that the algorithm outperforms two competing existing methods in terms of reconstruction accuracy. These results appear in [56].

We offer some concluding remarks and questions for further study in Chapter 8. The appendices containing supplementary material, namely, Appendix A for Chapter 2, Appendix B for Chapter 3, and Appendix C for Chapter 5 are included at the end of the thesis.

On the whole, the thesis presents three different sparsity models related to LDS which are of practical relevance. We develop rigorous recovery results for the three models, answering some fundamental questions on existence and recoverability of the solution.

1.4 List of Publications from this Thesis

Journal Articles

J1 **G. Joseph**, and C. R. Murthy, "A noniterative online Bayesian algorithm for the recovery of temporally correlated sparse vectors," *IEEE Transactions on Signal Processing*, vol. 65, no. 20, pp. 5510–5525, Oct. 2017.

J2 **G. Joseph**, and C. R. Murthy, "On the observability of a linear system with a sparse initial state," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 994–998, Jul. 2018.

J3 **G. Joseph**, and C. R. Murthy, "Measurement bounds for observability of linear dynamical systems under sparsity constraints," *IEEE Transactions on Signal Processing*, vol. 67, no. 8, pp. 1992–2006, Apr. 2019.

J4 **G. Joseph**, and C. R. Murthy, "Sparse-controllability of linear dynamical systems," *submitted to IEEE Transactions on Automatic Control.*

J5 **G. Joseph**, and C. R. Murthy, "A Bayesian algorithm for joint dictionary learning and sparse signal recovery," *submitted to IEEE Transactions on Signal Processing.*

Conference Proceedings

C1 **G. Joseph**, C. R. Murthy, R. Prasad, and B.D.Rao, "Online recovery of temporally correlated sparse signals using multiple measurement vectors," *IEEE Global Communications Conference*, San Diego, USA, Dec. 2015.

C2 **G. Joseph**, A. B. Zoubi, C. R. Murthy, and V. J. Mathews, "Anomaly imaging for structural health monitoring exploiting clustered sparsity," *IEEE International*

Conference on Acoustics, Speech and Signal Processing, Brighton, UK, May, 2019.

Chapter 2

Observability of Sparse Initial State

Answering problem Q3 for SM1

In this chapter, we look at the LDS sparsity model with only initial state unknown, and all the control inputs and the system matrices known (model SM1). The unknown initial state is assumed to sparse in a suitable basis. For this setting, we consider two cases: one, the observation matrices are independent random matrices, and two, they are identical to a single random matrix. We derive an upper bound on the number of measurements required for recovering the sparse initial state using classical compressive sensing algorithms. The results are probabilistic in nature and depend on the properties of the system matrices. Also, the bound is more general and stronger than the existing results in the regime where they are comparable.

2.1 Background

Observability is an important notion in control theory. It is concerned with the question of how well the state of a linear dynamical system can be inferred from its observations

and inputs [57]. The classical observability problem involves solving a linear system of equations:

$$\widetilde{\boldsymbol{y}}_{(K)} = \widetilde{\boldsymbol{A}}_{(K)}\boldsymbol{x}_0, \tag{2.1}$$

where the measurement vector $\widetilde{\boldsymbol{y}}_{(K)}$ and the observability matrix $\widetilde{\boldsymbol{A}}_{(K)}$ are known, and we need to estimate $\boldsymbol{x}_0$ exactly.[1] The standard results from linear algebra state that a discrete time system is observable if the rank of the observability matrix $\widetilde{\boldsymbol{A}}_{(K)}$ equals the system dimension [58]. This result applies to the general formulation of the problem, and hence, a large number of measurements are required to recover the initial state for systems with a high dimensional state [59–61]. However, if the initial state of the system is known to admit a sparse representation in a suitable basis, the number of measurements required can be potentially reduced by exploiting this additional information. As we mentioned in Chapter 1, diffusion processes in complex networks that model phenomena like disease or epidemic spreading in the human society [7,8], air or water pollution [5,6], virus spreading in computer and mobile phone networks [44, 45], information propagation in online social networks [46], etc., are known to have a sparse initialization. Identifying the initial state of these processes accurately is a critical first step towards their control [47]. Thus, a key problem in this context is the recoverability of the sparse system state using as few measurements as possible. Further, in some cases, the measurements are obtained as random linear projections of the system state. For example, in the problem of finding the source of pollution in a water body or in the atmosphere, measurements collected from sensors placed at spatially random locations can be mathematically modeled as random linear projections of the system state [62–65]. Hence, in this chapter, we provide guarantees

[1] We discuss the system model in detail in Section 2.2.

on the observability of a system when the observability matrix is random and possibly rank deficient, and the initial state admits a sparse representation. We establish these guarantees by analyzing the RIP of the *structured random observability matrix* arising out of a linear dynamical system.

2.1.1 Related Work

Our work focuses on two aspects: first, we explore the connection between compressed sensing and observability of the state of a linear dynamical system, and second, we derive sufficient conditions for state recovery by analyzing the RIP of the observability matrix. In our case, the observability matrix is a random matrix with a special structure. Hence, the existing results from the compressed sensing literature cannot be directly applied to our problem. In this subsection, we provide review the past literature in this direction.

Compressed sensing and observability

The connection between the compressed sensing and linear dynamical systems is a nascent topic, and has only recently been studied in the literature. The design of control algorithms based on sparsity in the state using tools from compressed sensing is presented in [66]. However, this paper does not discuss guarantees for recoverability of the system state in the presented framework. On the other hand, [67,68] assert that a linear dynamical system is observable if the observability matrix satisfies the RIP. However, conditions under which the observability matrix satisfies RIP are not discussed.

The results in [62–64] characterize the number of measurements required for the exact recovery of the initial state in a stochastic setting. However, the results are useful only under somewhat overly restrictive conditions such as the system transfer matrix being

unitary, the observation matrices being i.i.d. Gaussian, and the initial state being sparse in the canonical basis. Moreover, those results depend on the smallest singular value of the transfer matrix. As a consequence, they are not independent of scaling of the transfer matrix. In this chapter, we derive more general results on the observability of LDS under sparsity constraints, which are independent of the scaling of the matrices.

RIP of structured random matrices

We list a few types of structured random matrices which have been shown to satisfy the RIP in the literature:

- Subsampled bounded orthonormal systems [69, 70]
- Partial random circulant matrices and partial random Toeplitz matrices [71–73]
- Block diagonal measurement matrices where each block on the main diagonal is a subgaussian random matrix [74]
- The columnwise Khatri-Rao product of two matrices [75].

As we will see, the RIP of the structured random observability matrix that arises in our problem has not been studied in the past. Hence, it requires new analysis using tools from non-asymptotic random matrix theory.

In this chapter, we first derive guarantees on the system observability under a stochastic setting when the observation matrices are i.i.d. subgaussian random matrices and the system transfer matrix is nonzero. However, in many applications, due to hardware constraints, the measurement process could involve linear projection using a single, randomly selected matrix, rather than an independent matrix for each measurement instant. Hence,

it is more pertinent to derive recovery guarantees for the case when the observation matrix is fixed, but equal to an instantiation of a random matrix. We present a different, new analysis to obtain guarantees for uniform recovery of the state for the identical observation matrices cases. We also study the problem of joint recovery of the initial state and sparse input vectors. The key novelty in the results is the derivation of sufficient conditions on K and m required for the recovery of sparse initialization and inputs. The results presented here are of independent interest, since they provide insights to the RIP and NSP of the matrices with similar structure. In summary, we show that systems that are unobservable using classical control theory can be observable under the sparsity constraints.

2.2 System Model

We consider discrete-time linear system which is modeled as follows:

$$\boldsymbol{x}_k = \boldsymbol{D}\boldsymbol{x}_{k-1} \tag{2.2}$$

$$\boldsymbol{y}_k = \boldsymbol{A}_{(k)}\boldsymbol{x}_k, \tag{2.3}$$

for discrete time instants $k = 0, 1, \ldots, K-1$. Here, $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is a nonzero system transfer matrix and $\boldsymbol{A}_{(k)} \in \mathbb{R}^{m\times N}, m \ll N$ is the observation matrix of the system at time instant k. We are interested in the observability of the system when the initial state is sparse. We make the following points before proceeding further:

(a) Observability of the initial sparse state $\boldsymbol{x}_0$ implies the observability of $\boldsymbol{x}_k$ for all k.

(b) In (2.2), we do not include an innovation term as we did in (1.1). Since we are considering the problem of system observability, the system input is assumed to be

known. We can therefore simply subtract its effect from the system evolution as well as observation equations, resulting in the system model given by (2.2) and (2.3). We consider the joint recovery of the initial state and sparse innovation terms in Section 2.6.

(c) The system equations do not consider measurement noise or model mismatch. However, in the presence of these impairments, our results can be extended to robust recovery of the initial state; we discuss this in Section 2.4.4.

In view of the above, we formally define the notion of observability as follows:

Definition 2.1 (Observability)**.** *A system is said to be observable if any unknown s–sparse initial state $\boldsymbol{x}_0$ can be determined uniquely from the outputs $\{\boldsymbol{y}_k\}_{k=0}^{K-1}$, the transfer matrix $\boldsymbol{D}$, and the observation matrices $\left\{\boldsymbol{A}_{(k)}\right\}_{k=0}^{K-1}$.*

To recover the sparse initial vector, we consider the following equivalent linear system at time K:

$$\widetilde{\boldsymbol{y}}_{(K)} = \widetilde{\boldsymbol{A}}_{(K)}\boldsymbol{x}_0, \tag{2.4}$$

where the measurement vector $\widetilde{\boldsymbol{y}}_{(K)} \in \mathbb{R}^{Km}$ and the observability matrix $\widetilde{\boldsymbol{A}}_{(K)} \in \mathbb{R}^{Km\times N}$ are defined as

$$\widetilde{\boldsymbol{y}}_{(K)} = \begin{bmatrix} \boldsymbol{y}_0 \\ \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{K-1} \end{bmatrix}, \quad \widetilde{\boldsymbol{A}}_{(K)} = \begin{bmatrix} \boldsymbol{A}_{(0)} \\ \boldsymbol{A}_{(1)}\boldsymbol{D} \\ \vdots \\ \boldsymbol{A}_{(K-1)}\boldsymbol{D}^{K-1} \end{bmatrix}. \tag{2.5}$$

In order to ensure the recovery of $\boldsymbol{x}_0$ from (2.4) using sparse signal recovery techniques, we need to analyze the RIP of the observability matrix $\widetilde{\boldsymbol{A}}_{(K)}$. This, in turn, yields bounds on the number of measurement vectors required to recover any sparse initial state. Before

launching into the RIP analysis, we note that an overall scaling does not affect the RIP of a matrix. Now, let $\lambda_{\max} \neq 0$ be the largest singular value of $\boldsymbol{D}$. We can rewrite (2.4) as

$$\widetilde{\boldsymbol{L}}(\boldsymbol{\lambda}_{\max})\widetilde{\boldsymbol{y}}_{(K)} = \widetilde{\boldsymbol{L}}(\boldsymbol{\lambda}_{\max})\widetilde{\boldsymbol{A}}_{(K)}\boldsymbol{x}_0, \tag{2.6}$$

where $\boldsymbol{\lambda}_{\max} = \begin{bmatrix} 1 & \lambda_{\max} & \dots & \lambda_{\max}^{K-1} \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^K$ and the matrix function $\widetilde{\boldsymbol{L}}(\boldsymbol{\lambda}) : \mathbb{R}^K \to \mathbb{R}^{Km \times Km}$ is defined as

$$\widetilde{\boldsymbol{L}}(\boldsymbol{\lambda}) = \frac{1}{\sqrt{Km}} \begin{bmatrix} \boldsymbol{\lambda}_1 \boldsymbol{I} & & & \\ & \boldsymbol{\lambda}_2 \boldsymbol{I} & & \\ & & \ddots & \\ & & & \boldsymbol{\lambda}_K \boldsymbol{I} \end{bmatrix}^{-1}. \tag{2.7}$$

Therefore, we get the following relation:

$$\widetilde{\boldsymbol{L}}(\boldsymbol{\lambda}_{\max})\widetilde{\boldsymbol{A}}_{(K)} = \begin{bmatrix} \boldsymbol{A}_{(0)} \\ \boldsymbol{A}_{(1)}\bar{\boldsymbol{D}} \\ \dots \\ \boldsymbol{A}_{(K-1)}\bar{\boldsymbol{D}}^{K-1} \end{bmatrix}, \tag{2.8}$$

where $\bar{\boldsymbol{D}} = \boldsymbol{D}/\lambda_{\max}$ has the largest and the smallest singular values as 1 and λ, respectively. Here, λ is the ratio of the smallest to the largest singular value of $\boldsymbol{D}$. Analyzing the recoverability of $\boldsymbol{x}_0$ from (2.6), which is equivalent to (2.4), requires one to study the RIP of the matrix $\widetilde{\boldsymbol{L}}(\boldsymbol{\lambda}_{\max})\widetilde{\boldsymbol{A}}_{(K)}$. Therefore, in this chapter, we focus on the RIP of such a scaled version of $\widetilde{\boldsymbol{A}}_{(K)}$.

2.3 Preliminaries

In this section, we define a subgaussian random matrix and summarize some of its properties.

Definition 2.2 (Subgaussian random variable)**.** *A random variable A is said to be subgaussian with parameter c if, for any $\theta \in \mathbb{R}$,*

$$\mathbb{E}\left\{\exp\left(\theta A\right)\right\} \leq \exp\left(c\theta^2\right). \tag{2.9}$$

Definition 2.3 (Subgaussian random matrix)**.** *A random matrix $\boldsymbol{A} \in \mathbb{R}^{m\times N}$ is said to be a subgaussian random matrix if its entries are independent zero mean and unit variance subgaussian random variables with common parameter c.*

The subgaussian random matrix includes a large class of random matrices including independent and identically distributed (i.i.d.) Gaussian random matrices, and i.i.d. Bernoulli random matrices, etc. Next, we present two results that are necessary for the derivation of the main results in the chapter.

Lemma 2.1. *If A is a subgaussian random variable with parameter c, then $A^2 - \mathbb{E}\left\{A^2\right\}$ is a subexponential random variable with parameter $16c$, i.e., for $|\theta| \leq \frac{1}{16c}$, we have*

$$\mathbb{E}\left\{\exp\left[\theta\left(A^2 - \mathbb{E}\left\{A^2\right\}\right)\right]\right\} \leq \exp\left(128\theta^2c^2\right). \tag{2.10}$$

Proof. See [76, Lemma 1.12]. □

Proposition 2.1 (Bernstein-type inequality)**.** *Let* $\{A_l\}_{l=1,2,\dots,m}$ *be independent subexponential random variables such that* $a_{\min} \leq \mathbb{E}\{A_l\} \leq a_{\max}$*. That is, for all* $t \geq 0$*,*

$$\mathbb{P}\{|A_l - \mathbb{E}\{A_l\}| \geq t\} \leq c_1 \exp(-c_2 t), \tag{2.11}$$

for $l = 1, 2, \dots, m$*, and some constants* $c_1, c_2 > 0$*. Then, for any* $t > m \max\{a_{\max}, -a_{\min}\}$*,*

$$\mathbb{P}\left\{\left|\sum_{l=1}^{m} A_l\right| \geq t\right\} \leq \exp\left(-\frac{c_2^2(t - ma_{\max})^2/2}{m(2c_1 + c_2 a_{\max}) + c_2 t}\right) + \exp\left(-\frac{c_2^2(t + ma_{\min})^2/2}{m(2c_1 + c_2 a_{\min}) + c_2 t}\right). \tag{2.12}$$

Proof. See Appendix A.1. □

2.4 RIP For Independent Observation Matrices

In this section, we present the first main result of the chapter and its implications.

Theorem 2.1 (Independent random observation matrices)**.** *Suppose measurement matrices* $\boldsymbol{A}_{(k)}, k = 0, 1, \dots, K-1$ *are independent subgaussian random matrices with parameter* c*. Then, if*

$$Km\left(\delta - 1 + \lambda^{2(K-1)}\right)^2 \geq \widetilde{c}\left[9s \ln\left(\frac{eN}{s}\right) + 2\ln\left(\frac{2}{\epsilon}\right)\right], \tag{2.13}$$

the RIC δ_s *of the scaled version* $\widetilde{\boldsymbol{L}}(\boldsymbol{\lambda}_{\max})\widetilde{\boldsymbol{A}}_{(K)}$ *of* $\widetilde{\boldsymbol{A}}_{(K)}$ *satisfies* $\delta_s < \delta$ *for all* $1 - \lambda^{2(K-1)} < \delta < 1$ *with probability at least* $1 - \epsilon$*. Here,* $\widetilde{c}$ *is a constant dependent only on* c*, and* $\lambda \leq 1$ *is the ratio of the smallest to the largest singular values of* $\boldsymbol{D}$*. When* (2.13) *holds, the system is observable for sufficiently large* λ *with high probability.*

Proof. See Appendix A.2. □

Here, we note that the phrase "sufficiently large" λ refers to the fact that it must be

large enough so as to be able to meet the upper bound on the RIC set by the RIP based guarantees of different algorithms, as discussed in Section 1.1. We discuss this point in detail in Section 2.4.2.

Next, using the proof technique of the above theorem, we can show the following interesting corollaries. We omit their proofs as they are straightforward. The first corollary extends Theorem 2.1 to the case when the measurements are not necessarily taken over a contiguous set of time instants.

Corollary 2.1 (Non-consecutive measurements)**.** *Suppose the available measurements are* $\{\boldsymbol{y}_k\}_{k\in\mathcal{K}}$*, where* $\mathcal{K}$ *is an index set of cardinality* K*, and* $\boldsymbol{A}_{(k)}, k = 0, 1, \ldots, K-1$ *are independent subgaussian random matrices with parameter* c*. Then, if*

$$Km\left(\delta - 1 + \lambda^{2(\max\{\mathcal{K}\}-1)}\right)^2 \geq \widetilde{c}\left[9s\ln\left(\frac{eN}{s}\right) + 2\ln\left(\frac{2}{\epsilon}\right)\right], \tag{2.14}$$

then the RIC δ_s *of the scaled version* $\widetilde{\boldsymbol{L}}\left(\boldsymbol{\lambda}_{\max,\mathcal{K}}\right)\widetilde{\boldsymbol{A}}_{(K)}$ *of* $\widetilde{\boldsymbol{A}}_{(K)}$ *satisfies* $\delta_s < \delta$ *for all* $1 - \lambda^{2\max\{\mathcal{K}\}} < \delta < 1$ *with probability at least* $1 - \epsilon$*. Here,* $\widetilde{c}$ *is a constant dependent only on* c*, and* $\lambda \leq 1$ *is the ratio of the smallest to the largest singular values of* $\boldsymbol{D}$*. Also,* $\boldsymbol{\lambda}_{\max,\mathcal{K}} \in \mathbb{R}^K$ *has* j^{th} *entry as* $\lambda_{\max}^{\tilde{j}}$*. Hence, when* (2.15) *holds, the system is observable for sufficiently large* λ *with high probability.*

The next corollary extends Theorem 2.1 to the case when $\boldsymbol{x}_0$ is sparse under an arbitrary basis $\Psi \in \mathbb{R}^{N\times N}$ rather than the canonical basis.

Corollary 2.2 (Sparsifying basis other than the canonical basis)**.** *Suppose the observation matrices* $\boldsymbol{A}_{(k)}, k = 0, 1, \ldots, K-1$ *are independent subgaussian random matrices with parameter* c*, and the initial state is sparse under the basis* $\Psi \in \mathbb{R}^{N\times N}$*, which need not be*

the canonical basis. Then, if

$$Km\left(\delta - 1 + \widetilde{\lambda}^2\right)^2 \geq \widetilde{c}\left[9s\ln\left(\frac{eN}{s}\right) + 2\ln\left(\frac{2}{\epsilon}\right)\right], \tag{2.15}$$

the RIC δ_s of the scaled version $\widetilde{\boldsymbol{L}}\left(\widetilde{\boldsymbol{\lambda}}_{\max}\right)\widetilde{\boldsymbol{A}}_{(K)}$ of $\widetilde{\boldsymbol{A}}_{(K)}$ satisfies $\delta_s < \delta$, for all $1 - \widetilde{\lambda}^2 < \delta < 1$, with probability at least $1 - \epsilon$. Here, $\widetilde{c}$ is a constant dependent only on c, and $\widetilde{\lambda} \leq 1$ is the ratio of the smallest to the largest singular value of $\boldsymbol{D}^{(K-1)}\Psi$. Also, $\widetilde{\boldsymbol{\lambda}}_{\max} \in \mathbb{R}^K$ has j^{th} entry as the largest singular value of $\boldsymbol{D}^{j-1}\boldsymbol{\Psi}$. Hence, when (2.15) *holds, the system is observable for sufficiently large $\widetilde{\lambda}$ with high probability.*

It is also interesting to consider guarantees for the case where the matrix $\boldsymbol{D}$ is an RIP-compliant matrix. The following corollary gives guarantees similar to Theorem 2.3 based on the RIC of an RIP-compliant $N \times N$ matrix $\boldsymbol{D}$.

Corollary 2.3 (Relaxation based on the RIP of the transfer matrix)**.** *Suppose $\boldsymbol{A}_{(k)}, k = 0, 1, \ldots, K-1$ are independent subgaussian random matrices with parameter c. Then, if*

$$Km\left(\delta - 1 + \bar{\lambda}^{2(K-2)}(1 - \tilde{\delta}_s)^2\right)^2 \geq \widetilde{c}\left[9s\ln\left(\frac{eN}{s}\right) + 2\ln\left(\frac{2}{\epsilon}\right)\right], \tag{2.16}$$

then the RIC δ_s of the scaled version $\widetilde{\boldsymbol{L}}(\boldsymbol{\lambda}_{\max})\tilde{\boldsymbol{A}}_{(K)}$ of $\widetilde{\boldsymbol{A}}_{(K)}$ satisfies $\delta_s < \delta$ for all $1 - \bar{\lambda}^{2(K-2)}(1 - \tilde{\delta}_s)^2 < \delta < 1$ with probability at least $1 - \epsilon$. Here, $\widetilde{c}$ is a constant dependent only on c, and $\bar{\lambda} < 1$ and $\tilde{\delta}_s \leq 1$ are the smallest nonzero singular value and the RIC of $\boldsymbol{D}$ normalized to unit spectral norm. Hence, when (2.16) *holds, the system is observable for sufficiently small $\tilde{\delta}_s$ with high probability.*

Proof. When the matrix $\boldsymbol{D}$ is normalized to unit spectral norm, for any unit norm $s-$sparse

vector $\boldsymbol{z} \in \mathbb{R}^N$, we have

$$\left\|\boldsymbol{D}^k \boldsymbol{z}\right\| \geq \bar{\lambda}^{k-1} \left\|\boldsymbol{D}\boldsymbol{z}\right\| \geq \bar{\lambda}^{k-1}(1-\tilde{\delta}_s), \tag{2.17}$$

since the vector $\boldsymbol{D}\boldsymbol{z}$ belongs to the column space of $\boldsymbol{D}$. Thus, we can replace $\lambda^{(K-1)}$ with $\bar{\lambda}^{(K-2)}(1-\tilde{\delta}_s)$ in Theorem 2.3 to obtain the desired result. □

We note that $\bar{\lambda} \geq \lambda$ and $1-\tilde{\delta}_s \geq \lambda$, and thus the above corollary is a stronger result than Theorem 2.1. However, λ is easier to compute than the RIC constant of $\boldsymbol{D}$. In the following subsections, we discuss some implications of the above results.

2.4.1 Special Cases

1. Suppose $\boldsymbol{D}$ is a scaled unitary matrix. Then, $\lambda = 1$, and Theorem 2.1 simplifies to the recovery condition for the standard compressed sensing problem with Km measurements. Since the RIP of a matrix is invariant to multiplication by a unitary matrix, each new observation vector adds m new measurements to (2.4) as K increases.

2. Suppose $\boldsymbol{D}$ is rank-deficient. Then, $\lambda = 0$, and (2.13) does not hold for any $\delta < 1$, unless the following holds:

$$m \geq \frac{\widetilde{c}}{\delta^2}\left[9s\ln\left(\frac{eN}{s}\right) + 2\ln(2\epsilon^{-1})\right]. \tag{2.18}$$

This is intuitive, because when $\boldsymbol{x}_0$ lies in the null space of $\boldsymbol{D}$, $\boldsymbol{y}_k = 0$ for $k \geq 1$. Hence, the system is observable if it is observable from $\boldsymbol{y}_0$. Thus, the uniform recovery guarantee does not hold for a rank deficient $\boldsymbol{D}$.

3. Suppose that $\boldsymbol{D}$ is an ill-conditioned matrix, i.e., λ is close to zero. Then, the upper bound on δ required to guarantee observability may not hold [29,77,78]. This is because right multiplication of a matrix by another ill-conditioned matrix may severely degrade its RIP. However, Corollary 2.3 guarantees that if $\tilde{\delta}_s \neq 0$, it is possible to recover $\boldsymbol{x}_0$ even if (2.18) is not satisfied.

4. For $K = 1$, Theorem 2.1 reduces to the recovery condition of the standard compressed sensing problem [32]. Also, if the system is observable with m measurements (for example, when (2.18) is satisfied), the conditions in Theorem 2.1 hold for $K = 1$, as expected.

2.4.2 Number of Measurements

Theorem 2.1 shows that $Km = \mathcal{O}(s \ln(N/s))$ is sufficient for observability. Note that the number of measurements are independent of the scaling of $\boldsymbol{D}$. Thus, the number of measurements can be greatly reduced for large dimensional systems. In contrast, $Km = \mathcal{O}(N)$ measurements are necessary for observability of a general non-sparse initial state vector. We also recall from Section 1.1 that the initial state can be recovered using any of the compressed sensing techniques like basis pursuit, thresholding algorithms, or greedy algorithms.

The RIP based recovery guarantees available in the literature set an upper bound on the RIC. For example, using the necessary and sufficient condition for ℓ_1 based recovery: $\delta_s \leq 1/3$ [29], (2.13) reduces to

$$K\left(\lambda^{2(K-1)} - 2/3\right)^2 \geq \frac{\widetilde{c}}{m}\left[9s \ln\left(\frac{eN}{s}\right) + 2\ln(2\epsilon^{-1})\right], \tag{2.19}$$

for $\lambda^{2(K-1)} \geq 2/3$. In other words, if (2.19) is satisfied for some K which is less than $\lfloor(\ln(2/3))/(2\ln(\lambda))\rfloor + 1$, then the system is observable. However, note that, if the system is observable for K_1 measurements, it remains observable for $K > K_1$.

We note that $K\left(\lambda^{2(K-1)} - 2/3\right)^2$ is an increasing function of K, which gives a lower bound m from (2.19). Therefore, for $\lambda < 1$,

$$m = \mathcal{O}\left(\frac{\ln(N/s)}{K\left(\lambda^{2(K-1)} - 2/3\right)^2}\right). \tag{2.20}$$

We also note that value of m required decreases with λ and K. This is in agreement with the fact that as K increases, we get more measurements and a smaller m suffices for ensuring successful recovery of the initial state. Also, as λ increases, the matrix $\boldsymbol{D}$ becomes better conditioned, and, consequently, a smaller value of m is sufficient for exact recovery.

2.4.3 RIP of the Product of Matrices

We can derive an sufficient condition for the product of a subgaussian matrix and a deterministic matrix to satisfy the RIP property as follows:

Corollary 2.4. *Suppose $\boldsymbol{A} \in \mathbb{R}^{m\times N}$ is subgaussian random matrix with parameter c. If*

$$m\left(\delta - 1 + \lambda^2\right)^2 \geq \widetilde{c}\left[9s\ln\left(\frac{eN}{s}\right) + 2\ln\left(\frac{2}{\epsilon}\right)\right], \tag{2.21}$$

the RIC δ_s of a suitably scaled version of $\boldsymbol{AD}$ satisfies $\delta_s < \delta$, for all $1 - \lambda^2 < \delta < 1$, with probability at least $1 - \epsilon$. Here, $\widetilde{c}$ is a constant dependent only on c, and $\lambda \leq 1$ is the ratio of the smallest to the largest singular values of $\boldsymbol{D}$.

Corollary 2.4 is an immediate by-product of the proof of Theorem 2.1, but it is an

interesting and potentially useful result in its own right, as it provides conditions under which right-multiplication of a subgaussian random matrix by a deterministic matrix $\boldsymbol{D}$ preserves its RIP.

2.4.4 Extension to Robust Recovery

The RIP based analysis allows us to extend Theorem 2.1 to bound the ℓ_1 and ℓ_2 norm error in recovery of the initial state under bounded noise and model mismatch. These impairments correspond to the cases when the measurements are noisy and the initial state is not exactly sparse, respectively. In this case, the system model modifies as follows:

$$\boldsymbol{x}_k = \boldsymbol{D}^k(\boldsymbol{x}_0 + \check{\boldsymbol{x}}_0) \tag{2.22}$$

$$\boldsymbol{y}_k = \boldsymbol{A}_{(k)}\boldsymbol{x}_k + \boldsymbol{w}_k, \tag{2.23}$$

for discrete time instants $k = 0, 1, \ldots, K-1$. Here, $\boldsymbol{w}_k \in \mathbb{R}^m$ denotes the bounded measurement noise: $\|\boldsymbol{w}_k\| \leq W$; while $\check{\boldsymbol{x}}_0 \in \mathbb{R}^N$ represents the error in approximating the initial state by an $s-$sparse vector. That is, $\boldsymbol{x}_0 = \underset{\boldsymbol{v}\in\mathbb{R}^N:\|\boldsymbol{v}\|_0\leq s}{\arg\min} \|\boldsymbol{x}_0 + \check{\boldsymbol{x}}_0 - \boldsymbol{v}\|$. Therefore, the overall set of equations can be written as

$$\tilde{\boldsymbol{y}}_{(K)} = \tilde{\boldsymbol{A}}_{(K)}(\boldsymbol{x}_0 + \check{\boldsymbol{x}}_0) + \tilde{\boldsymbol{w}}, \tag{2.24}$$

where the bounded noise $\tilde{\boldsymbol{w}} \in \mathbb{R}^{Km}$ satisfies $\|\tilde{\boldsymbol{w}}\| \leq \sqrt{K}W$.

Corollary 2.5. *Suppose $\boldsymbol{A}_{(k)}, k = 0, 1, \ldots, K-1$ are independent subgaussian random matrices with parameter c. Suppose that, for some integer $p > 0$ and positive number c_{th},*

$$Km\left(c_{th} - 1 + \lambda^{2(K-1)}\right)^2 \geq \widetilde{c}\left[9ps\ln\left(\frac{eN}{ps}\right) + 2\ln\left(\frac{2}{\epsilon}\right)\right], \tag{2.25}$$

and $\lambda^{2(K-1)} > 1 - c_{th}$. *Here,* $\widetilde{c}$ *is a constant dependent only on* c, *and* $\lambda \leq 1$ *is the ratio of the smallest to the largest singular values of* $\boldsymbol{D}$. *Then, with probability at least* $1 - \epsilon$, *the initial vector* $\boldsymbol{x}_0 + \check{\boldsymbol{x}}_0$ *can be recovered from* (2.24) *with errors as follows:*

$$\left\|\boldsymbol{x}_0 + \check{\boldsymbol{x}}_0 - \hat{\boldsymbol{x}}_0\right\|_1 \leq c_1 \left\|\check{\boldsymbol{x}}_0\right\|_1 + c_2 \sqrt{\frac{s(1-\lambda_{\max}^{-2K})}{Km(1-\lambda_{\max}^{-2})}} W \tag{2.26}$$

$$\left\|\boldsymbol{x}_0 + \check{\boldsymbol{x}}_0 - \hat{\boldsymbol{x}}_0\right\| \leq \frac{c_1}{\sqrt{s}} \left\|\check{\boldsymbol{x}}_0\right\|_1 + c_2 \sqrt{\frac{(1-\lambda_{\max}^{-2K})}{Km(1-\lambda_{\max}^{-2})}} W, \tag{2.27}$$

where $\hat{\boldsymbol{x}}_0$ *is the estimate of the initial vector, and* $c_1, c_2 > 0$ *are universal constants. The constants* p *and* c_{th} *depend on the recovery algorithms as follows:*

- *For BP:* $p = 2$ *and* $c_{th} = \frac{4}{\sqrt{41}}$.
- *For IHT:* $p = 6$ *and* $c_{th} = \frac{1}{\sqrt{3}}$.
- *For compressive sampling matched pursuit (CoSAMP):* $p = 8$ *and* $c_{th} = \frac{\sqrt{\sqrt{11/3}-1}}{2}$.

Proof. Follows from the upper bound on the RIC required by the different algorithms to ensure robust recovery [32, Theorem 6.12, 6.21, 6.28]. □

We note the dependence on $\lambda_{\max}$ in the above expressions is not unexpected: it arises because of the scaling of the measurement matrix. The scaling operation is reasonable due to the following reasons:

- One can always scale the linear equations with no information loss. The scaling operation neither changes the problem nor affects any intuitive notion of SNR.
- The scaling matrix is diagonal, and therefore does not introduce any correlation between the noise terms which might affect the recovery. Moreover, the recovery

guarantees of the algorithms listed in Corollary 2.5 depend only on the ℓ_2 norm of the noise vector, and are independent of the individual variances of the noise terms.

- Note that $\lambda_{\max}$ determines the effective SNR of the system, and hence it plays an important role in recoverability of the initial state. The effect of $\lambda_{\max}$ appears as the factor $\sqrt{s}\sqrt{\frac{(1-\lambda_{\max}^{-2K})}{K(1-\lambda_{\max}^{-2})}}\frac{W}{\sqrt{m}}$ in (2.26). Here, $\sqrt{s}$ and $W/\sqrt{m}$ capture the same effect as those of the sparsity s and the average noise power per measurement $W/\sqrt{m}$, respectively, in the standard compressed sensing results. Further, we intuitively examine the term $\sqrt{\frac{(1-\lambda_{\max}^{-2K})}{K(1-\lambda_{\max}^{-2})}}$ via three special cases of $\lambda_{\max}$ below:

  (i) $\lambda_{\max} \gg 1$: When $\lambda_{\max}$ is large, this term reduces to $1/\sqrt{K}$, which has no dependence on $\lambda_{\max}$. This is because the effective SNR is large, and hence the noise term is negligible, for all measurements except for the first measurement vector, $\boldsymbol{y}_0$. Thus, we have one noisy and $K-1$ noiseless measurements, which leads to an error bound that decreases with K.

  (ii) $\lambda_{\max} \approx 1$: When $\lambda_{\max}$ is close to 1, this term reduces to 1. This is equivalent to having K noisy measurements with equal scaling factor and thus the error bound per measurement is independent of K. In this case, the advantage of having multiple observations comes in terms of the Km dependence of the number of measurements in (2.25).

  (iii) $\lambda_{\max} \ll 1$: When $\lambda_{\max}$ is small, this term reduces to $\lambda_{\max}^{-(K-1)}/\sqrt{K}$, which is a new dependence. In this case, the noise in the later measurements gets amplified by the scaling factor. Hence, the noise term in the last measurement dominates the average noise power. However, in practice, one would consider the smallest

value of K for which (2.25) is satisfied, and substitute that value of K in (2.26) and (2.27) to get the bound on robust recovery of the initial state.

2.4.5 Comparison With Prior Work

In [63,64], the authors address the same problem as ours and give a sufficient condition on number of measurements Km for successful recovery. In this subsection, we compare and contrast the two results. We begin with the result from [63,64], stated in our notation.

Theorem 2.2 (Prior work [63,64])**.** *Suppose that* $\boldsymbol{D} = a\boldsymbol{U}$ *where* $a \neq 0$ *and* $\boldsymbol{U} \in \mathbb{R}^{N\times N}$ *is unitary. Define* $b \triangleq \sum_{k=1}^{K} a^{2(k-1)}$*. Assume* $\boldsymbol{A}_{(k)}, k = 0, 1, \ldots, K-1$ *are independent Gaussian random matrices with mean zero and variance* $1/m$*. Then, if*

$$Km\delta^2 \geq 512 \left[s \ln\left(\frac{42}{\delta}\right) + 1 + \ln\left(\frac{N}{s}\right) + \ln\left(\frac{2}{\epsilon}\right) \right] \left[\frac{\|1-a^2|\, K + \min\{1, a^2\}}{\max\{1, a^2\}} \right], \quad (2.28)$$

the RIC δ_s *of* $\frac{1}{\sqrt{b}}\tilde{\boldsymbol{A}}_{(K)}$ *satisfies* $\delta_s < \delta < 1$ *with probability at least* $1-\epsilon$*.*

We make the following observations:

- *Restriction on* $\boldsymbol{D}$: Theorem 2.2 is applicable only when $\boldsymbol{D}$ is a scaled unitary matrix. Reference [64] extends the result to a certain type of positive definite matrices. Our results are more general, and hold true for any arbitrary matrix $\boldsymbol{D} \neq \boldsymbol{0}$.

- *Bound for scaled unitary matrices*: For the special case of $\boldsymbol{D} = a\boldsymbol{U}$, (2.13) reduces to the following:

$$Km\delta^2 \geq \widetilde{c} \left[9s \ln\left(\frac{eN}{s}\right) + 2\ln\left(\frac{2}{\epsilon}\right) \right], \quad (2.29)$$

for $0 < \delta < 1$. We see that there is an extra term on the right hand side of (2.28) of

Theorem 2.2. We can bound this term as follows:

$$\frac{\|1-a^2|\,K+\min\{1,a^2\}}{\max\{1,a^2\}} \geq \frac{\|1-a^2|+\min\{1,a^2\}}{\max\{1,a^2\}} = 1, \tag{2.30}$$

for all $a \neq 0$. Hence, our results are stronger than Theorem 2.2 for the scaled unitary matrix case.

- *Dependency on the eigenvalue*: The condition (2.28) heavily depends on the eigenvalue a of $\boldsymbol{D}$. The least number of measurements Km are required for $|a| = 1$, and as $|a|$ moves away from unity, the lower bound on Km increases. However, our results depend only on the ratio of the smallest to the largest singular value of $\boldsymbol{D}$, and therefore gives the best bound for all values of a. This is because our results make use of the fact that the recovery properties are independent of scaling due to the equivalence of (2.4) and (2.6). This critical observation allowed us to get stronger results compared to Theorem 2.2.

2.4.6 Extension to Identical Observation Matrices Case

Suppose we carry out a similar analysis for the case when all observation matrices are identical $\boldsymbol{A}_{(k)} = \boldsymbol{A}$ for $k = 0, 1, \ldots, K-1$, where $\boldsymbol{A}$ is a subgaussian random matrix with parameter c. The sufficient condition then obtained shows that the system is recoverable if (2.18) is satisfied. However, this condition ensures that the system is observable with $K = 1$. This is a weak result, because it implies that the availability of additional measurements does not improve the sufficient condition for observability. This is indeed true when $\boldsymbol{D} = \alpha \boldsymbol{I}$, for some $\alpha \in \mathbb{R}$, because we are only adding scaled versions of the rows of $\boldsymbol{A}$ to $\widetilde{\boldsymbol{A}}_{(K)}$ as K increases. For general $\boldsymbol{D}$, a different proof technique has to be used, which is

discussed in the next section.

2.5 RIP For Identical Observation Matrices Case

In this section, we present a result on the RIP of the observability matrix when the observation matrices are identical random matrices. First, we define the following quantities:

$$\widetilde{\boldsymbol{D}}_{(K,i)} \triangleq \left[\boldsymbol{I}_i \quad \boldsymbol{D}_i \quad \dots \quad \boldsymbol{D}_i^{K-1}\right], \tag{2.31}$$

$$L\left(\boldsymbol{D}, K\right) \triangleq \max_i \left\|\widetilde{\boldsymbol{D}}_{(K,i)}\right\|_2, \tag{2.32}$$

where $\boldsymbol{I}_i$ is the i^{th} column of identity matrix of size $N \times N$ and $\boldsymbol{D}_i^k$ is the i^{th} column of matrix $\boldsymbol{D}^k$.

Theorem 2.3 (Identical random observation matrices)**.** *Suppose all the observation matrices are identical, i.e.,* $\boldsymbol{A}_{(k)} = \boldsymbol{A}$ *for* $k = 0, 1, \dots, K-1$*, where* $\boldsymbol{A}$ *is a subgaussian random matrix with parameter* c*. Then, if*

$$Km\frac{\left(\delta - 1 + \lambda^{2(K-1)}\right)^2}{L^2\left(\boldsymbol{D}, K\right)} \geq \widetilde{c}s \max\left\{\ln^2 s \ln^2 N, \ln(2\epsilon^{-1})\right\}, \tag{2.33}$$

then the RIC δ_s *of the scaled version* $\widetilde{\boldsymbol{L}}(\boldsymbol{\lambda}_{\max})\widetilde{\boldsymbol{A}}_{(K)}$ *of* $\widetilde{\boldsymbol{A}}_{(K)}$ *satisfies* $\delta_s < \delta$ *for all* $1 - \lambda^{2(K-1)} < \delta < 1$ *with probability at least* $1 - \epsilon$*. Here,* $\widetilde{c}$ *is a constant dependent only on* c*, and* $\lambda \leq 1$ *is the ratio of the smallest to the largest singular values of* $\boldsymbol{D}$*. Hence, when* (2.33) *holds, the system is observable for sufficiently large* λ *with high probability.*

Proof. See Appendix A.3. □

In the following subsections, we provide more insights into the above results.

2.5.1 Special Cases

1. Suppose $\boldsymbol{D}$ is a scaled identity matrix. Then, $\lambda = 1$, and $L^2(\boldsymbol{D}, K) = K$, and hence from Theorem 2.3, we retrieve the recovery condition for a standard compressed sensing problem with m measurements, and the guarantee does not improve with increasing K. This is intuitive, because we are only adding scaled versions of the rows of $\boldsymbol{A}$ to $\widetilde{\boldsymbol{A}}_{(K)}$ as K increases.

2. Suppose $\boldsymbol{D}$ is rank-deficient. Then, $\lambda = 0$, and (2.33) does not hold for any $\delta < 1$, unless the following holds:

$$m \geq \widetilde{c} s \max\left\{\ln^2 s \ln^2 N, \ln(2\epsilon^{-1})\right\}, \tag{2.34}$$

   as expected.

3. Suppose that $\boldsymbol{D}$ is ill-conditioned, i.e., λ is close to zero. Then, the upper bound on δ required to guarantee observability may not hold [29, 77, 78], which is in similar vein as explained in the case of Theorem 2.1.

4. For $K = 1$, Theorem 2.3 reduces to the recovery condition of the standard compressed sensing problem [32]. Also, if the system is observable with m measurements (for example, when (2.34) is satisfied), the conditions in Theorem 2.3 hold for $K = 1$, as expected.

2.5.2 Number of Measurements

Theorem 2.3 shows that $Km = \mathcal{O}(s \ln^2 s \ln^2 N)$ is sufficient for observability, whereas $\mathcal{O}(N)$ measurements are necessary for observability of a non-sparse initial state vector.

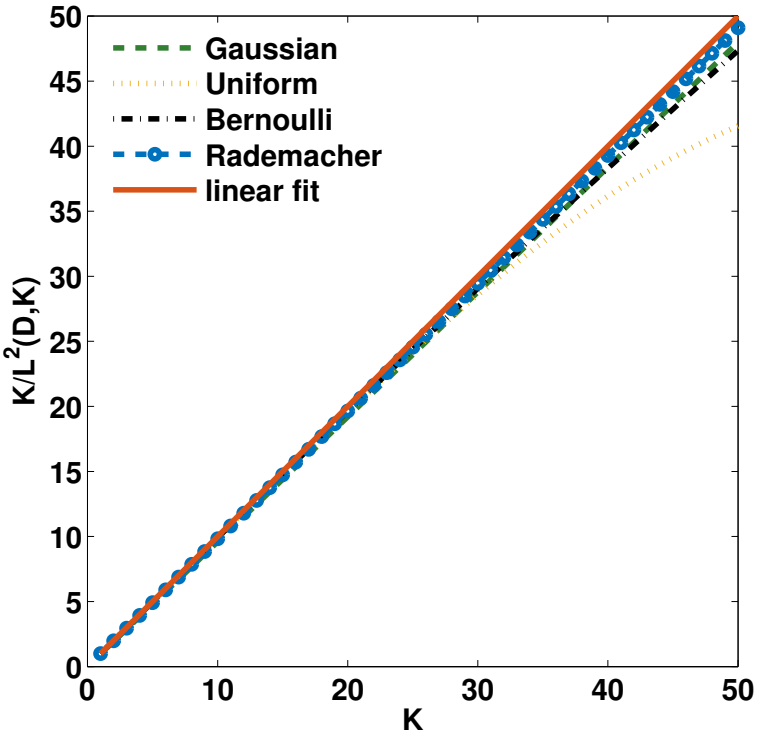


Figure 2.1: Variation of $K/L^2(\boldsymbol{D}, K)$ with K when $\boldsymbol{D}$ generated using different distributions. We see that $K/L^2(\boldsymbol{D}, K)$ is a (linearly) non-decreasing function of K.

Also, as mentioned in Section 1.1, the initial state can be recovered using any of the compressed sensing techniques like basis pursuit, thresholding algorithms, or greedy algorithms. As in the case of Theorem 2.1, the RIP based guarantees fix an upper bound on K, and hence a lower bound on m. However, note that, if the system is observable for K_1 measurements, it remains observable for $K > K_1$.

The main difference between the results in Theorem 2.1 and Theorem 2.3 is in the $L^2(\boldsymbol{D}, K)$ term. Hence, in order to gain intuition on the number of measurements required in the identical observation matrices case, we study the behavior of the $L(\boldsymbol{D}, K)$ term in the following proposition.

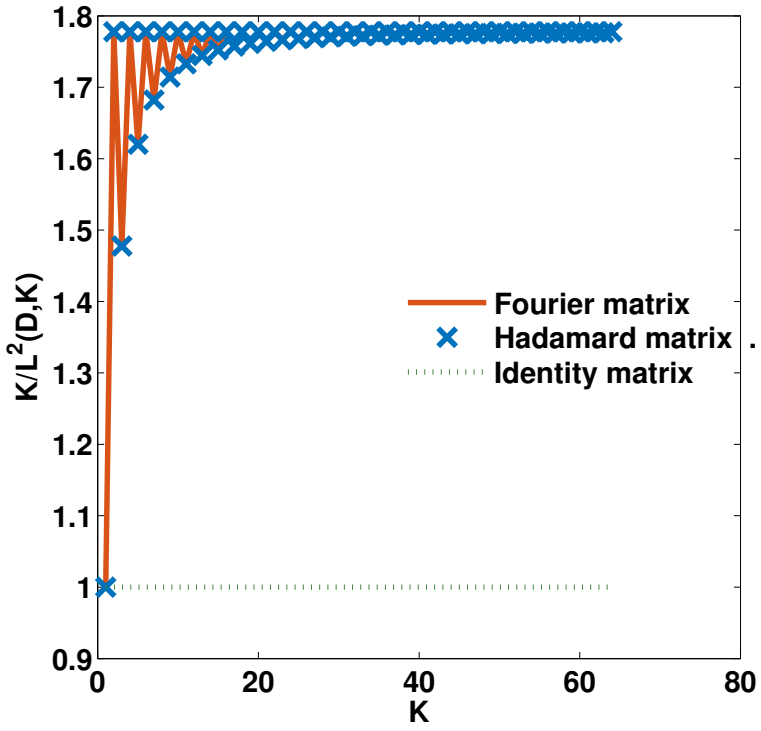


Figure 2.2: Variation of $K/L^2(\boldsymbol{D}, K)$ with K for Fourier, Hadamard and identity constructions of $\boldsymbol{D}$. We see that $K/L^2(\boldsymbol{D}, K)$ is not an increasing function of K.

Proposition 2.2. *The term* $K/L^2(\boldsymbol{D}, K)$ *satisfies*

$$1 \leq K/L^2(\boldsymbol{D}, K) \leq K. \tag{2.35}$$

Proof. See Appendix A.8. □

We note that the upper and the lower bounds are achieved by $\boldsymbol{D} = \boldsymbol{0}$, and $\boldsymbol{D} = \boldsymbol{I}$, respectively. Further, as discussed in Section 2.5.1, both cases are not favorable from the observability point of view. Although the $\boldsymbol{D} = \boldsymbol{0}$ case achieves the upper bound of the term $K/L^2(\boldsymbol{D}, K)$, this choice is not desirable since $\lambda = 0$.

In Figure 2.1, we empirically illustrate that if $\boldsymbol{D}$ is randomly chosen, the upper bound

can be nearly achieved. Also, when $\boldsymbol{D}$ is a random matrix, λ is non-zero with high probability [79], which makes this choice desirable. Random transfer matrices occur in some linear dynamical systems with sparse initial state, which models homogeneous spreading of epidemic or information or computer virus. For example, a dynamical system corresponding to a epidemic spread can be modeled using a Erdos-Renyi model in which case the transfer matrix has independent Bernoulli distributed entries [80–83]. In Figure 2.1, we use $N = 50$ and vary K from 1 to N [80]. The entries of $\boldsymbol{D}$ are drawn from the following four distributions:

1. Standard Gaussian distribution
2. Uniform distribution on $[0, 1]$
3. Bernoulli distribution with parameter 0.5
4. Rademacher distribution.

The curve $f(K) = K$ is labeled as *linear fit*. The value of $K/L^2(\boldsymbol{D}, K)$ is averaged over 100 trials and plotted along the Y-axis as a function of K. In all the four cases, the behavior of the curves is nearly linear, and $K/L^2(\boldsymbol{D}, K) \approx K$.

Before we discuss the implications of the result, we first give some intuition on reason behind this linear behavior. Since $L^2(\boldsymbol{D}, K)$ is a complicated function of $\boldsymbol{D}$, we focus on the regime where N is large and the distribution of the entries of $\boldsymbol{D}$ is Gaussian. We note that, from the Marchenko-Pastur theorem [84], the spectral norm of a Gaussian matrix with zero mean and unit variance is close to $2\sqrt{N}$. Thus, after normalization, as K increases, $\boldsymbol{D}^K$ goes to $\mathbf{0}$, and the extra column that gets added to the matrix $\tilde{\boldsymbol{D}}_{K,i}$ as K increases is approximately $\mathbf{0}$. Therefore, we have $\left\|\tilde{\boldsymbol{D}}_{(K,i)}\right\|_2 \approx \left\|\tilde{\boldsymbol{D}}_{(2,i)}\right\|_2$. Further,

$\left\|\tilde{\boldsymbol{D}}_{(2,i)}\right\|_2$ is the same as the largest eigenvalue of the matrix: $\begin{bmatrix} 1 & \boldsymbol{D}_{ii} \\ \boldsymbol{D}_{ii} & \|\boldsymbol{D}_i\|^2 \end{bmatrix}$. Also, for large N, $\|\boldsymbol{D}_i\|^2 \approx 1$ and $\boldsymbol{D}_{ii}^2 \approx 0$, which gives $\left\|\tilde{\boldsymbol{D}}_{(K,2)}\right\|_2 \approx 1$. Hence, we have the following relation:

$$L(\boldsymbol{D}, K) = \arg\max_i \left\|\tilde{\boldsymbol{D}}_{(K,i)}\right\|_2 \approx \left\|\tilde{\boldsymbol{D}}_{(K,2)}\right\|_2 \approx 1. \tag{2.36}$$

Thus, intuitively, $K/L^2(\boldsymbol{D}, K) \approx K$ for sufficiently large N. Also, from Figure 2.1, we see that $N = 50$ is large enough for the argument to hold.

Next, we discuss some implications of Figure 2.1. The RIP of the observability matrix $\widetilde{\boldsymbol{A}}_{(K)}$ is on par with an unstructured random Gaussian matrix. In turn, this suggests that it is not necessary to use independent observation matrices to ensure observability; identical observation matrices result in a penalty of only $O(\ln^2 s \ln N)$ in terms of the number of measurements required, provided $\boldsymbol{D}$ is a random matrix. Hence, we conclude that even though $\boldsymbol{I}$ and $\boldsymbol{0}$ are poor choices for $\boldsymbol{D}$, matrices with good recovery properties are plenty in number.

Another important observation from the plot is that $K/L^2(\boldsymbol{D}, K)$ is a non-decreasing function of K. Thus, as K increases, we need a smaller value of m for exact recovery of the sparse initial state. That is, one can trade-off K and m while ensuring observability. The following result shows that the function is increasing for the special case when $\boldsymbol{D}$ is a positive semi-definite (psd) matrix.

Proposition 2.3. *When $\boldsymbol{D}$ is a psd matrix, the function $K/L^2(\boldsymbol{D}, K)$ is a non-decreasing function of K.*

Proof. See Appendix A.9. □

Remark 1: The above result does not, in general, imply that $K/L^2(\boldsymbol{D}, K)$ increases with

K. For example, from Figure 2.2, we see that $K/L^2\left(\boldsymbol{D}, K\right)$ for standard dictionaries like Fourier and Hadamard matrices is not strictly increasing with K.

Remark 2: Theorem 2.3 also has corollaries similar to Corollary 2.1-Corollary 2.3 and Corollary 2.5. We omit those results to avoid repetition.

2.6 Joint Recovery of Sparse Initial State and Sparse Inputs

We now discuss the extension of the results presented thus far to the problem of jointly estimating the initial state as well as the input sequence, under sparsity constraints [68]. The system model in this case is as follows:

$$\boldsymbol{x}_k = \boldsymbol{D}\boldsymbol{x}_{k-1} + \boldsymbol{H}\boldsymbol{h}_k, \tag{2.37}$$

$$\boldsymbol{y}_k = \boldsymbol{A}_{(k)}\boldsymbol{x}_k, \tag{2.38}$$

where $\boldsymbol{H} \in \mathbb{R}^{n\times L}$ is the input matrix and $\boldsymbol{h}_k \in \mathbb{R}^L$ is the input vector such that $\|\boldsymbol{h}_k\|_0 \leq s_{in}$. Therefore, the sparse recovery problem is given by the following equation:

$$\widetilde{\boldsymbol{y}}_{(K)} = \tilde{\boldsymbol{A}}_{(K)}\boldsymbol{x}_0 + \tilde{\boldsymbol{J}}_{(K)}\tilde{\boldsymbol{h}}_{(K)}, \tag{2.39}$$

where the measurement vector $\tilde{\boldsymbol{y}}_{(K)} \in \mathbb{R}^{Km}$ (as defined in (2.5)), the unknown sparse vector $\widetilde{\boldsymbol{h}}_{(K)} \triangleq \begin{bmatrix}\boldsymbol{h}_1^{\mathsf{T}} & \dots & \boldsymbol{h}_{K-1}^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^{(K-1)L}$ which is at most $\tilde{s} = s + (K-1)s_{in}$ sparse,

and the matrix $\tilde{\boldsymbol{J}}_{(K)} \in \mathbb{R}^{Km\times(K-1)L}$ is defined follows:

$$\tilde{\boldsymbol{J}}_{(K)} = \begin{bmatrix} \mathbf{0} \in \mathbb{R}^{m\times(K-1)L} \\ \boldsymbol{A}_{(1)}\tilde{\boldsymbol{H}}_{(1)} \in \mathbb{R}^{m\times L} \quad \mathbf{0} \in \mathbb{R}^{m\times(K-2)L} \\ \boldsymbol{A}_{(2)}\tilde{\boldsymbol{H}}_{(2)} \in \mathbb{R}^{m\times 2L} \quad \mathbf{0} \in \mathbb{R}^{m\times(K-3)L} \\ \vdots \\ \boldsymbol{A}_{(K-1)}\tilde{\boldsymbol{H}}_{(K-1)} \in \mathbb{R}^{m\times(K-1)L} \end{bmatrix} \tag{2.40}$$

$$\tilde{\boldsymbol{H}}_{(k)} = \begin{bmatrix} \boldsymbol{D}^{k-1}\boldsymbol{H} & \boldsymbol{D}^{k-2}\boldsymbol{H} & \dots & \boldsymbol{H} \end{bmatrix} \in \mathbb{R}^{N\times kL}. \tag{2.41}$$

Comparing (2.39) with (2.4), the effective measurement matrix of the recovery problem takes the form

$$\begin{bmatrix} \boldsymbol{A}_{(0)}\boldsymbol{U}_{(0)} \\ \boldsymbol{A}_{(1)}\boldsymbol{U}_{(1)} \\ \vdots \\ \boldsymbol{A}_{(K-1)}\boldsymbol{U}_{(K-1),} \end{bmatrix},$$

where we define

$$\boldsymbol{U}_{(k)} \triangleq \begin{bmatrix} \boldsymbol{D}^k & \tilde{\boldsymbol{H}}_{(k)} & \mathbf{0}_{N\times(K-1-k)L} \end{bmatrix} \in \mathbb{R}^{N\times(N+(K-1)L)}. \tag{2.42}$$

To state results similar to Theorem 2.1 and Theorem 2.3, we define $\tilde{\delta}_{s,\max}$ as the largest of the RICs among the matrices $\left\{\boldsymbol{U}_{(k)}\right\}_{k=0}^{K-1}$.The proofs of the two theorems below are similar to that of the earlier results, and hence are omitted.

Theorem 2.4 (Independent random observation matrices)**.** *Suppose the measurement*

matrices $\boldsymbol{A}_{(k)}, k = 0, 1, \ldots, K-1$ *are independent subgaussian random matrices with parameter* c*. Then, if*

$$Km\left(\delta - 1 + (1-\tilde{\delta}_{s,\max})^2\right)^2 \geq \widetilde{c}\left[9s\ln\left(\frac{eN}{s}\right) + 2\ln(2\epsilon^{-1})\right], \tag{2.43}$$

the RIC δ_s *of a suitably scaled version* $\tilde{\boldsymbol{L}}(\boldsymbol{\delta})\begin{bmatrix}\tilde{\boldsymbol{A}}_{(K)} & \tilde{\boldsymbol{J}}_{(K)}\end{bmatrix}$ *of* $\begin{bmatrix}\tilde{\boldsymbol{A}}_{(K)} & \tilde{\boldsymbol{J}}_{(K)}\end{bmatrix}$ *satisfies* $\delta_s < \delta$ *for all* $1-(1-\tilde{\delta}_{s,\max})^2 < \delta < 1$ *with probability at least* $1-\epsilon$*. Here,* $\widetilde{c}$ *is a constant dependent only on* c*, and* $\boldsymbol{\delta}_j = 1 - \tilde{\delta}_{s,j}$ *where* $\tilde{\delta}_{s,j}$ *is the RIC of matrix* $\boldsymbol{U}_{(j)}$*. Hence, when* (2.43) *holds for* $s = \tilde{s}$*, the system is observable for sufficiently small* $\tilde{\delta}_{s,\max}$ *with high probability.*

Next, to state the corresponding result for the identical observation matrices case, we define quantities similar to (2.31) and (2.32) as follows:

$$\widetilde{\boldsymbol{U}}_{(K,i)} \triangleq \begin{bmatrix}\boldsymbol{U}_{(0)i} & \boldsymbol{U}_{(1)i} & \ldots & \boldsymbol{U}_{(K-1)i}\end{bmatrix}, \tag{2.44}$$

$$L_{\boldsymbol{U}}(\boldsymbol{D}, \boldsymbol{H}, K) \triangleq \max_i \left\|\widetilde{\boldsymbol{U}}_{(K,i)}\right\|_2, \tag{2.45}$$

where $\boldsymbol{U}_{(k)i}$ denotes the i^{th} column of $\boldsymbol{U}_{(k)}$.

Theorem 2.5 (Identical random observation matrices)**.** *Suppose all observation matrices are identical, i.e.,* $\boldsymbol{A}_{(k)} = \boldsymbol{A}$ *for* $k = 0, 1, \ldots, K-1$*, where* $\boldsymbol{A}$ *is a subgaussian random matrix with parameter* c*. Then, if*

$$Km\frac{\left(\delta - 1 + (1-\tilde{\delta}_{s,\max})^{2(K-1)}\right)^2}{L^2_{\boldsymbol{U}}(\boldsymbol{D}, \boldsymbol{H}, K)} \geq \widetilde{c}s\max\left\{\ln^2 s\ln^2 N, \ln(2\epsilon^{-1})\right\}, \tag{2.46}$$

then the RIC δ_s *of a suitably scaled version* $\tilde{\boldsymbol{L}}(\boldsymbol{\delta})\begin{bmatrix}\tilde{\boldsymbol{A}}_{(K)} & \tilde{\boldsymbol{J}}_{(K)}\end{bmatrix}$ *of* $\begin{bmatrix}\tilde{\boldsymbol{A}}_{(K)} & \tilde{\boldsymbol{J}}_{(K)}\end{bmatrix}$ *satisfies* $\delta_s < \delta$ *for all* $1-(1-\tilde{\delta}_{s,\max})^2 < \delta < 1$ *with probability at least* $1-\epsilon$*. Here,* $\widetilde{c}$ *is a constant*

dependent only on c, and $\boldsymbol{\delta}_j = 1 - \tilde{\delta}_{s,j}$ *where* $\tilde{\delta}_{s,j}$ *is the RIC of matrix* $\boldsymbol{U}_{(j)}$*. Hence, when* (2.46) *holds for* $s = \tilde{s}$*, the system is observable for sufficiently small* $\tilde{\delta}_{s,\max}$ *with high probability.*

Remark 1: As before, we can extend the above results to the nonconsecutive measurements, noncanonical basis and robust recovery cases. Also, conditions in Theorem 2.4 and Theorem 2.5 can be made less stringent using the RIC of $\boldsymbol{D}$. We omit explicitly stating the results to avoid repetition.

Remark 2: The above three theorems show how to extend three main results of the chapter (Theorem 2.1 and Theorem 2.3) to derive a sufficient condition for the structured random matrix in (2.40) to satisfy the RIP. These results could be of independent interest: they provide insight to the RIP of two special types of structured random matrices (resulting from independent and identical $\boldsymbol{A}_{(k)}$).

2.7 Summary

In this chapter, we derived the conditions for a linear dynamical system to be observable using the knowledge of its noiseless observations and inputs, when the initial state is sparse. We derived the results in the stochastic setting, both when the observation matrices are independent random matrices and when they are identical to a single random matrix. We characterized the number of measurements that are sufficient to observe the state of the linear dynamical system, using tools from compressed sensing. Thus, we completed the detailed theoretical analysis for the model with only initial state unknown. In the next chapter, we progress to the next level model in which both control inputs and the initial state is unknown (model SM2).

Chapter 3

Sparse-Controllability

Answering problem Q1 for SM2

In this chapter, we consider at the LDS sparsity model with both initial state and sparse inputs being unknown, and all system matrices known (model SM2). The unknown control inputs are assumed to sparse in a suitable basis. For this setting, we examine the conditions for sparse-controllability which is defined as the existence of a set of sparse control inputs that can drive the system from any arbitrary state to any other arbitrary final state. We note that unlike the previous chapter, we do not assume that the initial state is sparse. This chapter covers the necessary and sufficient conditions for the controllability, upper and lower bounds on the number of input vectors that ensure controllability, a state space decomposition to separate sparse-controllable and sparse-uncontrollable spaces, and extensions to the output controllability case.

3.1 Background

Networked control systems have attracted intense research attention from both academia and industry over the past decades [85–89]. In such a system, the notion of controllability refers to the ability to drive the system from an arbitrary initial state to a desired final state within a finite amount of time. Complete characterization of controllability of linear dynamical systems using unconstrained inputs have pure algebraic forms, and are rather easily verifiable [57, 90]. These conditions involve verification of the rank conditions of suitably defined matrices. However, in applications involving networked control systems, it is often necessary to select a small subset of the available sensors or actuators at each time instant, due to cost or energy constraints. For example, in an energy constrained network, energy-aware scheduling of actuators can help to extend the battery life of the nodes [91]. Similarly, in a system where the controller and plant communicate over a network, sparse control signals are required to meet the bandwidth constraints imposed by the links over which the control signals are exchanged [48, 49]. Now, when the number of actuators or input variables that can be activated is limited, the system may become uncontrollable because all the feasible control signals are restricted to lie in the union of low-dimensional subspaces. Thus, the controllability of linear dynamical systems under sparse input constraints is an important problem, and is the focus of this chapter.

3.1.1 Related Literature

We first discuss the relationship between the problem considered in this chapter and the existing literature in control theory and sparse signal porcessing.

Time-varying actuator scheduling problem

This problem focuses on finding a schedule for sparse actuator control, such that the system is sparse-controllable [88, 89, 91]. These works rely on a well known condition for controllability, namely, an extended version of the Kalman rank test. This test depends on the rank of the so-called Gramian matrix of the sparsity-constrained system.[1] However, finding sequence of control inputs that satisfy the rank condition on the Gramian matrix is a combinatorial problem, and it is known to be NP-hard [92, 93]. Moreover, it has been recently shown that the relatively simpler problem of finding a sparse set of actuators to guarantee reachability of a particular state is hard to approximate, even when a solution is known to exist [94]. Hence, different quantitative measures of controllability based on the Gramian matrix have been considered: smallest eigenvalue, the trace of the inverse, inverse of the trace, the determinant, maximum entry in the diagonal, etc. [91]. Based on these metrics, several algorithms and related guarantees are available in the literature [85, 88, 89]. However, none of the above mentioned references directly address the fundamental question of whether or not the system can be controlled by sparse inputs. Further, direct extension of the Kalman rank test leads to a combinatorial problem that is computationally infeasible to solve in practice. Thus, the goal of our chapter is to study the controllability of a linear dynamical system under sparsity constraints without directly relying on Gramian matrix. We are not interested in finding the optimal actuator selection; rather we deal with the more basic problem of deriving conditions for the existence of a selection that drives the system from any initial state to any final state.

[1] Refer to [91, Section II.B] for details.

Minimal input selection problem

The minimal input selection involves selecting a small set of input variables so that the system is controllable using the selected set [92,93,95]. This problem is a special case of our sparse input problem because of the extra constraint that the support of the control input remains unchanged for all time instants. Moreover, the controllability conditions for the minimal input selection problem can be easily be derived from the classical controllability results for the unconstrained system. We discuss and contrast the two cases in detail in Section 3.3.3.

Design of sparse control inputs

Some recent works connecting compressive sensing and control theory focus on the design of control inputs [68, 96, 97]. They propose algorithms for the recovery (design) of sparse control inputs based on the observations, and derive conditions under which the input can be uniquely recovered using a limited number of observations [68, 96, 97]. These problems do not deal with controllability related issues, rather assume the existence of sparse control inputs and initial state for reaching a given final state.

Observability under sparsity constraints

Due to the recent advances in sparse signal processing and compressed sensing, researchers have recently started looking at the observability of linear systems with a sparse initial state [50, 51, 62, 67]. For a system with unconstrained inputs, observability and controllability are dual problems and do not require separate analysis. However, our problem assumes a general initial state and sparse control inputs, whereas [50, 51, 62, 67] consider a sparse initial state and known control inputs. Therefore, the problems have different

sparsity pattern models, and consequently require separate analysis.

Sparse signal recovery guarantees

The sparse controllability problem studies the conditions that ensure the existence of sparse control inputs to drive a linear system from any given state to any other state. Moreover, it is not required that the solution be unique. In contrast, the focus of traditional sparse signal processing studies is on developing algorithms and guarantees for the cases where the linear system is already known to admit a sparse solution [9–11,32]. Also, the structure of the effective measurement matrix that arises in the context of linear dynamical systems is different from the type of random measurement matrices that are usually considered in the compressed sensing literature.

In the light of the discussion thus far, the primary questions that we address in this chapter are as follows:

1. What are necessary and sufficient conditions for ensuring controllability under sparse input constraints? Can we devise a simple method to test for controllability?

2. If a system is controllable using sparse inputs, what is the minimum number of control input vectors needed to drive the system from a given initial state to an arbitrary final state?

3. If the system is not controllable using sparse inputs, what parts of the state space are reachable using sparse inputs? In other words, how does one decompose the state space into three subspaces: uncontrollable, uncontrollable using sparse inputs and controllable using sparse inputs?

Answering above questions requires a fresh look at controllability, and we start by deriving a Popov-Belevitch-Hautus (PBH)-like test [90]. Unlike the Gamian matrix based test discussed above, the new approach presented in this chapter allows one to check for sparse-controllability of a system without solving a combinatorial problem. In a nutshell, this chapter studies theoretical aspects of the one of the most important notion in control theory: controllability under sparsity constraints on the input. We also note that the classical results for the unconstrained system can be recovered as a special case of our results, by relaxing the sparsity constraint.

3.2 System Model

We consider the discrete-time linear dynamical system

$$\boldsymbol{x}_k = \boldsymbol{D}\boldsymbol{x}_{k-1} + \boldsymbol{H}\boldsymbol{h}_k, \tag{3.1}$$

where the transfer matrix $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ and input matrix $\boldsymbol{H} \in \mathbb{R}^{N\times L}$. Here, the input vectors are assumed to be sparse, i.e., $\|\boldsymbol{h}_k\|_0 \leq s$ for all k. We denote the rank of the matrices $\boldsymbol{D}$ and $\boldsymbol{H}$ using $R_{\boldsymbol{D}}$ and $R_{\boldsymbol{H}}$, respectively.

We revisit the problem of controllability in the context of sparsity. We formally define the notion of controllability using sparse inputs as follows:

Definition 3.1 (Sparse-controllability)**.** *The linear system defined by* (3.1) *is said to be* $s-$*sparse-controllable if for any initial state* $\boldsymbol{x}_0$ *and any final state* $\boldsymbol{x}_K$*, there exists an input sequence* $\boldsymbol{h}_k$*,* $k = 1, 2, \cdots, K$ *such that* $\|\boldsymbol{h}_k\|_0 \leq s$*, which steers the system from the state* $\boldsymbol{x}_0$ *to* $\boldsymbol{x}_K$ *for some finite* K*.*

Next, to characterize the sparse-controllability of the system, we consider the following

equivalent system of equations:

$$\boldsymbol{x}_K - \boldsymbol{D}^K \boldsymbol{x}_0 = \tilde{\boldsymbol{H}}_{(K)} \boldsymbol{h}_{(K)}, \tag{3.2}$$

where we define the matrices as follows:

$$\tilde{\boldsymbol{H}}_{(K)} = \begin{bmatrix} \boldsymbol{D}^{K-1}\boldsymbol{H} & \boldsymbol{D}^{K-2}\boldsymbol{H} & \dots \boldsymbol{H} \end{bmatrix} \in \mathbb{R}^{N \times KL} \tag{3.3}$$

$$\boldsymbol{h}_{(K)} = \begin{bmatrix} \boldsymbol{h}_1^\mathsf{T} & \boldsymbol{h}_2^\mathsf{T} & \dots & \boldsymbol{h}_K^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{KL}. \tag{3.4}$$

Note that $\boldsymbol{h}_{(K)}$ is a *piecewise sparse vector*, i.e., it is formed by concatenating K sparse vectors, each with sparsity at most s.

3.3 Necessary and Sufficient Conditions for Sparse-Controllability

This section addresses question 1 in Section 3.1. It is well-known that the system is sparse-controllable if for some finite K, there exists a submatrix of $\tilde{\boldsymbol{H}}_{(K)}$ with rank N of the following form:

$$\begin{bmatrix} \boldsymbol{D}^{K-1}\boldsymbol{H}_{\mathcal{S}_1} & \boldsymbol{D}^{K-2}\boldsymbol{H}_{\mathcal{S}_2} & \dots & \boldsymbol{H}_{\mathcal{S}_K} \end{bmatrix} \in \mathbb{R}^{N \times Ks},$$

such that the index set $\mathcal{S}_i \subseteq \{1, 2, \dots, L\}$ and $|\mathcal{S}_i| = s$, for $i = 1, 2, \dots, K$. In the sequel, we refer this condition to as the *Kalman-type rank test*. Note that the first $(K-1)N$ columns of $\tilde{\boldsymbol{H}}_{(K)}$ belong to $\mathcal{CS}\{\boldsymbol{D}\}$. Hence, to satisfy the Kalman-type rank test, $\mathcal{S}_K$ should be such that $\mathcal{CS}\{\boldsymbol{H}_{\mathcal{S}_K}\}$ should contain the left null space of $\boldsymbol{D}$. Thus, the Kalman-type rank test naturally leads to the necessary condition for sparse-controllability as the existence

of an index set $\mathcal{S}$ with s entries such that rank of the matrix $\begin{bmatrix}\boldsymbol{D} & \boldsymbol{H}_{\mathcal{S}}\end{bmatrix} \in \mathbb{R}^{N\times(N+s)}$ is N. With this intuition in mind, we next show that the above condition is not only necessary but also sufficient for a controllable system to be $s-$sparse-controllable.

Theorem 3.1. *The system given by* (3.1) *is $s-$sparse-controllable if and only if the following two conditions hold:*

1. *For all $\lambda \in \mathbb{C}$, rank of the matrix $\begin{bmatrix}\lambda\boldsymbol{I}-\boldsymbol{D} & \boldsymbol{H}\end{bmatrix} \in \mathbb{R}^{N\times(N+L)}$ is N.*
2. *There exists an index set $\mathcal{S} \subseteq \{1,2,\ldots,L\}$ with s entries such that rank of matrix $\begin{bmatrix}\boldsymbol{D} & \boldsymbol{H}_{\mathcal{S}}\end{bmatrix} \in \mathbb{R}^{N\times(N+s)}$ is N.*

Proof. See Appendix B.1. □

We make the following remarks from Theorem 3.1:

- From condition 2, if a system is $s-$sparse-controllable, then for all $s \leq \tilde{s} \leq L$, it is $\tilde{s}-$sparse-controllable. This is intuitive since every $s-$sparse vector is also $\tilde{s}-$sparse.
- From condition 2, the system is $s-$sparse-controllable only if

$$\min\{R_{\boldsymbol{H}}, s\} \geq N - R_{\boldsymbol{D}}. \tag{3.5}$$

  This relation gives a necessary condition on the minimum sparsity s required to ensure the controllability using sparse inputs. Also, we note that for an unconstrained system, $\min\{R_{\boldsymbol{H}}, s\} = R_{\boldsymbol{H}}$, and thus $R_{\boldsymbol{H}} + R_{\boldsymbol{D}} \geq N$ is a necessary condition for controllability.

- For $s = L$, Theorem 3.1 reduces to the PBH test [90] since there is no constraint

on the input. Similarly, when $L = 1$, the notion of controllability and sparse-controllability are the same, and hence Theorem 3.1 reduces to the PBH test.

- If the system defined by the transfer matrix-input matrix pair $(\boldsymbol{D}, \boldsymbol{H}_{\mathcal{S}})$ is controllable for some index set $\mathcal{S}$ with s entries, the system is $s-$sparse-controllable. In particular, a controllable system with $R_{\boldsymbol{H}} \leq s$ is $s-$sparse-controllable.

Before we present the detailed implications of the theorem, we present some interesting corollaries of Theorem 3.1. The theorem assumes that the input vectors are sparse in the canonical basis. However, the result can be extended to the more general class of inputs that are sparse under a basis $\boldsymbol{\Psi} \in \mathbb{R}^{L\times L}$ other than the canonical basis to get the following corollary:

Corollary 3.1. *The system given by* (3.1) *is controllable using inputs which are $s-$sparse under a basis $\boldsymbol{\Psi} \in \mathbb{R}^{L\times L}$ if and only if the following two conditions hold:*

1. *For all $\lambda \in \mathbb{C}$, rank of $\begin{bmatrix}\lambda \boldsymbol{I} - \boldsymbol{D} & \boldsymbol{H}\end{bmatrix}$ is N.*
2. *There exists an index set $\mathcal{S} \subseteq \{1, 2, \ldots, L\}$ with s entries such that the rank of $\begin{bmatrix}\boldsymbol{D} & \boldsymbol{H}\boldsymbol{\Psi}_{\mathcal{S}}\end{bmatrix}$ is N.*

Proof. Since the input vector is sparse under the basis $\boldsymbol{\Psi}$, the effective input matrix becomes $\boldsymbol{H}\boldsymbol{\Psi}$. Thus, replacing $\boldsymbol{H}$ with $\boldsymbol{H}\boldsymbol{\Psi}$ in Theorem 3.1, we obtain a similar result for a non-canonical basis. Further, To obtain condition 1, we note that

$$\begin{bmatrix}\lambda \boldsymbol{I} - \boldsymbol{D} & \boldsymbol{H}\boldsymbol{\Psi}\end{bmatrix} = \begin{bmatrix}\lambda \boldsymbol{I} - \boldsymbol{D} & \boldsymbol{H}\end{bmatrix} \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Psi}\end{bmatrix}, \tag{3.6}$$

and the matrix $\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Psi}\end{bmatrix} \in \mathbb{R}^{N+L\times N+L}$ is invertible as $\boldsymbol{\Psi}$ is invertible. Therefore, rank of

$\begin{bmatrix}\lambda \boldsymbol{I} - \boldsymbol{D} & \boldsymbol{H}\boldsymbol{\Psi}\end{bmatrix}$ is N for all $\lambda \in \mathbb{C}$ if and only if rank of $\begin{bmatrix}\lambda \boldsymbol{I} - \boldsymbol{D} & \boldsymbol{H}\end{bmatrix}$ is N for all $\lambda \in \mathbb{C}$, which gives condition 1. □

Our next corollary gives a sufficient condition under which controllability without any input constraints is equivalent to sparse-controllability, for the system given by (3.1).

Corollary 3.2. *A reversible system, i.e., a system with an invertible state transition matrix $\boldsymbol{D}$, is $s-$sparse-controllable for any $0 < s \leq L$ if and only if it is controllable.*

Proof. See Appendix B.2. □

In the following subsection, we bring out the usefulness of Theorem 3.1 by developing a simple procedure to check the controllability of a system under sparsity constraints.

3.3.1 Verification Procedure

We present the following procedure to verify the conditions of Theorem 3.1:

1. Compute the eigendecomposition of $\boldsymbol{D}^{\mathsf{T}}$.
2. If $s < N - R_{\boldsymbol{D}}$, the system is not sparse-controllable.
3. Check condition 1 for λ set to be equal to each of the eigenvalues of $\boldsymbol{D}^{\mathsf{T}}$. If the condition fails, the system is not sparse-controllable.
4. If none of the eigenvalues of $\boldsymbol{D}^{\mathsf{T}}$ is zero, the system is sparse-controllable.
5. Project the columns of $\boldsymbol{H}$ onto the null space of $\boldsymbol{D}^{\mathsf{T}}$ obtained from its eigendecomposition to get $\boldsymbol{H}^{\perp} \in \mathbb{R}^{N \times L}$. If rank of the $\boldsymbol{H}^{\perp}$ is $N - R_{\boldsymbol{D}}$, the system is sparse-controllable.

Here, step 2 follows from necessary condition for sparse-controllability given by (3.5). Next, step 3 verifies condition 1 for all values of λ. This is because if there exists $\boldsymbol{z} \in \mathbb{R}^N, \lambda \in \mathbb{C}$ such that $\boldsymbol{z}^\mathsf{T} \begin{bmatrix} \lambda \boldsymbol{I} - \boldsymbol{D} & \boldsymbol{H} \end{bmatrix} = \boldsymbol{0}$, then $\boldsymbol{z}^\mathsf{T}\boldsymbol{D} = \lambda \boldsymbol{z}^\mathsf{T}$. Thus, it suffices to verify condition 1 for at most N values of λ. The step 4 follows from Corollary 3.2 if $\boldsymbol{D}$ has full column rank. Finally, in step 5, since the columns of $\boldsymbol{H}^\perp$ are orthogonal to the columns of $\boldsymbol{D}$, we have

$$\mathsf{Rank}\left\{\boldsymbol{H}^\perp\right\} \leq N - R_{\boldsymbol{D}} \leq s. \tag{3.7}$$

Here, the last inequality follows from step 2. Further, we also have

$$\mathsf{Rank}\left\{\begin{bmatrix} \boldsymbol{D} & \boldsymbol{H}_{\mathcal{S}} \end{bmatrix}\right\} = \mathsf{Rank}\left\{\begin{bmatrix} \boldsymbol{D} & \boldsymbol{H}_{\mathcal{S}}^\perp \end{bmatrix}\right\} \tag{3.8}$$

$$= \mathsf{Rank}\left\{\boldsymbol{D}\right\} + \mathsf{Rank}\left\{\boldsymbol{H}_{\mathcal{S}}^\perp\right\}. \tag{3.9}$$

Therefore, an index $\mathcal{S}$ satisfies condition 2 if and only if $\mathsf{Rank}\left\{\boldsymbol{H}_{\mathcal{S}}^\perp\right\} = N - R_{\boldsymbol{D}}$. Moreover, (3.7) ensures that this condition is equivalent to $\mathsf{Rank}\left\{\boldsymbol{H}^\perp\right\} = N - R_{\boldsymbol{D}}$, as verified by step 5.

The complexity of the procedure depends on the computations required for the eigendecomposition in step 1, $N + 1$ rank computations in steps 2 and 5, and the matrix multiplication in step 5 required for the projection operation using the obtained eigendecomposition. It is known that the complexity of finding the eigendecomposition of a matrix is $\mathcal{O}(N^3)$ [98]. Therefore, Theorem 3.1 allows us to verify the controllability of any discrete system in polynomial time in N. It is interesting to note that the complexity is independent of the sparsity s. We reiterate that the above procedure ensures that there exists a set of $s-$sparse control inputs for every pair of initial and final states of the system. However, it does not reveal any insight on the support pattern of the input sequence. The

determination of the support sequence is a completely different problem and is known to be NP-hard.

In the following two subsections, we discuss the several implications of Theorem 3.1 by relating it to the Kalman-type rank test and the minimal input selection problem.

3.3.2 Comparison with the Kalman-type Rank Test

The two conditions of Theorem 3.1 and the Kalman-type rank test described at the beginning of Section 3.3 are based on the two different characteristics of controllability, and provide insights into two aspects of the problem. The Kalman-type rank test identifies the range spaces of the possible controllability matrices for different sparsity patterns of the input. The union of these range spaces represents the set of all states that can be reached from zero-initial condition. This observation is immediate from (3.2) with $\boldsymbol{x}_0 = \mathbf{0}$. Therefore, the minimum number of input vectors required to satisfy the Kalman-type rank test characterizes the length of the input sequence, K, required to ensure controllability. We exploit this fact to characterize the minimum number of input vectors that ensures controllability in Section 3.4. Moreover, the Kalman-type rank test also identifies the support pattern of the input sequence that can drive the system from any given state to any other final state.

On the other hand, the conditions of Theorem 3.1 are based on recognizing the uncontrollable and sparse-uncontrollable parts of the system. Interestingly, these conditions are independent of the number of input vectors, K. The first condition is the same as the PBH test. Thus, it elegantly separates systems into three categories: *sparse-controllable, controllable but sparse-uncontrollable, and uncontrollable.* The system is *sparse-controllable* if both conditions 1 and 2 of the theorem are satisfied; *controllable but sparse-uncontrollable*

if only condition 1 is satisfied; and *uncontrollable* if condition 1 is violated. We exploit this observation to design a recipe to decompose the system into sparse-controllable and sparse-uncontrollable parts in Section 3.5.

Next, we comment on the computational effort required to verify the two tests. To verify the Kalman-type rank test, we need to do at most $\binom{L}{s}^N$ rank computations. However, as outlined in Section 3.3.1, Theorem 3.1 requires one to do at most $N+1$ rank computations, one eigendecomposition and a matrix multiplication. Therefore, the computational cost required for Theorem 3.1 is polynomial in N and independent of s. In contrast, the computational complexity of the Kalman-type rank test grows exponentially with N and s. We also note that, since the Kalman-type rank test involves powers of $\boldsymbol{D}$, numerical stability also needs to be considered. Overall, conditions of Theorem 3.1 are computationally easier to verify compared to the Kalman-type rank test.

3.3.3 Inputs with Common Support

We recall the minimal input selection problem discussed in Section 3.1. For such a problem, the system is controlled using sparse inputs with a common support, i.e., when the indices of the nonzero entries of all the inputs coincide. In this case, the effective system has the transfer matrix-input matrix pair as $(\boldsymbol{D}, \boldsymbol{H}_{\mathcal{S}})$ for some index set $\mathcal{S}$ such that $|\mathcal{S}| = s$. Hence, the controllability conditions are given as follows:

1. For some finite K, there exists a submatrix of $\tilde{\boldsymbol{H}}_{(K)}$ with rank N of the following form:

$$\begin{bmatrix} \boldsymbol{D}^{K-1}\boldsymbol{H}_{\mathcal{S}} & \boldsymbol{D}^{K-2}\boldsymbol{H}_{\mathcal{S}} & \dots & \boldsymbol{H}_{\mathcal{S}} \end{bmatrix} \in \mathbb{R}^{N \times Ks},$$

   such that the index set $\mathcal{S} \subseteq \{1, 2, \dots, L\}$ and $|\mathcal{S}| = s$.

2. For all $\lambda \in \mathbb{C}$, rank of the matrix $\begin{bmatrix} \lambda \boldsymbol{I} - \boldsymbol{D} & \boldsymbol{H}_{\mathcal{S}} \end{bmatrix} \in \mathbb{R}^{N \times (N+s)}$ is N, for some index set $\mathcal{S} \subseteq \{1, 2, \ldots, L\}$ such that $|\mathcal{S}| = s$.

We see that, due to the additional constraint of controllability using a common support, the above conditions are more stringent than those in Theorem 3.1. Thus, a system with sparse inputs offers greater flexibility and control, and incurs a similar communication cost,[2] compared to a system restricted to using sparse inputs with common support.

Finally, we provide some illustrative numerical examples in the following subsection.

3.3.4 Illustrative Examples

We first give an example to demonstrate that a controllable system which does not satisfy condition 2 of Theorem 3.1 is not sparse-controllable.

Example 3.1. *Consider a linear system with $N = 3$, $L = 2$,*

$$\boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \text{ and } \boldsymbol{H} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.10}$$

Using the PBH test, it is easy to see that the system is controllable. Also, using the procedure given Section 3.3.1, the system does not satisfy the conditions of Theorem 3.1.

We verify that the system is not $1-$sparse-controllable using the initial state $\boldsymbol{x}_0 = \boldsymbol{0}$ and final state $\boldsymbol{x}_f = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^{\mathsf{T}}$. From (3.2), we have,

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \sum_{k=1}^{K} \boldsymbol{D}^{K-k} \boldsymbol{H} \boldsymbol{h}_k = \begin{bmatrix} \sum_{k=1}^{K} \boldsymbol{h}_k[1] + \boldsymbol{h}_k[2] \\ \boldsymbol{h}_K[1] \\ \boldsymbol{h}_K[2] \end{bmatrix}. \tag{3.11}$$

[2]The communication cost remains of order s, since the support can be conveyed using $s \log(N)$ bits.

Since $\boldsymbol{h}_K$ *is* $1-$*sparse, the above system of equations does not have any solution, for any finite value of* K*. Thus, the system is not* $1-$*sparse-controllable.*

Our next example illustrates the benefits of using sparse control in a linear system over the sparse control with common support discussed in Section 3.3.3.

Example 3.2. *Consider a linear system with* $N = 3$*,* $L = 3$*,*

$$\boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \text{ and } \boldsymbol{H} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}. \tag{3.12}$$

This system satisfies the conditions in Theorem 3.1 for $s = 2$*, and is hence* $2-$*sparse-controllable. There are three possible unconstrained systems with input matrices of size* 3×2*:*

$$\boldsymbol{H}_{(1)} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \quad \boldsymbol{H}_{(2)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \boldsymbol{H}_{(3)} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

However, the three subsystems described by the matrix pair $(\boldsymbol{D}, \boldsymbol{H}_{(k)})$ *for* $k = 1, 2, 3$ *are individually uncontrollable. Hence, sparse control allows the system to be controllable without adding much communication burden.*

Finally, we give an example of a system with non-invertible $\boldsymbol{D}$ which is both controllable and sparse-controllable. This example shows that the condition in Corollary 3.2 that $\boldsymbol{D}$ is invertible is not necessary, but sufficient for a controllable system to be sparse-controllable.

Example 3.3. *Consider a linear system with* $N = 3$*,* $L = 2$*,*

$$\boldsymbol{D} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \text{ and } \boldsymbol{H} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}. \tag{3.13}$$

We note that $\boldsymbol{D}$ *is not an invertible matrix. Further, the system satisfies the conditions in Theorem 3.1 for* $s = 1$*, and hence it is* $1-$*sparse-controllable.*

In summary, in this section, we derived necessary and sufficient conditions that a system needs to satisfy to ensure sparse-controllability. We also presented a procedure with polynomial complexity to verify the conditions. Next, we address the question 2 in Section 3.1, and derive bounds on the number of sparse input vectors required to drive the system from a given state to any desired final state.

3.4 Minimum Number of Control Input Vectors

In this section, we upper and lower bound the minimum number of input vectors that are required to drive the system from any given state to any final state. Before we present the main result of the section, for comparison, we state the corresponding result for the unconstrained system. To state the result, we denote the degree of minimal polynomial of $\boldsymbol{D}$ using q.

Theorem 3.2. *For a controllable system, the minimum number of input vectors* K *required to steer the system from any given state to any other state satisfies*

$$N/R_{\boldsymbol{H}} \leq K \leq \min\{q, N - R_{\boldsymbol{H}} + 1\} \leq N. \tag{3.14}$$

Proof. See [99, Section 6.2.1]. □

We note that when we restrict the admissible inputs to sparse vectors, the minimum number of input vectors required can possibly increase. The following theorem gives bounds on the number of sparse control inputs.

Theorem 3.3. *For an $s-$sparse-controllable system, the minimum number of $s-$sparse input vectors K^* required to steer the system from any given state to any other state satisfies*

$$\frac{N}{\min\{R_{\boldsymbol{H}}, s\}} \leq K^* \leq \min\left\{q\left\lceil\frac{R_{\boldsymbol{H}}}{s}\right\rceil, N - R^*_{\boldsymbol{H},s} + 1\right\} \leq N, \tag{3.15}$$

where $R^*_{\boldsymbol{H},s} = \max\limits_{\substack{\mathcal{S}\subseteq\{1,2,\ldots,L\}\\|\mathcal{S}|=s}} \mathsf{Rank}\{\boldsymbol{H}_{\mathcal{S}}\}$.

Proof. See Appendix B.3. □

We can relax the above upper bound to get a simpler relation without $R^*_{\boldsymbol{H},s}$ as follows.

Corollary 3.3. *For an $s-$sparse-controllable system, the minimum number of input vectors K^* required to steer the system from any given state to any other state satisfies*

$$\frac{N}{\min\{R_{\boldsymbol{H}}, s\}} \leq K^* \leq \min\left\{q\left\lceil\frac{R_{\boldsymbol{H}}}{s}\right\rceil, R_{\boldsymbol{D}} + 1, N\right\}. \tag{3.16}$$

Proof. The result follows from condition 2 of Theorem 3.1 which gives the following:

$$R^*_{\boldsymbol{H},s} \geq \max\{N - R_{\boldsymbol{D}}, 1\}. \tag{3.17}$$

□

We make the following observations from Theorem 3.3.

- *Invariance:* The bound is invariant under right or left multiplication of $\boldsymbol{H}$ by a non-singular matrix, and under any similarity transform on $\boldsymbol{D}$.
- *Relation to s:* Both the upper and the lower bounds decrease with s. This is intuitive: as s increases, the system has more flexibility to choose its inputs, and thus it requires fewer number of input vectors to ensure controllability.

- *Equivalence between Theorem 3.2 and Theorem 3.3:* We consider three cases for comparison:

  1. When $s = L$, which corresponds to the unconstrained case, Theorem 3.3 reduces to Theorem 3.2, as expected.

  2. When $s \geq R_{\boldsymbol{H}}$, Theorem 3.3 reduces to Theorem 3.2, as $R^*_{\boldsymbol{H},s} = R_{\boldsymbol{H}}$. This follows because when $s \geq R_{\boldsymbol{H}}$, $\mathcal{CS}\left\{\tilde{\boldsymbol{H}}_{(K)}\right\}$ is the same as the column space of an $N \times Ks$ submatrix of $\tilde{\boldsymbol{H}}_{(K)}$ with maximum rank.

  3. When $\min\{q, N - \boldsymbol{R}_{\boldsymbol{H}} + 1\} = N$, the system requires the same number of control inputs to achieve controllability and $s-$sparse-controllability for any s. However, this is possible only if $\boldsymbol{R}_{\boldsymbol{H}} = 1$, and any system with $s \geq \boldsymbol{R}_{\boldsymbol{H}}$ is equivalent to an unconstrained system, as discussed above.

- *Equality:* We note that the upper and lower bounds in Theorem 3.2 meet when $N/R_{\boldsymbol{H}} = N - R_{\boldsymbol{H}} + 1$, which gives $R_{\boldsymbol{H}}$ as 1 or N. Similarly, for $s = 1$, the lower and upper bounds in Theorem 3.3 are equal, and $K^* = N$. Further, if $R_{\boldsymbol{H}} \geq s$, we get $R^*_{\boldsymbol{H},s} = s$, and thus the bounds are equal when $s = N$.

3.5 Decomposing Sparse-controllable States

In this section, we consider question 3 in Section 3.1, and present a decomposition of the state space into sparse-controllable, sparse-uncontrollable and uncontrollable subspaces. We begin with the observation that $s-$sparse-controllability inherits the *invariance under a change of basis* property of the conventional controllability.

Proposition 3.1 (Invariance under change of basis)**.** *The system defined by the matrix*

pair $(\boldsymbol{D}, \boldsymbol{H})$ *is* $s-$*sparse-controllable if and only if the system defined by the matrix pair* $(\boldsymbol{U}^{-1}\boldsymbol{D}\boldsymbol{U}, \boldsymbol{U}^{-1}\boldsymbol{H})$ *is* $s-$*sparse-controllable for every nonsingular* $\boldsymbol{U} \in \mathbb{R}^{N\times N}$.

Proof. We note that when $\boldsymbol{D}$ and $\boldsymbol{H}$ are replaced with $\boldsymbol{U}^{-1}\boldsymbol{D}\boldsymbol{U}$ and $\boldsymbol{U}^{-1}\boldsymbol{H}$ respectively, in (3.3), we get $\boldsymbol{U}^{-1}\tilde{\boldsymbol{H}}_{(K)}$ instead of $\tilde{\boldsymbol{H}}_{(K)}$. Now, the result follows from the Kalman-type rank test and the fact that the rank of every submatrix of $\tilde{\boldsymbol{H}}_{(K)}$ and $\boldsymbol{U}^{-1}\tilde{\boldsymbol{H}}_{(K)}$ are the same. □

Inspired by the above proposition and in the same spirit as the Kalman decomposition [100], we transform the original system to an equivalent *standard form* using a change of basis, such that, the transformed state-space is separated into an $s-$sparse-controllable subspace and an orthogonal $s-$sparse-uncontrollable subspace. As discussed in Section 3.3.2, the key idea of such a decomposition is as follows: if a system is not controllable, then it is not sparse-controllable. Therefore, the first step to decompose the system is to separate the controllable and uncontrollable states using the Kalman decomposition. Next, we identify the sparse-controllable part of the controllable part, for which we use Theorem 3.1. Recall that a controllable system is sparse-controllable if it satisfies condition 2 of Theorem 3.1. Hence, we need to find a basis for the controllable part such that the transformed state-space separates into two subsystems: one which satisfies condition 2 of Theorem 3.1, and the other which does not. The two bases together give the transform that decomposes the system to sparse-controllable and sparse-uncontrollable parts. We next describe the procedure for the decomposition followed by an explanation on why the procedure works.

1. Find a basis for $\mathcal{CS}\left\{\tilde{\boldsymbol{H}}_{(N)}\right\}$ as $\{\boldsymbol{u}_i\}_{i=1}^{R_1}$, where $R_1 \leq N$ is the rank of $\tilde{\boldsymbol{H}}_{(N)}$. Extend the basis by adding $N - R_1$ linearly independent vectors $\{\boldsymbol{u}_i\}_{i=R_1+1}^{N}$ to form a basis

for $\mathbb{R}^N$. Define an invertible matrix $\boldsymbol{U} \triangleq \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 \dots \boldsymbol{u}_N \end{bmatrix} \in \mathbb{R}^{N \times N}$.

2. Compute $\check{\boldsymbol{D}} = \boldsymbol{U}^{-1}\boldsymbol{D}\boldsymbol{U}$ and $\check{\boldsymbol{H}} = \boldsymbol{U}^{-1}\boldsymbol{H}$ which take the following forms:

$$\check{\boldsymbol{D}} = \begin{bmatrix} \check{\boldsymbol{D}}_{(1)} \in \mathbb{R}^{R_1 \times R_1} & \check{\boldsymbol{D}}_{(2)} \in \mathbb{R}^{R_1 \times N - R_1} \\ \boldsymbol{0} \in \mathbb{R}^{N - R_1 \times R_1} & \check{\boldsymbol{D}}_{(3)} \in \mathbb{R}^{N - R_1 \times N - R_1} \end{bmatrix} \tag{3.18}$$

$$\check{\boldsymbol{H}} = \begin{bmatrix} \check{\boldsymbol{H}}_{(1)} \in \mathbb{R}^{R_1 \times L} \\ \boldsymbol{0} \in \mathbb{R}^{N - R_1 \times L} \end{bmatrix}. \tag{3.19}$$

3. Use the QR-decomposition to get $\check{\boldsymbol{D}}_{(1)} = \check{\boldsymbol{U}}_{(1)}\tilde{\boldsymbol{D}}_{(1)}$, where $\check{\boldsymbol{U}}_{(1)} \in \mathbb{R}^{R_1 \times R_1}$ is an orthogonal matrix, and $\tilde{\boldsymbol{D}}_{(1)} \in \mathbb{R}^{R_1 \times R_1}$ is an upper triangular matrix. The matrix $\check{\boldsymbol{U}}_{(1)}$ takes the following form:

$$\check{\boldsymbol{U}}_{(1)} = \begin{bmatrix} \check{\boldsymbol{U}}_{(11)} \in \mathbb{R}^{R_1 \times R_2} & \check{\boldsymbol{U}}_{(12)} \in \mathbb{R}^{R_1 \times R_1 - R_2} \end{bmatrix}, \tag{3.20}$$

where R_2 is the rank of $\check{\boldsymbol{D}}_{(11)}$.

4. Let $R_3 = \min\left\{s, \mathsf{Rank}\left\{\check{\boldsymbol{U}}_{(12)}^{\mathsf{T}}\check{\boldsymbol{H}}_{(1)}\right\}\right\}$. Find a set of R_3 independent rows of the matrix $\check{\boldsymbol{U}}_{(12)}^{\mathsf{T}}\check{\boldsymbol{H}}_{(1)}$, indexed by $\mathcal{T}$. Define $\bar{\boldsymbol{U}}_{(1)}$ by rearranging the columns of $\check{\boldsymbol{U}}_{(1)}$ as follows:

$$\bar{\boldsymbol{U}}_{(1)} \triangleq \begin{bmatrix} \check{\boldsymbol{U}}_{(11)} & \check{\boldsymbol{U}}_{(12)\mathcal{T}} & \check{\boldsymbol{U}}_{(12)\mathcal{T}^c} \end{bmatrix} \in \mathbb{R}^{R_1 \times R_1}, \tag{3.21}$$

where the matrices $\check{\boldsymbol{U}}_{(12)\mathcal{T}} \in \mathbb{R}^{R_3 \times L}$ and $\check{\boldsymbol{U}}_{(12)\mathcal{T}^c} \in \mathbb{R}^{R_1 - R_2 - R_3 \times L}$ are the submatrices of $\check{\boldsymbol{U}}_{(12)}$ with columns are indexed by $\mathcal{T}$ and $\mathcal{T}^c$, respectively. Define an invertible matrix $\bar{\boldsymbol{U}} \in \mathbb{R}^{N \times N}$ using some arbitrary invertible matrix $\bar{\boldsymbol{U}}_{(2)} \in \mathbb{R}^{N - R_1 \times N - R_1}$ as follows:

$$\bar{\boldsymbol{U}} \triangleq \begin{bmatrix} \bar{\boldsymbol{U}}_{(1)} \in \mathbb{R}^{R_1 \times R_1} & \boldsymbol{0} \in \mathbb{R}^{R_1 \times N - R_1} \\ \boldsymbol{0} \in \mathbb{R}^{N - R_1 \times R_1} & \bar{\boldsymbol{U}}_{(2)} \in \mathbb{R}^{N - R_1 \times N - R_1} \end{bmatrix}. \tag{3.22}$$

5. Compute $\bar{\boldsymbol{D}} = \bar{\boldsymbol{U}}^{-1}\check{\boldsymbol{D}}\bar{\boldsymbol{U}}$ and $\bar{\boldsymbol{H}} = \bar{\boldsymbol{U}}^{-1}\check{\boldsymbol{H}}$, which take the following forms:

$$\bar{\boldsymbol{D}} = \begin{bmatrix} \bar{\boldsymbol{D}}_{(1)} \in \mathbb{R}^{R_2+R_3 \times R_2+R_3} & \bar{\boldsymbol{D}}_{(2)} \\ \boldsymbol{0} \in \mathbb{R}^{N-R_2-R_3 \times R_2+R_3} & \bar{\boldsymbol{D}}_{(3)} \end{bmatrix} \tag{3.23}$$

$$\bar{\boldsymbol{H}} = \begin{bmatrix} \bar{\boldsymbol{H}}_{(1)} \in \mathbb{R}^{R_2+R_3 \times L} \\ \bar{\boldsymbol{H}}_{(2)} \in \mathbb{R}^{R_1-R_2-R_3 \times L} \\ \boldsymbol{0} \in \mathbb{R}^{N-R_1 \times L} \end{bmatrix}. \tag{3.24}$$

The $(R_2 + R_3)-$dimensional part corresponding to the matrix pair $(\bar{\boldsymbol{D}}_{(1)}, \bar{\boldsymbol{H}}_{(1)})$ is $s-$sparse-controllable, while the remaining part is $s-$sparse-uncontrollable. Also, since $\bar{\boldsymbol{D}} = \left(\boldsymbol{U}\bar{\boldsymbol{U}}\right)^{-1}\boldsymbol{D}\left(\boldsymbol{U}\bar{\boldsymbol{U}}\right)$ and $\bar{\boldsymbol{H}} = \left(\boldsymbol{U}\bar{\boldsymbol{U}}\right)^{-1}\boldsymbol{H}$, the new basis is $\boldsymbol{U}\bar{\boldsymbol{U}}$.

Here, in steps 1 and 2 are the same as the Kalman decomposition, and thus the R_1-dimensional part corresponding to $(\check{\boldsymbol{D}}_{(1)}, \check{\boldsymbol{H}}_{(1)})$ is controllable, while the part corresponding to $(\check{\boldsymbol{D}}_{(2)}, \boldsymbol{0})$ is uncontrollable. From the PBH test based conditions, we know that $(\check{\boldsymbol{D}}_{(1)}, \check{\boldsymbol{H}}_{(1)})$ satisfies condition 1 of Theorem 3.1.

Next, in steps 3 and step 4, we find a basis that separates the sparse-controllable part from the controllable part corresponding to $(\check{\boldsymbol{D}}_{(1)}, \check{\boldsymbol{H}}_{(1)})$, i.e., the part which satisfies condition 2 of Theorem 3.1. In step 4, since $R_3 \leq \mathsf{Rank}\left\{\check{\boldsymbol{U}}_{(12)}^{\mathsf{T}}\check{\boldsymbol{H}}_{(1)}\right\}$, we can always find R_3 linearly independent rows of $\check{\boldsymbol{U}}_{(12)}^{\mathsf{T}}\check{\boldsymbol{H}}_{(1)}$. After step 4, we have

$$\begin{bmatrix} \check{\boldsymbol{U}}_{(1)}^{-1}\check{\boldsymbol{D}}_{(1)}\check{\boldsymbol{U}}_{(1)} & \check{\boldsymbol{U}}_{(1)}^{-1}\check{\boldsymbol{H}}_{(1)} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{D}}_{(11)} & \check{\boldsymbol{U}}_{(11)}^{\mathsf{T}}\check{\boldsymbol{H}}_{(1)} \in \mathbb{R}^{R_2 \times R_1} \\ \boldsymbol{0} & \check{\boldsymbol{U}}_{(12)\mathcal{T}}^{\mathsf{T}}\check{\boldsymbol{H}}_{(1)} \in \mathbb{R}^{R_3 \times R_1} \\ \boldsymbol{0} & \check{\boldsymbol{U}}_{(12)\mathcal{T}^c}^{\mathsf{T}}\check{\boldsymbol{H}}_{(1)} \in \mathbb{R}^{(R_1-R_2-R_3) \times R_1} \end{bmatrix}, \tag{3.25}$$

since the rank of $\check{\boldsymbol{D}}_{(1)}$ has rank R_2. The first R_2 rows of the matrix are linearly independent, as $\bar{\boldsymbol{D}}_{(11)}$ has full row-rank. Further, we note that $\mathcal{T}$ is chosen such that it is the largest

index set such that $\left(\check{\boldsymbol{U}}_{(12)\mathcal{T}}\right)^{\mathsf{T}} \check{\boldsymbol{H}}_{(1)}$ has a submatrix with s columns and has rank as R_3. Thus, we get the following:

$$\max_{\substack{\mathcal{S}\subseteq\{1,2,\ldots,L\}\\ \|\mathcal{S}\|=s}} \mathsf{Rank}\left\{\begin{bmatrix}\check{\boldsymbol{U}}_{(1)}^{-1}\check{\boldsymbol{D}}_{(1)}\check{\boldsymbol{U}}_{(1)} & \check{\boldsymbol{U}}_{(1)}^{-1}\check{\boldsymbol{H}}_{(1)\mathcal{S}}\end{bmatrix}\right\} = \mathsf{Rank}\left\{\bar{\boldsymbol{D}}_{(11)}\right\} + \max_{\substack{\mathcal{S}\subseteq\{1,2,\ldots,L\}\\ \|\mathcal{S}\|=s}} \mathsf{Rank}\left\{\check{\boldsymbol{U}}_{(12)}^{\mathsf{T}}\check{\boldsymbol{H}}_{(1)\mathcal{S}}\right\}. \quad (3.26)$$

Further, we have

$$\begin{aligned}\max_{\substack{\mathcal{S}\subseteq\{1,2,\ldots,L\}\\ \|\mathcal{S}\|=s}} \mathsf{Rank}\left\{\begin{bmatrix}\check{\boldsymbol{U}}_{(1)}^{-1}\check{\boldsymbol{D}}_{(1)}\check{\boldsymbol{U}}_{(1)} & \check{\boldsymbol{U}}_{(1)}^{-1}\check{\boldsymbol{H}}_{(1)\mathcal{S}}\end{bmatrix}\right\} &= \mathsf{Rank}\left\{\bar{\boldsymbol{D}}_{(11)}\right\} + \min\left\{s, \mathsf{Rank}\left\{\check{\boldsymbol{U}}_{(12)}^{\mathsf{T}}\check{\boldsymbol{H}}_{(1)}\right\}\right\} && (3.27)\\ &= \mathsf{Rank}\left\{\bar{\boldsymbol{D}}_{(11)}\right\} + \mathsf{Rank}\left\{\check{\boldsymbol{U}}_{(12)\mathcal{T}}^{\mathsf{T}}\check{\boldsymbol{H}}_{(1)}\right\} && (3.28)\\ &= R_2 + R_3. && (3.29)\end{aligned}$$

Thus, condition 2 of Theorem 3.1 is satisfied by the reduced space of dimension $R_2 + R_3 \leq R_1$, and therefore, it is the sparse-controllable part. Also, since R_3 is nondecreasing in s, the dimension of the sparse-controllable part is also nondecreasing in s.

Finally, in step 5, we extend the basis $\bar{\boldsymbol{U}}_{(1)}$ obtained in step 4, to span $\mathbb{R}^N$. Overall, the basis for the sparse-controllability decomposition is $\boldsymbol{U}\bar{\boldsymbol{U}}$, and the dimension of the sparse-controllable part of the system is $R_2 + R_3 \leq R_1$.

We illustrate the decomposition procedure with the following example.

Example 3.4. *Consider a linear system with $N = 4$, $L = 3$, $s = 1$:*

$$\boldsymbol{D} = \begin{bmatrix} 5.65 & 0 & -1.25 & -7.95 \\ 3.3 & 0 & -0.9 & -4.7 \\ -0.55 & 0 & 0.35 & 0.85 \\ 3.4 & 0 & -0.8 & -4.8 \end{bmatrix} \tag{3.30}$$

$$\boldsymbol{H} = \begin{bmatrix} 0.25 & 1.25 & 1.5 \\ 0.25 & 1.25 & 1.5 \\ -0.5 & -0.75 & -1.25 \\ 0.25 & 1 & 1.25 \end{bmatrix}. \tag{3.31}$$

Following the above procedure, from step 1

$$\boldsymbol{U} = \begin{bmatrix} 1 & 0 & 4 & 1 \\ 2 & -1 & 3 & 0 \\ -2 & 0 & -1 & 1 \\ 1 & 0 & 3 & 0 \end{bmatrix}. \tag{3.32}$$

Step 2 gives the following with $R_1 = 3$:

$$\check{\boldsymbol{D}}_{(1)} = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \check{\boldsymbol{H}}_{(1)} = \begin{bmatrix} 0.25 & 0.25 & 0.5 \\ 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0.25 \end{bmatrix}. \tag{3.33}$$

In step 3, we get $R_2 = 1$, and

$$\check{\boldsymbol{U}}_{(11)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \check{\boldsymbol{U}}_{(12)} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.34}$$

In step 4, we compute $\check{\boldsymbol{U}}_{(12)}^{\mathsf{T}} \check{\boldsymbol{H}}_{(1)} = \begin{bmatrix} 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0.25 \end{bmatrix}$. *Thus, $R_3 = 1$, and $\mathcal{T} = \{1\}$ or*

$\{2\}$ *for* $s = 1$. *With* $\mathcal{T} = \{1\}$, *we get*

$$\bar{\boldsymbol{U}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{U}\bar{\boldsymbol{U}} = \begin{bmatrix} 1 & 4 & 0 & 1 \\ 2 & 3 & -1 & 0 \\ -2 & -1 & 0 & 1 \\ 1 & 3 & 0 & 0 \end{bmatrix}. \tag{3.35}$$

Finally, step 5 gives

$$\bar{\boldsymbol{D}}_{(1)} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0 \end{bmatrix}, \bar{\boldsymbol{H}}_{(1)} = \begin{bmatrix} 0.25 & 0.25 & 0.5 \\ 0 & 0.25 & 0.25 \end{bmatrix}, \tag{3.36}$$

which corresponds to the $1-$*sparse-controllable part of the system which has dimension* $R_2+R_3 = 2$. *It can be easily verified that the system defined using* $(\bar{\boldsymbol{D}}_{(1)}, \bar{\boldsymbol{H}}_{(1)})$ *is* $1-$*sparse-controllable using Theorem 3.1.*

3.6 Output Sparse-Controllability

The notion of controllability we discussed thus far has been concerned with state controllability: we analyzed the conditions for driving the system state from any initial state to any arbitrary final state using a finite number of sparse inputs. In this section, we extend our results to a variant of controllability called *output controllability*, which is related to transferring any initial state to any desired final output. Hence, we consider the following linear dynamical system:

$$\boldsymbol{x}_k = \boldsymbol{D}\boldsymbol{x}_{k-1} + \boldsymbol{H}\boldsymbol{h}_k \tag{3.37}$$

$$\boldsymbol{y}_k = \boldsymbol{A}\boldsymbol{x}_k, \tag{3.38}$$

where the output matrix $\boldsymbol{A} \in \mathbb{R}^{m \times N}$ with $m < N$, and the state evolution equation (3.37) is same as (3.1). Next, we define the notion of *output sparse-controllability* as follows:

Definition 3.2 (Output sparse-controllability)**.** *The system defined by* (3.37) *and* (3.38) *is called output* $s-$*sparse-controllable if for any given initial state* $\boldsymbol{x}_0$ *and any desired final output vector* $\boldsymbol{y}_K$*, there exists an input sequence* $\boldsymbol{h}_k$*,* $k = 1, 2, \cdots, K$ *such that* $\|\boldsymbol{h}_k\|_0 \leq s$*, which steers the system from initial state* $\boldsymbol{x}_0$ *to a final output* $\boldsymbol{y}_K$ *for some finite* K*.*

Now, to characterize the output sparse-controllability, we consider the following equivalent system of equations:

$$\boldsymbol{y}_K - \boldsymbol{A}\boldsymbol{D}^K \boldsymbol{x}_0 = \boldsymbol{A}\tilde{\boldsymbol{H}}_{(K)} \boldsymbol{h}_{(K)}. \tag{3.39}$$

In the following subsections, we extend the results in the previous sections to the output sparse-controllability case.

3.6.1 Necessary and Sufficient Conditions for Output Sparse-Controllability

We begin by stating necessary and sufficient conditions for output controllability of an *unconstrained* system. In [101], a Kalman test for output controllability is derived, which states that an unconstrained system given by (3.37) and (3.38) is output controllable if and only if the matrix $\boldsymbol{A}\tilde{\boldsymbol{H}}_{(K)}$ has full row rank for some finite K. However, a direct extension of this result to the case of output sparse-controllability leads to a combinatorial test, which is computationally expensive. Hence, we first derive a PBH-type test for output (unconstrained) controllability, which we present as the following proposition.

Proposition 3.2. *For an unconstrained system given by* (3.37) *and* (3.38)*, the system is output controllable only if the rank of the matrix* $\boldsymbol{A}\begin{bmatrix}\lambda \boldsymbol{I} - \boldsymbol{D} & \boldsymbol{H}\end{bmatrix} \in \mathbb{R}^{m \times (N+L)}$ *is* m *for*

all $\lambda \in \mathbb{C}$.

Proof. Our proof is by contradiction. Suppose that, the matrix $\boldsymbol{A}\begin{bmatrix}\lambda \boldsymbol{I} - \boldsymbol{D} & \boldsymbol{H}\end{bmatrix}$ does not have full row rank, for some $\lambda \in \mathbb{C}$. Then, there exists a $\mathbf{0} \neq \boldsymbol{z} \in \mathbb{C}^m$ such that

$$\boldsymbol{z}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{D} = \lambda \boldsymbol{z}^{\mathsf{T}}\boldsymbol{A} \text{ and } \boldsymbol{z}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{H} = \mathbf{0}, \tag{3.40}$$

which implies $\boldsymbol{z}^{\mathsf{T}}\boldsymbol{A}\tilde{\boldsymbol{H}}_{(K)} = \mathbf{0}$ for all K. Hence, the Kalman test is violated, and thus the system is not output controllable, as required. □

We note that the PBH test for output controllability only gives us a necessary condition for output controllability. We illustrate this using the following example:

Example 3.5. *Let* $m = 3$, $N = 5$ *and* $L = 3$, *and suppose the system given by* (3.37) *and* (3.38) *is defined by the following matrices:*

$$\boldsymbol{D} = \begin{bmatrix} 1 & 2 & 4 & 5 & 9 \\ 7 & 2 & 3 & 1 & 7 \\ 0 & 0 & 1 & 2 & 5 \\ 0 & 0 & 3 & 4 & 7 \\ 0 & 0 & 1 & 6 & 9 \end{bmatrix}, \quad \boldsymbol{H} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \textit{and} \quad \boldsymbol{A} = \begin{bmatrix} 0 & 0.019 & -0.14 & 0.02 & 0.99 \\ 0 & -0.08 & 0.24 & 0.97 & 0.018 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.41}$$

It can be verified that the system fails the Kalman test, as $\mathsf{Rank}\left\{\boldsymbol{A}\tilde{\boldsymbol{H}}_{(K)}\right\} < m$ *for all* K. *However, for all values of* $\lambda \in \mathbb{C}$, $\mathsf{Rank}\left\{\boldsymbol{A}\begin{bmatrix}\lambda \boldsymbol{I} - \boldsymbol{D} & \boldsymbol{H}\end{bmatrix}\right\} = 3 = m$. *Thus, the condition of the PBH test is necessary but not sufficient for output controllability.*

Before we present the main result, we note that the Kalman-type test for output (unconstrained) controllability [101] immediately extends to output sparse-controllability as

follows. The system is output controllable if and only if, for some finite K, there exists a submatrix of $\boldsymbol{A}\tilde{\boldsymbol{H}}_{(K)}$ with rank m of the following form:

$$\boldsymbol{A}\begin{bmatrix}\boldsymbol{D}^{K-1}\boldsymbol{H}_{\mathcal{S}_1} & \boldsymbol{D}^{K-2}\boldsymbol{H}_{\mathcal{S}_2} & \dots & \boldsymbol{H}_{\mathcal{S}_K}\end{bmatrix} \in \mathbb{R}^{m\times Ks},$$

such that the index set $\mathcal{S}_i \subseteq \{1, 2, \dots, L\}$ and $|\mathcal{S}_i| = s$, for $i = 1, 2, \dots, K$. Next, we extend Theorem 3.1 to the output sparse-controllability, as follows:

Theorem 3.4. *The system given by* (3.37) *and* (3.38) *is output $s-$sparse-controllable only if the following conditions are satisfied:*

1. *For all $\lambda \in \mathbb{C}$, rank of $\boldsymbol{A}\begin{bmatrix}\lambda\boldsymbol{I}-\boldsymbol{D} & \boldsymbol{H}\end{bmatrix} \in \mathbb{R}^{m\times(N+L)}$ is m.*

2. *There exists an index set $\mathcal{S} \subseteq \{1, 2, \dots, L\}$ with s entries such that rank of the matrix $\boldsymbol{A}\begin{bmatrix}\boldsymbol{D} & \boldsymbol{H}_{\mathcal{S}}\end{bmatrix} \in \mathbb{R}^{m\times(N+s)}$ is m.*

Proof. The result can be proved by using an approach similar to the proof of Theorem 3.1 given in Appendix B.1. We replace $\boldsymbol{z}$ in the third part of the proof with $\boldsymbol{Az}$ to show the necessity of the above conditions. □

Theorem 3.4 is the same as Theorem 3.1, except for a pre-multiplication with $\boldsymbol{A}$. We make the following observations:

- When $\boldsymbol{A} = \boldsymbol{I}$, Theorem 3.4 reduces to Theorem 3.1, as expected.

- We know that $\mathsf{Rank}\{\boldsymbol{AH}^*\} \leq \mathsf{Rank}\{\boldsymbol{A}\}$, for any matrix $\boldsymbol{H}^*$. Thus, if $\mathsf{Rank}\{\boldsymbol{A}\} < m$, the Kalman test for output sparse-controllability fails. Hence, the system is not output sparse-controllable.

- Suppose $\mathsf{Rank}\{\boldsymbol{A}\} = m$ for an $s-$sparse-controllable system. Invoking Sylvester's rank inequality [102], we get

$$m = \mathsf{Rank}\{\boldsymbol{A}\} + \mathsf{Rank}\{\boldsymbol{H}^*\} - N \leq \mathsf{Rank}\{\boldsymbol{A}\boldsymbol{H}^*\} \leq \mathsf{Rank}\{\boldsymbol{A}\} = m, \tag{3.42}$$

  where $\boldsymbol{H}^* \in \mathbb{R}^{N \times Ks}$ is the submatrix of $\tilde{\boldsymbol{H}}_{(K)}$ that satisfies the Kalman test for state sparse-controllability, for some finite K. Hence, the system is output $s-$sparse-controllable. Therefore, the conditions in Theorem 3.4 are less restrictive than those in Theorem 3.1, as the output dimension $m < N$, provided $\boldsymbol{A}$ has rank m.

From the last observation, we see that it is possible that the system is output $s-$sparse-controllable, even if it is *not* $s-$sparse-controllable, provided $\mathsf{Rank}\{\boldsymbol{A}\} = m$. We illustrate this using the following example.

Example 3.6. *Let $m = 2$, $N = 3$ and $L = 2$, and suppose the system given by* (3.37) *and* (3.38) *is defined by the following matrices:*

$$\boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{H} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{3.43}$$

It can be verified that the system is not $1-$sparse-controllable, but the system is output $1-$sparse-controllable.

From Theorem 3.4, we can derive a procedure similar to the one given in Section 3.3.1, to verify output sparse-controllability of a system. We omit it to avoid repetition.

3.6.2 Minimum Number of Input Vectors for Output Controllability

A bound on the smallest number of $s-$sparse input vectors that ensures output controllability is given the following theorem:

Theorem 3.5. *For an output $s-$sparse-controllable system, the minimum number of input vectors K^* required to steer any initial output to any final output satisfies*

$$\frac{m}{\min\left\{\mathsf{Rank}\left\{\boldsymbol{AH}\right\}, s\right\}} \leq K^* \leq \min\left\{q\left\lceil\frac{R_{\boldsymbol{H}}}{s}\right\rceil, m - R^*_{\boldsymbol{AH},s} + 1\right\} \leq m, \tag{3.44}$$

where $R^*_{\boldsymbol{AH},s} = \max\limits_{\substack{\mathcal{S}\subseteq\{1,2,\ldots,L\}\\|\mathcal{S}|=s}} \mathsf{Rank}\left\{\boldsymbol{AH}_{\mathcal{S}}\right\}$, $R_{\boldsymbol{H}}$ is the rank of $\boldsymbol{H}$, and q is the degree of the minimal polynomial of $\boldsymbol{D}$.

Proof. The proof is along similar lines as the proof in Appendix B.1. We start by premultiplying the right-hand side of (B.6) by $\boldsymbol{A}$ to get a modified definition of submatrices of $\boldsymbol{A}\tilde{\boldsymbol{H}}$, and then follow the same arguments as in Appendix B.1 to get the desired result. □

It is interesting to note that the bounds in Theorem 3.5 are smaller than those in Theorem 3.3. This is because the dimension of the output space, m, is smaller than that of the state space, N. Also, the above result depends only on the degree of the minimal polynomial of $\boldsymbol{D}$, and not on the degree of the minimal polynomial of $\boldsymbol{AD}$.

A relaxed version of the above result, similar to Corollary 3.3, is as follows:

Corollary 3.4. *For an $s-$sparse-controllable system, the minimum number of input vectors K^* required to steer any initial output to any other final output satisfies*

$$\frac{m}{\min\left\{\mathsf{Rank}\left\{\boldsymbol{AH}\right\}, s\right\}} \leq K^* \leq \min\left\{q\left\lceil\frac{R_{\boldsymbol{H}}}{s}\right\rceil, \mathsf{Rank}\left\{\boldsymbol{AD}\right\} + 1, m\right\}. \tag{3.45}$$

Theorem 3.5 provides bounds on the minimum number of input vectors to ensure output controllability without any constraints on the input, by substituting $s = L$. We get the following result:

Corollary 3.5. *For an output controllable system, the minimum number of input vectors K required to steer any initial output to any final output satisfies*

$$\frac{m}{\mathsf{Rank}\{\boldsymbol{AH}\}} \leq K \leq \min\{q, m - \mathsf{Rank}\{\boldsymbol{AH}\} + 1\} \leq m. \tag{3.46}$$

3.6.3 Change of Basis Property

Output sparse-controllability possesses invariance under a change of basis, similar to state sparse-controllability.

Proposition 3.3 (Invariance under change of basis)**.** *The linear system defined by the matrix tuple $(\boldsymbol{D}, \boldsymbol{H}, \boldsymbol{A})$ is output $s-$sparse-controllable if and only if the system defined by $(\boldsymbol{U}^{-1}\boldsymbol{D}\boldsymbol{U}, \boldsymbol{U}^{-1}\boldsymbol{H}, \boldsymbol{A}\boldsymbol{U})$ is output $s-$sparse-controllable for every nonsingular matrix $\boldsymbol{U} \in \mathbb{R}^{N\times N}$.*

Proof. We note that when $\boldsymbol{D}$ and $\boldsymbol{H}$ are replaced with $\boldsymbol{U}^{-1}\boldsymbol{D}\boldsymbol{U}$ and $\boldsymbol{U}^{-1}\boldsymbol{H}$ respectively, in (3.3), we get $\boldsymbol{U}^{-1}\tilde{\boldsymbol{H}}_{(K)}$ instead of $\tilde{\boldsymbol{H}}_{(K)}$. Further, replacing $\boldsymbol{A}$ with $\boldsymbol{A}\boldsymbol{U}$ does not change in rank criterion in the Kalman-type rank test for output sparse controllability. □

Unlike state controllability, the change of basis does not change the equivalent linear system of equations given by (3.39). However, designing a procedure for decomposing the output space into sparse-controllable and sparse-uncontrollable subspaces similar to the one proposed for sparse-controllability is non-trivial, and we defer it to future work.

3.7 Summary

This chapter presented the necessary and sufficient conditions for controllability of linear systems subject to sparsity constraints on the input. We derived two easily verifiable conditions equivalent the Kalman-type rank test for the sparse-controllability of the system. We also presented a simple procedure with polynomial complexity to verify the conditions of the theorem. Further, we bounded the minimum number of input vectors that ensures controllability. In addition, the sparse-controllability tests also led to a Kalman decomposition-like procedure for decomposing the system into sparse-controllable, controllable but sparse-uncontrollable and uncontrollable parts. Thus, we have addressed three important aspects of controllability of a system with sparse inputs. Finally, we extended our results on state controllability to the output controllability.

This chapter dealt with the first question Q1 on the existence of a sparse solution for the model SM2. In the next chapter, we proceed to the next question Q2 in Chapter 1 for the same sparsity model.

Chapter 4

Bayesian Recovery Algorithms for Jointly Sparse Control Inputs

Answering problem Q2 for SM2

In this chapter, we present recovery algorithms for the recovery of jointly sparse control inputs, i.e., we consider the special case when $\boldsymbol{D}$ is a diagonal matrix. Also, we are interested in designing a sequential (online) algorithm with low complexity. These type of algorithms are particularly useful in case of applications like wireless channel estimation. The measurements are processed one after another in a sequential fashion, without waiting for the complete input to arrive. Such algorithms require significantly lower computational and memory resources compared to their offline counterparts. Also, estimates of the sparse vectors become available after a fixed delay from the time observations arrive.

4.1 Background

In many applications, such as wireless channel tracking [3], radar signal processing [103, 104], and biomedical imaging [105–108], the goal is to recover a sequence of sparse vectors

that exhibit additional structure, such as a common support and temporal correlation. For example, successive instantiations of a time-varying wireless channel have the same power delay profile, and the nonzero coefficients of these instantiations are temporally correlated, and can be modeled using a first-order auto-regressive (AR) process. Exploiting this additional structure in the multiple measurement vectors (MMV) can improve the recovery performance, but at the cost of higher latency, memory, and computational complexity. Hence, the goal of this chapter is to develop algorithms that exploit the structure in the signal to reconstruct a sequence of sparse vectors. We are particularly interested in developing algorithms with low complexity and bounded latency.

In the literature, there are many offline recovery algorithm that exploits the temporal correlation across the sparse vectors [3, 42, 43]. These algorithms are offline in nature, i.e., they process the entire set of measurement vectors in a single batch. Hence, when the data set is large, these algorithms suffer from poor efficiency and scalability. On the other hand, *online* algorithms process small batches of the measurement vectors at a time and recover the sparse vectors sequentially, resulting in low-complexity implementations. Online algorithms offer the additional benefit of low latency between the measurement and estimation, which may be necessary in certain applications. For example, in a real-time broadband communication system with high data rate and high mobility, offline estimation of the wireless channel is infeasible.

Several sequential algorithms for sparse signal recovery have been presented in the literature [109–117]. An online algorithm for recovery for sparse signal with common support is presented in [109]. However, the algorithm does not account for the temporal correlation in the signal. A non-iterative modified OMP algorithm for sequential recovery of sparse

signals is described in [110] for the case when the coefficient in the autoregression is unity. A combination of Kalman filtering and dynamic programming is givem in [111]. This algorithm is slow because it runs l_1 optimization multiple times for every measurement vector. Another iterative sequential algorithm that decouples the support recovery step from the Kalman filtering-based amplitude estimation step is presented in [112]. However, the algorithm requires one to tune a number of parameters beforehand. An alternate iterative online algorithm that jointly estimates the amplitude and support is hierarchical Bayesian Kalman filtering [113]. This algorithm does not require one to tune many parameters, but suffers from high complexity. Another algorithm for the sequential recovery of sparse signals is dynamic sparse coding [114]. The algorithm executes an optimization procedure based on gradient descent, and is also iterative in nature.

The above discussed algorithms do not allow one to improve the current estimate using a small set of future measurements. For scenarios that often arise in communication related applications (e.g., wireless channel estimation), a small delay is allowed if the estimation performance can be improved. Therefore, we need to use a smoothing operation instead of a filtering operation, and then filtering becomes a special case of smoothing when the allowed delay is zero. We present two algorithms in this chapter: *iterative online* algorithm and *non-iterative online* algorithm. The iterative algorithm allows a bounded delay between the measurement and estimation by combining the Kalman smoothing and the SBL framework. The algorithm runs multiple rounds of the expectation-maximization (EM) procedure for every measurement vector. Next, we improve this algorithm to obtain a non-iterative algorithm which has simpler implementation with minimal resource requirements. The non-iterative online algorithm where as every measurement vector arrives,

we do not run an iterative procedure until convergence of some metric. The algorithm does one round of update using the measurement vector, and waits for the next measurement vector. We reiterte that both these algorithms do not require parameter tuning and allows a small delay between the measurement and estimation, for the reconstruction of temporally correlated sparse vectors with common support.

Our online algorithms are based on the SBL framework [22,41]. The SBL approach offers superior performance compared to other algorithms like l_1 minimization and OMP, and does not require one to tune the algorithm parameters. Moreover, it naturally extends to incorporate the temporal correlation structure in the signal model. However, its complexity and memory requirements increase with the number of measurements to be processed, which limits its practical application. Our algorithms overcome this drawback, and is computationally efficient, while retaining the good performance of SBL.

4.2 Problem Formulation

We consider a special case of LDS presented in Chapter 1, where $\boldsymbol{x}_0 \triangleq \mathbf{0}$ and $\boldsymbol{D} \in [0,1)^{N\times N}$ and $\boldsymbol{H} \in \mathbb{R}^{N\times L}$ are the known diagonal matrices. The system model is given by

$$\boldsymbol{x}_k = \boldsymbol{D}\boldsymbol{x}_{k-1} + \boldsymbol{H}\boldsymbol{h}_k \tag{4.1}$$

$$\boldsymbol{y}_k = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{w}_k, k = 1,2,\ldots \tag{4.2}$$

Here, $\boldsymbol{w}_k$ is a zero mean Gaussian distributed noise with a full rank covariance matrix $\boldsymbol{R}_k$. The number of measurements m is assumed to be smaller than the number of unknowns N which makes the system underdetermined. The unknown sequence of vectors $\{\boldsymbol{h}_k, k = 1,2,\ldots\}$ are sparse, i.e., the number of nonzero entries, S, is small compared to

the size of the vector, N. The $\boldsymbol{h}_k$ are simultaneously sparse, that is, they share a common support. This implies that the indices of the nonzero entries of all the sparse vectors coincide. Note that, in our model, the sparse vectors are temporally correlated, but because $\boldsymbol{D}$ and $\boldsymbol{H}$ are both assumed to be diagonal, there is no intra-vector correlation. Also, the support of $\boldsymbol{x}_k$ coincides with that of $\{\boldsymbol{h}_k\}_{k\in\mathbb{N}}$.

4.2.1 Estimation Objectives

The objective of this work is to estimate the sparse vectors on-the-fly, without storing all the measurement data and the corresponding measurement matrices. The maximum delay allowed between the measurement and estimation is $\Delta < \infty$, and therefore our goal is to recursively estimate $\boldsymbol{x}_k$ using the measurements up to time $k+\Delta$, denoted by $\boldsymbol{y}^{k+\Delta}$. Throughout the chapter, we use subscripts to denote the value of a variable at a particular time instant (e.g., $\boldsymbol{y}_k$ denotes the observation at time k), and superscripts to denote the sequence of observations up to a particular time instant $\big($e.g., $\boldsymbol{y}^\ell$ denotes the sequence of observations $\{\boldsymbol{y}_k, k = 1, 2, \ldots, \ell\}$).

We design an online scheme inspired by the SBL algorithm [22], [41]. The extension of SBL for the recovery of simultaneous sparse vectors imposes a common prior on the unknown vectors, namely, $\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma})$ [41]. The covariance matrix $\boldsymbol{\Gamma} \in \mathbb{R}_+^{N\times N}$ is a diagonal matrix with N hyperparameters $\boldsymbol{\gamma} \in \mathbb{R}_+^N$ along the diagonal. In SBL, we compute the ML estimate $\boldsymbol{\gamma}_{\mathrm{ML}}$ of $\boldsymbol{\gamma}$, which in turn gives the MAP estimate of the sparse vectors.

In the following subsections, we contrast the offline and online approaches to estimating the hyperparameters and sparse vectors, which serves to bring out the primary estimation objectives of this work. We start with the online case.

Online

Let $\boldsymbol{\gamma}^{k-1}$ denote the sequence of estimates of the hyperparameters $\boldsymbol{\gamma}$ till time $k-1$. At time k, we want to compute the estimate of the hyperparameter vector $\boldsymbol{\gamma}_k$, using $\boldsymbol{y}^{k+\Delta}$ and $\boldsymbol{\gamma}^{k-1}$. Since we do not want to store the complete set of past measurements, we recursively update $\boldsymbol{\gamma}_k$ using a small set of measurements $\{\boldsymbol{y}_t, t = k, k+1, \ldots, k+\Delta\}$ and $\boldsymbol{\gamma}_{k-1}$. The update rule for $\boldsymbol{\gamma}_k$ is discussed in Section 4.4.

Using $\boldsymbol{\gamma}_k$, the online estimate of $\boldsymbol{x}_k$ is computed as its conditional mean given $\boldsymbol{y}^{k+\Delta}$, with $\boldsymbol{\Gamma}_t$ as the covariance of $\boldsymbol{x}_t$ for $t = 1, 2, \ldots, k-1$, and $\boldsymbol{\Gamma}_k$ as the covariance of $\boldsymbol{x}_t$ for $t = k, k+1, \ldots, k+\Delta$. Mathematically,

$$\hat{\boldsymbol{x}}_k = \mathbb{E}\left\{\boldsymbol{x}_k | \boldsymbol{y}^{k+\Delta}; \boldsymbol{\gamma}^{k-1}, \boldsymbol{\gamma}_k\right\}. \tag{4.3}$$

The estimate $\hat{\boldsymbol{x}}_k$ is obtained using fixed interval Kalman smoothing on a data block of size $\Delta + 1$ [2]. That is, $\boldsymbol{x}_k$ is recursively updated using the set of measurement vectors $\{\boldsymbol{y}_t, t = k, k+1, \ldots, k+\Delta\}$ and $\boldsymbol{\gamma}_k$. Note that $\boldsymbol{\gamma}^{k-1}$ is not used in the estimation of $\boldsymbol{x}_k$.

We emphasize that, with the estimate of $\boldsymbol{\gamma}_k$ in hand, the estimation of $\boldsymbol{x}_k$ is a straightforward application of the Kalman filtering principle. The key contribution of this chapter is the development of a recursive, online technique for estimating $\boldsymbol{\gamma}_k$ and its convergence analysis. We next discuss the offline case.

Offline

In the offline setting, we find the ML estimate $\boldsymbol{\gamma}^{\text{OFF}}$ of $\boldsymbol{\gamma}$ given the entire sequence $\boldsymbol{y}^K$, where K denotes the total number of measurements [3, 41]. The estimation procedure is detailed in Section 4.2.2. The estimate of $\boldsymbol{x}_k$ is computed as its conditional mean given

$\boldsymbol{y}^K$, using $\mathsf{Diag}\left\{\boldsymbol{\gamma}^{\text{OFF}}\right\}$ as the signal covariance matrix. Mathematically,

$$\hat{\boldsymbol{x}}_k^{\text{OFF}} = \mathbb{E}\left\{\boldsymbol{x}_k | \boldsymbol{y}^K; \boldsymbol{\gamma}^{\text{OFF}}\right\}, \tag{4.4}$$

for $k = 1, 2, \ldots, K$. These estimates are computed efficiently using fixed interval Kalman smoothing on the data block $\boldsymbol{y}^K$.

Thus, the primary goal in both the offline and online algorithms is the estimation of $\boldsymbol{\gamma}$. In the offline case, a single estimate of $\boldsymbol{\gamma}$ is computed using the entire set of observations. In the online version, a sequence of estimates are computed using small batches of observations, and in a recursive manner.

In the next subsection, we first describe the offline SBL algorithm for the correlated MMV problem, which we refer to as the *offline Kalman MMV SBL (KM-SBL)* algorithm [3].

4.2.2 Offline KM-SBL Algorithm

The offline algorithm uses the expectation-maximization (EM) procedure, which treats the unknowns $\boldsymbol{x}^K$ as the hidden data and the observations $\boldsymbol{y}^K$ as the known data. The EM procedure iterates between two steps: an expectation step (E-step) and a maximization step (M-step). Let $\boldsymbol{\gamma}^{(r-1)}$ be the estimate of $\boldsymbol{\gamma}$ at the r^{th} iteration.[1] The E-step computes $Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right)$, which is the marginal log-likelihood of the observed data. The M-step computes the hyperparameters that maximize $Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right)$.

$$\begin{aligned} \textbf{E-step: } Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right) &= \mathbb{E}_{\boldsymbol{x}^K|\boldsymbol{y}^K;\boldsymbol{\gamma}^{(r-1)}}\left\{\log p\left(\boldsymbol{y}^K, \boldsymbol{x}^K; \boldsymbol{\gamma}\right)\right\} \\ \textbf{M-step: } \boldsymbol{\gamma}^{(r)} &= \underset{\boldsymbol{\gamma} \in \mathbb{R}_+^{N\times 1}}{\arg\max}\; Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right). \end{aligned} \tag{4.5}$$

[1] For ease of notation, we omit the superscript OFF here.

Simplifying $Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right)$ we get,

$$Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right) = c_K - \frac{K}{2}\log|\boldsymbol{\Gamma}| - \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{\Gamma}^{-1}\boldsymbol{C}_{1|K,\boldsymbol{\gamma}^{(r-1)}}\right\} - \frac{1}{2}\sum_{t=2}^{K}\mathsf{Tr}\left\{\boldsymbol{\Gamma}^{-1}\left(\boldsymbol{I}-\boldsymbol{D}^2\right)^{-1}\boldsymbol{T}_{t|K,\boldsymbol{\gamma}^{(r-1)}}\right\}. \quad (4.6)$$

where the constant c_K is independent of $\boldsymbol{\gamma}$, and the $N \times N$ matrices are defined as follows:

$$\boldsymbol{T}_{t|K,\boldsymbol{\gamma}^{(r-1)}} \triangleq \boldsymbol{C}_{t|K,\boldsymbol{\gamma}^{(r-1)}} + \boldsymbol{D}\boldsymbol{C}_{t-1|K,\boldsymbol{\gamma}^{(r-1)}}\boldsymbol{D} - 2\boldsymbol{D}\boldsymbol{C}_{t,t-1|K,\boldsymbol{\gamma}^{(r-1)}} \quad (4.7)$$

$$\boldsymbol{C}_{t|K,\boldsymbol{\gamma}^{(r-1)}} \triangleq \boldsymbol{P}_{t|K,\boldsymbol{\gamma}^{(r-1)}} + \widehat{\boldsymbol{x}}_{t|K,\boldsymbol{\gamma}^{(r-1)}}\widehat{\boldsymbol{x}}^{\mathsf{T}}_{t|K,\boldsymbol{\gamma}^{(r-1)}} \quad (4.8)$$

$$\boldsymbol{C}_{t,t-1|K,\boldsymbol{\gamma}^{(r-1)}} \triangleq \boldsymbol{P}_{t,t-1|K,\boldsymbol{\gamma}^{(r-1)}} + \widehat{\boldsymbol{x}}_{t|K,\boldsymbol{\gamma}^{(r-1)}}\widehat{\boldsymbol{x}}^{\mathsf{T}}_{t-1|K,\boldsymbol{\gamma}^{(r-1)}}, \quad (4.9)$$

for $t \leq K$. Here, the mean $\widehat{\boldsymbol{x}}_{t|K,\boldsymbol{\gamma}^{(r-1)}} \triangleq \mathbb{E}\left\{\boldsymbol{x}_t|\boldsymbol{y}^K; \boldsymbol{\gamma}^{(r-1)}\right\}$; and the covariance $\boldsymbol{P}_{t|K,\boldsymbol{\gamma}^{(r-1)}}$ and the cross-covariance $\boldsymbol{P}_{t,t-1|K,\boldsymbol{\gamma}^{(r-1)}}$ are defined as

$$\boldsymbol{P}_{t|K,\boldsymbol{\gamma}^{(r-1)}} \triangleq \mathbb{E}\left\{\tilde{\boldsymbol{x}}_t\tilde{\boldsymbol{x}}_t^{\mathsf{T}}\left|\boldsymbol{y}^K; \boldsymbol{\gamma}^{(r-1)}\right.\right\} \quad (4.10)$$

$$\boldsymbol{P}_{t,t-1|K,\boldsymbol{\gamma}^{(r-1)}} \triangleq \mathbb{E}\left\{\tilde{\boldsymbol{x}}_t\tilde{\boldsymbol{x}}_{t-1}^{\mathsf{T}}\left|\boldsymbol{y}^K; \boldsymbol{\gamma}^{(r-1)}\right.\right\}, \quad (4.11)$$

where $\tilde{\boldsymbol{x}}_t = \boldsymbol{x}_t - \widehat{\boldsymbol{x}}_{t|K,\boldsymbol{\gamma}^{(r-1)}}$. The calculation of the variables $\widehat{\boldsymbol{x}}_{t|K,\boldsymbol{\gamma}^{(r-1)}}$, $\boldsymbol{P}_{t|K,\boldsymbol{\gamma}^{(r-1)}}$, and $\boldsymbol{P}_{t,t-1|K,\boldsymbol{\gamma}^{(r-1)}}$ is implemented using fixed interval Kalman smoothing [2]. Maximizing $Q\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(r-1)}\right)$ with respect to $\boldsymbol{\gamma}$, we get the following M-step:

$$\boldsymbol{\gamma}^{(r)} = \frac{1}{K}\mathsf{Diag}\left\{\left(\boldsymbol{I}-\boldsymbol{D}^2\right)^{-1}\sum_{t=2}^{K}\boldsymbol{T}_{t|K,\boldsymbol{\gamma}^{(r-1)}} + \boldsymbol{C}_{1|K,\boldsymbol{\gamma}^{(r-1)}}\right\}. \quad (4.12)$$

We note that the latency in estimating $\boldsymbol{x}_K$ is 0, that of $\boldsymbol{x}_{K-1}$ is 1, and so on. Hence, the average latency of the offline KM-SBL algorithm is $\frac{1}{K}\sum_{t=1}^{K}(K-t) = (K-1)/2$. We now present our online algorithm.

4.3 Iterative Online Algorithm Development

4.3.1 Fixed Lag KSBL

The natural extension of the offline KSBL to partial data is to run the forward and backward recursions on the available data at each time, as each new measurement vector arrives. However, this approach requires storage of all forward variables, and it is computationally expensive. Hence, we propose to estimate the hyperparameters using data in a small fixed time window of size $\Delta + 1$, and produce smoothed estimates of all sparse vectors in that window. This leads us to the fixed lag Kalman smoothing, which estimates the state in a sliding window of time [118]. Note that the conventional fixed lag smoothing algorithms assume complete knowledge of the signal statistics, i.e., $\mathbf{\Gamma}$, whereas here we have to adaptively estimate $\mathbf{\Gamma}$ while computing the smoothed estimates. We combine fixed lag Kalman smoothing with the EM algorithm to learn the hyperparameter $\mathbf{\Gamma}$ in the M-step, and perform fixed lag filtering to estimate the sparse vector in the E-step.

When $\boldsymbol{y}_k$ arrives, the algorithm estimates the hyperparameter $\mathbf{\Gamma}$ using the EM algorithm using data over a time window of length $\Delta + 1$. The E-step and the M-step for the fixed lag smoothing scheme are

$$\textbf{E-step: } \mathcal{Q}(\mathbf{\Gamma}|\mathbf{\Gamma}_k^{(r-1)}) = \mathbb{E}\left\{\log\left[p\left(\boldsymbol{Y}_{k-\Delta:k}, \boldsymbol{X}_{k-\Delta:k}\right)\right]\right\}$$

$$\textbf{M-step: } \mathbf{\Gamma}_k^{(r)} = \underset{\boldsymbol{\gamma}\in\mathbb{R}_+^{N\times 1}; \mathbf{\Gamma}=\mathsf{Diag}\{\boldsymbol{\gamma}\}}{\arg\max} \mathcal{Q}(\mathbf{\Gamma}|\mathbf{\Gamma}^{(r-1)}),$$

where the expectation operation is over the distribution of $\boldsymbol{X}_{k-\Delta:k}$ conditioned on $\boldsymbol{Y}_{k-\Delta:k}$ and parameterized $\mathbf{\Gamma}_k^{(r-1)}$. Similar to the offline KSBL, the E-step involves computation of the posterior statistics of the sparse vectors over the time window. We employ fixed

Algorithm 1 E-Step of Fixed Lag KSBL at time k in the r^{th} EM iteration

Input: $\boldsymbol{y}_k, \boldsymbol{A}_k, \boldsymbol{\Gamma}_k^{(r-1)}$
  Initialization: $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_k^{(r-1)}$
  Prediction:
  $\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{D}\hat{\boldsymbol{x}}_{k-1|k-1}$
  $\boldsymbol{P}_{k|k-1} = \boldsymbol{D}\boldsymbol{P}_{k-1|k-1}\boldsymbol{D} + \boldsymbol{\Gamma}(\boldsymbol{I}_N - \boldsymbol{D}^2)$
  $\boldsymbol{J}_k = \boldsymbol{A}_k^{\mathsf{T}} \left(\boldsymbol{A}_k \boldsymbol{P}_{k|k-1} \boldsymbol{A}_k^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}$
  $\boldsymbol{v}_k = \boldsymbol{y}_k - \boldsymbol{A}_k \hat{\boldsymbol{x}}_{k|k-1}$
  for $t = 0, 1, 2, \ldots, \Delta$ **do**
    Smoothing:
    $\boldsymbol{G}_k^{(t)} = \boldsymbol{P}_{k,k-t|k-1}^{\mathsf{T}} \boldsymbol{J}_k$
    $\hat{\boldsymbol{x}}_{k-t|k} = \hat{\boldsymbol{x}}_{k-t|k-1} + \boldsymbol{G}_k^{(t)} \boldsymbol{v}_k$
    $\boldsymbol{P}_{k-t|k} = \boldsymbol{P}_{k-t|k-1} - \boldsymbol{G}_k^{(t)} \boldsymbol{A}_k \boldsymbol{P}_{k,k-t|k-1}$
    if $t \neq \Delta$ **then**
      $\boldsymbol{P}_{k-t,k-t-1|k} = \left(\boldsymbol{I}_N - \boldsymbol{G}_k^{(t)} \boldsymbol{A}_k\right) \boldsymbol{P}_{k,k-t-1|k-1}$
    end if
    $\boldsymbol{P}_{k+1,k-t|k} = \boldsymbol{D}\left(\boldsymbol{I}_N - \boldsymbol{G}_k^{(0)} \boldsymbol{A}_k\right) \boldsymbol{P}_{k,k-t|k-1}$
  end for

lag smoothing to estimate these, and the pseudocode is given in Algorithm 1, where $\boldsymbol{\Gamma}_k^{(0)} = \boldsymbol{\Gamma}_{k-1}$ and $\boldsymbol{\Gamma}_1^{(0)} = \boldsymbol{I}_N$. At the start of algorithm ($k = 1$), we initialize all vectors and matrices except $\boldsymbol{P}_{1|0}$ with all zero entries, and $\boldsymbol{P}_{1|0} = \boldsymbol{I}_N$. For each successive k, the variables are initialized with the estimates obtained in the previous iteration. The M-step in the r^{th} iteration can be simplified to a closed form expression for the new estimate of hyperparameters given by

$$\mathsf{Diag}\left\{\boldsymbol{\Gamma}_k^{(r)}\right\} = \frac{1}{\Delta+1}\mathsf{Diag}\left\{\left(\boldsymbol{I}_N - \boldsymbol{D}^2\right)^{-1} \sum_{t=k-\Delta+1}^{k} \boldsymbol{T}_{t|k;\boldsymbol{\Gamma}_k^{(r-1)}} + \boldsymbol{C}_{k-\Delta|k;\boldsymbol{\Gamma}_k^{(r-1)}}\right\}. \quad (4.13)$$

The proof is similar to that for the offline KSBL [3], and hence omitted. The E-step and the M-step are repeated until convergence of $\boldsymbol{\Gamma}_k$, and upon convergence the algorithm outputs the estimate of $\boldsymbol{x}_{k-\Delta}$ as $\hat{\boldsymbol{x}}_{k-\Delta|k}$.

4.3.2 Sawtooth Lag KSBL

Although our fixed lag KSBL possess low computational complexity and storage, our numerical experiments show that the performance of the fixed lag comes close to that of offline KSBL only for large number of measurements K. The reason for the poor performance of the fixed lag filter is that it has only one step of forward recursion and Δ steps of backward recursions. This results in using different values of covariance matrix estimates $\boldsymbol{\Gamma}$ for updating the estimate of state statistics at each value of t, i.e., $\{\hat{\boldsymbol{x}}_t, \boldsymbol{P}_t, \boldsymbol{P}_{t,t-1}\}$, in different rounds of the EM iterations. In the initial part of the algorithm, the estimates of $\boldsymbol{\Gamma}$ in consecutive rounds of EM algorithm could possibly have different support. Hence, the support recovery performance of the algorithm is poor when K is small. In this section, we present another online scheme for the same recovery problem using a smoothing scheme known as the sawtooth lag filter, which uses the same value of $\boldsymbol{\Gamma}$ for computing the state statistics at any given t [119]. The sawtooth lag scheme is a compromise between the fixed interval and the fixed lag smoothing. Here, the fixed interval smoothing is run on overlapping blocks of data. In the E-step, the forward and backward recursions are run on a data block of size $\Delta + 1$, as given in Algorithm 2. The M-step is given by (4.13). Upon convergence of the EM procedure, the algorithm outputs the estimates of sparse vectors corresponding to the first $\bar{\Delta} < \Delta$ indices in the block, i.e., at time k, the estimates at times $t = k - \Delta, \ldots, k - \Delta + \bar{\Delta}$ are declared as $\hat{\boldsymbol{x}}_{t|k}$. Then, algorithm waits for next set of $\bar{\Delta}$ measurement vectors before proceeding further, shifting the time window by $\bar{\Delta}$.

Unlike the previous algorithm, the sawtooth lag KSBL waits for a block of data of size $\bar{\Delta}$, which is typically much smaller than the total number of observations. The EM procedure is run at times $k = i\bar{\Delta} + \Delta + 1, i = 0, 1, \ldots$ and when i^{th} EM procedure converges estimates

Algorithm 2 E-Step of Sawtooth Lag KSBL at time $k = i\bar{\Delta} + \Delta + 1$ in the r^{th} iteration

Input: $\boldsymbol{Y}_{k-\Delta:k}, \boldsymbol{A}_{k-\Delta:k}, \boldsymbol{\Gamma}_i^{(r-1)}$
  Initialization: $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_i^{(r-1)}$
  for $t = k - \Delta, \ldots, k$ **do**
    Prediction:
    $\hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{D}\hat{\boldsymbol{x}}_{t-1|t-1}$
    $\boldsymbol{P}_{t|t-1} = \boldsymbol{D}\boldsymbol{P}_{t-1|t-1}\boldsymbol{D} + \boldsymbol{\Gamma}(\boldsymbol{I}_N - \boldsymbol{D}^2)$
    Filtering:
    $\boldsymbol{G}_t = \boldsymbol{P}_{t|t-1}\boldsymbol{A}_t^{\mathsf{T}}\left(\boldsymbol{R} + \boldsymbol{A}_t\boldsymbol{P}_{t|t-1}\boldsymbol{A}_t^{\mathsf{T}}\right)^{-1}$
    $\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \boldsymbol{G}_t(\boldsymbol{y}_t - \boldsymbol{A}_t\hat{\boldsymbol{x}}_{t|t-1})$
    $\boldsymbol{P}_{t|t} = (\boldsymbol{I}_N - \boldsymbol{G}_t\boldsymbol{A}_t)\boldsymbol{P}_{t|t-1}$
  end for
  $\boldsymbol{P}_{k,k-1|k} = (\boldsymbol{I}_N - \boldsymbol{G}_k\boldsymbol{A}_k)\,\boldsymbol{D}\boldsymbol{P}_{k-1|k-1}$
  for $t = k, k-1, \ldots, k - \Delta + 1$ **do**
    Smoothing:
    $\boldsymbol{J}_{t-1} = \boldsymbol{P}_{t-1|t-1}\boldsymbol{D}\boldsymbol{P}_{t|t-1}^{-1}$
    $\hat{\boldsymbol{x}}_{t-1|K} = \hat{\boldsymbol{x}}_{t-1|t-1} + \boldsymbol{J}_{t-1}(\hat{\boldsymbol{x}}_{t|K} - \hat{\boldsymbol{x}}_{t|t-1})$
    $\boldsymbol{P}_{t-1|K} = \boldsymbol{P}_{t-1|t-1} + \boldsymbol{J}_{t-1}(\boldsymbol{P}_{t|K} - \boldsymbol{P}_{t|t-1})\boldsymbol{J}_{t-1}^{\mathsf{T}}$
    if $t \neq \Delta$ **then**
      $\boldsymbol{P}_{t,t-1|k} = \boldsymbol{P}_{t|t}\boldsymbol{J}_{t-1}^{\mathsf{T}} + \boldsymbol{J}_t\left(\boldsymbol{P}_{t+1,t|k} - \boldsymbol{D}\boldsymbol{P}_{t|t}\right)\boldsymbol{J}_{t-1}^{\mathsf{T}}$
    end if
  end for

of sparse vectors at time $t = i\bar{\Delta}+1, i\bar{\Delta}+2, \ldots, (i+1)\bar{\Delta}$, are declared. Hence, the latency in estimation is not fixed, but varies between $\Delta - \bar{\Delta} + 1$ and Δ; the average latency is $\Delta - (\bar{\Delta} - 1)/2$. As in the previous case, at the start of algorithm ($i = 0$), we initialize the algorithm with $\boldsymbol{\Gamma}_0^{(0)} = \boldsymbol{I}_N$, $\hat{\boldsymbol{x}}_{1|0} = \boldsymbol{0}_N$, and $\boldsymbol{P}_{1|0} = \boldsymbol{I}_N$. For each successive i, the variables are initialized with the estimates obtained in the previous iteration. Based on our numerical experiments, the computational and storage demand of the sawtooth lag KSBL is comparable to that of the fixed lag KSBL, while its the performance is close to that of the offline KSBL. This happens because the sawtooth lag scheme has both forward and backward recursions, whereas the fixed lag scheme has only backward computations. Thus, it combines the best of both the fixed lag and the fixed interval procedures. *Remark:* Sawtooth lag smoothing reduces to the optimal offline KSBL if when $\bar{\Delta} = \Delta = K$.

Typically, $\bar{\Delta}$ is chosen as $\Delta/2$ or smaller [119].

4.3.3 Complexity Analysis

Computational Cost

We assume that the multiplication of a $p \times q$ matrix with a $q \times r$ matrix is of the order pqr flops, and inversion of a $p \times p$ matrix is of the order p^3 flops [120]. Also, we assume that $m \ll N$, and for simplicity, we neglect lower order terms involved in computational complexity. We also note that the overall computational complexity of the fixed lag smoothing scheme and the sawtooth lag smoothing scheme scale with the number of observation vectors K, but the complexity per EM iteration is independent of K. However, simulation results show that the overall run time of our online algorithms is much smaller then the offline algorithm.

Memory Requirement

In the case of the offline algorithm, we need to save all forward variables, which demands memory that scales with K. For the fixed lag and the sawtooth lag smoothing schemes, data is processed over a small time window. Thus, the memory requirements do not scale with K, a primary advantage of our online algorithms. The variables that need to be stored are the statistics of the sparse vectors, which is of the order N^2.

We compare the computational demands and memory requirements of the three algorithms in Table 4.1. Next, we present an improved version of the above algorithms which demands lesser computational resources.

Smoothing	Computational cost per EM iteration	Memory Requirement	Average latency
Offline scheme	$\mathcal{O}(KN^3)$	$\mathcal{O}(KN^2)$	$(K-1)/2$
Fixed Lag	$\mathcal{O}(\Delta N^2 m)$	$\mathcal{O}(\Delta N^2)$	Δ
Sawtooth Lag	$\mathcal{O}(\Delta N^3)$	$\mathcal{O}(\Delta N^2)$	$\Delta - (\bar{\Delta}-1)/2$

Table 4.1: Comparison of online schemes against offline scheme when K observations are available

4.4 Non-iterative Online Algorithm Development

In the non-iterative version of KM-SBL, we process the data sequentially, without waiting for the complete input to arrive or storing all the data that has already arrived. Since we do not store data, it is not feasible to compute the mean $\widehat{\boldsymbol{x}}_{t|K}$,[2] the covariance $\boldsymbol{P}_{t|K}$, and the cross-covariance $\boldsymbol{P}_{t,t-1|K}$. Instead, we approximate them with $\widehat{\boldsymbol{x}}_{t|t+\Delta}$, $\boldsymbol{P}_{t|t+\Delta}$, and $\boldsymbol{P}_{t,t-1|t+\Delta}$, respectively. Then,

$$Q_k\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{k-1}\right) \approx a_k - \frac{k}{2}\log|\boldsymbol{\Gamma}| - \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{\Gamma}^{-1}\boldsymbol{C}_{1|\Delta}\right\} - \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{\Gamma}^{-1}\left(\boldsymbol{I}-\boldsymbol{D}^2\right)^{-1}\sum_{t=2}^{k}\boldsymbol{T}_{t|t+\Delta}\right\}, \quad (4.14)$$

where the constant a_k is independent of $\boldsymbol{\gamma}$.

Maximizing $Q_k\left(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{k-1}\right)$ with respect to $\boldsymbol{\gamma}$, we have the following recursion

$$\boldsymbol{\gamma}_k = \frac{1}{k}\mathsf{Diag}\left\{\left(\boldsymbol{I}-\boldsymbol{D}^2\right)^{-1}\sum_{t=2}^{k}\boldsymbol{T}_{t|t+\Delta} + \boldsymbol{C}_{1|\Delta}\right\} \quad (4.15)$$

$$= \boldsymbol{\gamma}_{k-1} + \frac{1}{k}\mathsf{Diag}\left\{\left(\boldsymbol{I}-\boldsymbol{D}^2\right)^{-1}\boldsymbol{T}_{k|k+\Delta} - \boldsymbol{\Gamma}_{k-1}\right\}. \quad (4.16)$$

Thus, $\boldsymbol{\gamma}_k$ can be estimated using $\boldsymbol{\gamma}_{k-1}$ and $\boldsymbol{T}_{k|k+\Delta}$. We next present a procedure to

[2]For brevity, we drop $\boldsymbol{\gamma}$ from the subscript.

recursively estimate $\boldsymbol{T}_{k|k+\Delta}$.

4.4.1 Implementation of the Algorithm

In order to compute $\boldsymbol{T}_{k|k+\Delta}$, we need to recursively update the mean $\widehat{\boldsymbol{x}}_{k|k+\Delta}$, the auto-covariance $\boldsymbol{P}_{k|k+\Delta}$, and the cross-covariance $\boldsymbol{P}_{k,k-1|k+\Delta}$. We describe two implementations: a fixed lag scheme and a sawtooth lag scheme.

Fixed Lag Scheme

We consider a Kalman filter designed for the following state space model with state variables as $\boldsymbol{x}_k$ and measurement variables as $\widetilde{\boldsymbol{y}}_k \triangleq \boldsymbol{y}_{k+\Delta}$. From (4.1),

$$\widetilde{\boldsymbol{y}}_k = \boldsymbol{A}_{k+\Delta}\boldsymbol{D}^{\Delta}\boldsymbol{x}_k + \boldsymbol{A}_{k+\Delta}\sum_{i=0}^{\Delta-1}\boldsymbol{D}^i\boldsymbol{z}_{k+\Delta-i} + \boldsymbol{w}_{k+\Delta} = \widetilde{\boldsymbol{A}}_k\boldsymbol{x}_k + \widetilde{\boldsymbol{w}}_k, \tag{4.17}$$

where $\widetilde{\boldsymbol{A}}_k \triangleq \boldsymbol{A}_{k+\Delta}\boldsymbol{D}^{\Delta}$ and $\widetilde{\boldsymbol{w}}_k \sim \mathcal{N}\left(0, \widetilde{\boldsymbol{R}}_k\right)$. Since the covariance of $\boldsymbol{z}_{k+\Delta-i}$ is $(\boldsymbol{I}-\boldsymbol{D}^2)\boldsymbol{\Gamma}$, it is easy to show that

$$\widetilde{\boldsymbol{R}}_k = \boldsymbol{A}_{k+\Delta}\left(\boldsymbol{I}-\boldsymbol{D}^{2\Delta}\right)\boldsymbol{\Gamma}\boldsymbol{A}_{k+\Delta}^{\mathsf{T}} + \boldsymbol{R}_{k+\Delta}. \tag{4.18}$$

The new state space model is given by (4.1) and (4.17). The Kalman filter equations for the new system are given below:

$$\widehat{\boldsymbol{x}}_{k|k+\Delta-1} = \boldsymbol{D}\widehat{\boldsymbol{x}}_{k-1|k+\Delta-1} \tag{4.19}$$

$$\boldsymbol{P}_{k|k+\Delta-1} = \boldsymbol{D}\boldsymbol{P}_{k-1|k+\Delta-1}\boldsymbol{D} + \left(\boldsymbol{I}-\boldsymbol{D}^2\right)\boldsymbol{\Gamma} \tag{4.20}$$

$$\boldsymbol{J}_k = \boldsymbol{P}_{k|k+\Delta-1}\widetilde{\boldsymbol{A}}_k^{\mathsf{T}}\left(\widetilde{\boldsymbol{A}}_k\boldsymbol{P}_{k|k+\Delta-1}\widetilde{\boldsymbol{A}}_k^{\mathsf{T}} + \widetilde{\boldsymbol{R}}_k\right)^{-1} \tag{4.21}$$

$$\widehat{\boldsymbol{x}}_{k|k+\Delta} = (\boldsymbol{I}-\boldsymbol{J}_k\widetilde{\boldsymbol{A}}_k)\widehat{\boldsymbol{x}}_{k|k+\Delta-1} + \boldsymbol{J}_k\boldsymbol{y}_{k+\Delta} \tag{4.22}$$

$$\boldsymbol{P}_{k|k+\Delta} = (\boldsymbol{I} - \boldsymbol{J}_k \widetilde{\boldsymbol{A}}_k)\boldsymbol{P}_{k|k+\Delta-1} \tag{4.23}$$

$$\boldsymbol{P}_{k,k-1|k+\Delta} = (\boldsymbol{I} - \boldsymbol{J}_k \widetilde{\boldsymbol{A}}_k)\boldsymbol{D}\boldsymbol{P}_{k-1|k+\Delta-1}. \tag{4.24}$$

As every measurement vector $\boldsymbol{y}_{k+\Delta}$ arrives, the algorithm updates $\boldsymbol{\gamma}$ using (4.16). Then, the online estimate of $\boldsymbol{x}_k$ can be computed using forward and backward recursions of a fixed interval Kalman smoother on the block of data of size $\Delta+1$, at times $t = k, k+1, \ldots, k+\Delta$, as described in Section 4.2.1.

Remark: The above scheme is not applicable when $\boldsymbol{D} = \boldsymbol{0}$ and $\Delta > 0$, because $\boldsymbol{y}_{k+\Delta}$ is independent of $\boldsymbol{x}_k$ in this case. Also, the fixed lag scheme only uses the latest measurement vector to update $\boldsymbol{\gamma}$, while one can achieve better performance by using all the available measurements in a window around the time instant of interest. In the following subsection, we present a sawtooth lag scheme that addresses the above issues.

Sawtooth Lag Scheme

In this scheme, we update $\boldsymbol{\gamma}$ as every data block of size $\bar{\Delta} \leq \Delta + 1$ arrives. Consider $k \in [k_l + 1, k_l + \bar{\Delta}]$ where $k_l \triangleq (l-1)\bar{\Delta}$ for the update index $l = 1, 2, \ldots$. We replace the fixed lag variables $\widehat{\boldsymbol{x}}_{k|k+\Delta}$, $\boldsymbol{P}_{k|k+\Delta}$, and $\boldsymbol{P}_{k,k-1|k+\Delta}$ with variables $\widehat{\boldsymbol{x}}_{k|\check{k}_l}$, $\boldsymbol{P}_{k|\check{k}_l}$, and $\boldsymbol{P}_{k,k-1|\check{k}_l}$, respectively, where $\check{k}_l \triangleq k_l + \Delta + 1$. We compute these variables using the estimate of $\boldsymbol{\gamma}$ obtained in the previous update, $\boldsymbol{\gamma}_{l-1}$. For the l^{th} update, (4.15) modifies to

$$\begin{aligned}\boldsymbol{\gamma}_l &= \frac{1}{k_{l+1}}\mathsf{Diag}\left\{\left(\boldsymbol{I} - \boldsymbol{D}^2\right)^{-1}\sum_{i=1}^{l}\sum_{\substack{t=k_i+1,\\ t\neq 1}}^{k_{i+1}} \boldsymbol{T}_{t|\check{k}_i} + \boldsymbol{C}_{1|\Delta}\right\} \\ &= \boldsymbol{\gamma}_{l-1} + \frac{1}{k_{l+1}}\sum_{t=k_l+1}^{k_{l+1}} \mathsf{Diag}\left\{\left(\boldsymbol{I} - \boldsymbol{D}^2\right)^{-1}\boldsymbol{T}_{t|\check{k}_l} - \boldsymbol{\Gamma}_{l-1}\right\}. \end{aligned} \tag{4.25}$$

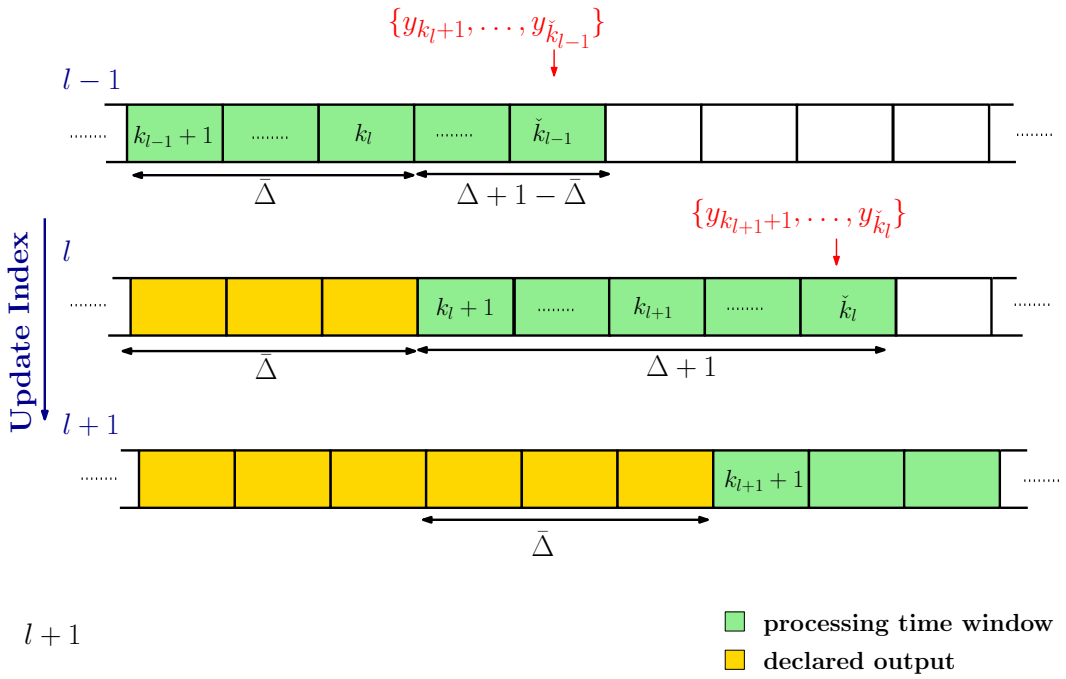


Figure 4.1: The sawtooth lag processing scheme

To compute $\boldsymbol{T}_{t|\check{k}_l}$, we run the fixed interval Kalman smoothing algorithm on overlapping blocks of data of size $\Delta+1$, and discard the last $\Delta+1-\bar{\Delta}$ values of every block (this is referred to as sawtooth lag smoothing [119]). The processing window is shifted by $\bar{\Delta}$ after every update. The update equations are comprised of forward recursions and backward recursions. In the forward recursions, we estimate $\widehat{\boldsymbol{x}}_{t|t}$ and $\boldsymbol{P}_{t|t}$ for $t = k_l+1, k_l+2, \ldots, \check{k}_l$ using a Kalman filter as given below:

$$\widehat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{D}\hat{\boldsymbol{x}}_{t-1|t-1} \tag{4.26}$$

$$\boldsymbol{P}_{t|t-1} = \boldsymbol{D}\boldsymbol{P}_{t-1|t-1}\boldsymbol{D} + (\boldsymbol{I}-\boldsymbol{D}^2)\boldsymbol{\Gamma} \tag{4.27}$$

$$\boldsymbol{J}_t = \boldsymbol{P}_{t|t-1}\boldsymbol{A}_t^{\mathsf{T}}\left(\boldsymbol{A}_t\boldsymbol{P}_{t|t-1}\boldsymbol{A}_t^{\mathsf{T}} + \boldsymbol{R}_t\right)^{-1} \tag{4.28}$$

$$\hat{\boldsymbol{x}}_{t|t} = (\boldsymbol{I}-\boldsymbol{J}_t\boldsymbol{A}_t)\hat{\boldsymbol{x}}_{t|t-1} + \boldsymbol{J}_t\boldsymbol{y}_t \tag{4.29}$$

$$\boldsymbol{P}_{t|t} = (\boldsymbol{I}-\boldsymbol{J}_t\boldsymbol{A}_t)\boldsymbol{P}_{t|t-1} \tag{4.30}$$

$$\boldsymbol{P}_{\check{k}_l,\check{k}_l-1|\check{k}_l} = \left(\boldsymbol{I} - \boldsymbol{J}_{\check{k}_l}\boldsymbol{A}_{\check{k}_l}\right)\boldsymbol{D}\boldsymbol{P}_{\check{k}_l-1|\check{k}_l-1}. \tag{4.31}$$

In the backward recursions, we estimate $\widehat{\boldsymbol{x}}_{t|\check{k}_l}$, $\boldsymbol{P}_{t|\check{k}_l}$ and $\boldsymbol{P}_{t,t-1|\check{k}_l}$ in the reverse order. For $t = \check{k}_l, \check{k}_l - 1, \ldots, k_l + 2$ we get the following smoothing equations:

$$\boldsymbol{G}_{t-1} = \boldsymbol{P}_{t-1|t-1}\boldsymbol{D}\boldsymbol{P}_{t|t-1}^{-1} \tag{4.32}$$

$$\hat{\boldsymbol{x}}_{t-1|\check{k}_l} = \hat{\boldsymbol{x}}_{t-1|t-1} + \boldsymbol{G}_{t-1}(\hat{\boldsymbol{x}}_{t|\check{k}_l} - \hat{\boldsymbol{x}}_{t|t-1}) \tag{4.33}$$

$$\boldsymbol{P}_{t-1|\check{k}_l} = \boldsymbol{P}_{t-1|t-1} + \boldsymbol{G}_{t-1}(\boldsymbol{P}_{t|\check{k}_l} - \boldsymbol{P}_{t|t-1})\boldsymbol{G}_{t-1}^{\mathsf{T}} \tag{4.34}$$

For $t \neq \check{k}_l$

$$\boldsymbol{P}_{t,t-1|\check{k}_l} = \boldsymbol{P}_{t|t}\boldsymbol{G}_{t-1}^{\mathsf{T}} + \boldsymbol{G}_t\left(\boldsymbol{P}_{t+1,t|\check{k}_l} - \boldsymbol{D}\boldsymbol{P}_{t|t}\right)\boldsymbol{G}_{t-1}^{\mathsf{T}}. \tag{4.35}$$

The scheme is illustrated in Section 4.4.1. In the figure, each box represents a time (sampling) instant with which it is indexed, and each row corresponds to an update index, with the index indicated in blue. The set of $\boldsymbol{y}$ in red represents the new measurement set processed in each update. A green box (with indices $k_l + 1 = (l-1)\bar{\Delta} + 1$ to $\check{k}_l = (l-1)\bar{\Delta} + \Delta + 1$) indicates that the state statistics corresponding to the index on box are updated, a yellow box (with indices $k \leq k_l = (l-1)\bar{\Delta}$) indicates that the state statistics are not updated, and a white box (with indices $k \geq \check{k}_l = (l-1)\bar{\Delta} + \Delta + 1$) indicates that the state statistics have not been computed yet. The processing window indicated by green is shifted by $\bar{\Delta}$ after every update. The average latency of the fixed lag scheme is Δ, whereas that of the sawtooth lag scheme is $\Delta - \left(\bar{\Delta} - 1\right)/2$. In the sawtooth lag scheme, $\bar{\Delta}$ also controls the frequency of update of $\boldsymbol{\gamma}$. If $\bar{\Delta}$ is large, the average latency decreases, but the $\boldsymbol{\gamma}$ gets updated more slowly. So, there is a tradeoff between the accuracy and the latency in selecting $\bar{\Delta}$.

Next, we discuss the special case of $\boldsymbol{D} = \boldsymbol{0}$. We refer to this algorithm as the *online M-SBL algorithm*, as there is no role for Kalman filtering when $\boldsymbol{D} = \boldsymbol{0}$.

Online M-SBL: When the sparse vectors are uncorrelated, i.e., $\boldsymbol{D} = \boldsymbol{0}$, (4.25) simplifies to the following recursion:

$$\boldsymbol{\gamma}_l = \boldsymbol{\gamma}_{l-1} + \frac{1}{k_{l+1}} \sum_{t=k_l+1}^{k_{l+1}} \text{Diag}\left\{ \boldsymbol{P}_t(\boldsymbol{\gamma}_{l-1}) + \widehat{\boldsymbol{x}}_t(\boldsymbol{y}_t, \boldsymbol{\gamma}_{l-1}) \widehat{\boldsymbol{x}}_t(\boldsymbol{y}_t, \boldsymbol{\gamma}_{l-1})^{\mathsf{T}} - \boldsymbol{\Gamma}_{l-1} \right\}, \tag{4.36}$$

where

$$\boldsymbol{P}_t(\boldsymbol{\gamma}) \triangleq \boldsymbol{\Gamma} - \boldsymbol{\Gamma} \boldsymbol{A}_t^{\mathsf{T}} \left(\boldsymbol{A}_t \boldsymbol{\Gamma} \boldsymbol{A}_t^{\mathsf{T}} + \boldsymbol{R}_t \right)^{-1} \boldsymbol{A}_t \boldsymbol{\Gamma} \tag{4.37}$$

$$\widehat{\boldsymbol{x}}_t(\boldsymbol{y}, \boldsymbol{\gamma}) \triangleq \boldsymbol{P}_t(\boldsymbol{\gamma}) \boldsymbol{A}_t^{\mathsf{T}} \boldsymbol{R}_t^{-1} \boldsymbol{y}. \tag{4.38}$$

We note that this implementation depends only on $\bar{\Delta}$, and not on Δ, because the set of measurement vectors $\left\{ \boldsymbol{y}_t, t = k_{l+1} + 1, k_{l+1} + 2, \ldots, \check{k}_l \right\}$ and the set of sparse vectors $\{ \boldsymbol{x}_t, t = k_l + 1, k_l + 2, \ldots, k_{l+1} \}$ are independent.

To summarize, we have presented a fixed lag scheme and a sawtooth lag scheme, for computing $\boldsymbol{T}_{k|k+\Delta}$ recursively using the data in batches. We next discuss the initialization of the algorithm and several interesting special cases.

4.4.2 Discussion

Initialization

The initial estimate of $\boldsymbol{\gamma}$ can be obtained from the first $\Delta + 1$ input measurements vectors using the offline KM-SBL algorithm. The one round of the offline KM-SBL algorithm can be interpreted as an estimation step, and the recursive update of $\boldsymbol{\gamma}$ using (4.16) can be interpreted as a tracking process. In fact, if $\boldsymbol{\gamma}$ is slowly varying over time, the recursive

update step (4.16) can track its temporal variations.

Special Cases

We make a few interesting observations about the algorithm in the following special cases:

(a) When $\boldsymbol{D} = \boldsymbol{0}$, the sparse vectors are uncorrelated and thus $\widehat{\boldsymbol{x}}_{t|K} = \widehat{\boldsymbol{x}}_{t|t+\Delta}$, $\boldsymbol{P}_{t|K} = \boldsymbol{P}_{t|t+\Delta}$, and $\boldsymbol{P}_{t,t-1|K} = \boldsymbol{P}_{t,t-1|t+\Delta}$. Hence, there is no approximation in (4.14). On the other hand, as the correlation coefficient increases, the approximation in (4.14) becomes loose.

(b) When $\boldsymbol{D} = \boldsymbol{0}$ and $\Delta = 0$, the fixed lag and the sawtooth lag schemes become identical.

(c) When $\Delta = 0$, the filter for the modified state space reduces to the original Kalman filter equations [2].

(d) When $\bar{\Delta} = 1$, the latency of the sawtooth lag scheme equals Δ for all sparse vectors, similar to the fixed lag scheme. Nonetheless, the two schemes are different, because of the forward and backward recursions in the sawtooth lag scheme.

4.4.3 Refinements

Different Learning Rates

Instead of $1/k$ in (4.16), any sequence of positive numbers b_k can be used in the recursive algorithm as long as the following conditions are satisfied:

$$0 \leq b_k \leq 1 \qquad \sum_{k=1}^{\infty} b_k = \infty \qquad \sum_{k=1}^{\infty} b_k^2 < \infty. \tag{4.39}$$

The modified algorithm is given by

$$\boldsymbol{\gamma}_k = \boldsymbol{\gamma}_{k-1} + b_k \mathsf{Diag}\left\{ \left(\boldsymbol{I} - \boldsymbol{D}^2\right)^{-1} \boldsymbol{T}_{k|k+\Delta} - \boldsymbol{\Gamma}_{k-1} \right\}. \tag{4.40}$$

A good choice for the sequence is $b_k = 1/k^\alpha$, $1/2 < \alpha \leq 1$, since $\sum_{k=1}^{\infty} 1/k^\alpha$ converges if $\alpha > 1$ and diverges otherwise. In Section 4.5, we empirically show that the modified algorithm converges faster than the original version (see Figure 5.1).

Improved Online M-SBL

Notice that the online M-SBL algorithm in (4.36) does not use the observations $\boldsymbol{y}_t$, $t = k_{l+1}+1, k_{l+1}+2, \ldots, \check{k}_l$, even though they are available at time k_{l+1}. Hence, we modify the update step in (4.36) to update $\boldsymbol{\gamma}$ using all the available measurement vectors $\boldsymbol{y}^{\check{k}_l}$, and then estimate the sparse vectors $\hat{\boldsymbol{x}}_{k_l+1}$ to $\hat{\boldsymbol{x}}_{k_{(l+1)}}$, as follows:

$$\boldsymbol{\gamma}_l = \boldsymbol{\gamma}_{l-1} + \frac{1}{\check{k}_l} \sum_{t=\check{k}_l-\bar{\Delta}+1}^{\check{k}_l} \mathrm{Diag}\Big\{ \boldsymbol{P}_t(\boldsymbol{\gamma}_{l-1}) + \widehat{\boldsymbol{x}}_t(\boldsymbol{y}_t, \boldsymbol{\gamma}_{l-1}) \widehat{\boldsymbol{x}}_t(\boldsymbol{y}_t, \boldsymbol{\gamma}_{l-1})^{\mathsf{T}} - \boldsymbol{\Gamma}_{l-1} \Big\}. \tag{4.41}$$

Thus, for each update, we use only the latest available block of size $\bar{\Delta}$, and not the past values which have already been used. Hence, in this case, we need not store any of the past measurements or the sparse vector estimates.

4.4.4 Complexity Analysis

We now briefly discuss the computational complexity and memory requirements of our algorithms. We note that Table 4.1 summarized the computational demands of every iteration of the iterative algorithms whereas here in Table 4.2 we summarize the total computational requirements of the non-iterative algorithm as every measurement block

Scheme		**Computational cost**	**Memory demand**
KM-SBL ($\boldsymbol{D} \neq \mathbf{0}$)	Offline	$\mathcal{O}(KN^3)$	$\mathcal{O}\left(KN^2\right)$
	Fixed lag	$\mathcal{O}\left(KN^2m\right)$	$\mathcal{O}\left(\Delta N^2\right)$
	Sawtooth lag	$\mathcal{O}(KN^3)$	$\mathcal{O}\left(\Delta N^2\right)$
M-SBL ($\boldsymbol{D} = \mathbf{0}$)	Offline	$\mathcal{O}(KN^2m)$	$\mathcal{O}\left(Km + N^2\right)$
	Online	$\mathcal{O}(KN^2m)$	$\mathcal{O}\left(\Delta m + N^2\right)$

Table 4.2: Comparison of the online schemes with the offline scheme when K observations are available.

arrive.

Computational Cost

We assume that the multiplication of a $p \times q$ matrix with a $q \times r$ matrix requires $\mathcal{O}(pqr)$ floating-point operations (flops), and the inversion of a $p \times p$ positive definite matrix requires $\mathcal{O}(p^3)$ flops [120].

We note that the computational cost per update of $\boldsymbol{\gamma}$ in the online scheme depends only on Δ (which is $\ll K$), although the overall computational complexity does depend on the number of sparse vectors K. However, simulation results show that the overall run time of our online algorithms grow slowly with K when compared to their offline counterparts (see Figure 4.2a). The order-wise complexity of the online M-SBL algorithm (4.36) is similar to the online KM-SBL fixed-lag scheme, but its run time is much smaller than KM-SBL since it does not involve Kalman filtering or smoothing. Note that, the computational cost of the offline algorithms correspond to the complexity of a single iteration, while that of the online algorithms correspond to the overall complexity, as they are non-iterative in nature.

4.5 Numerical Results: Non-iterative Algorithm

We use the following setup to evaluate the performance of the algorithm and corroborate the theoretical results. We generate sparse signals of length $N = 60$, each with $s = 6$ nonzero entries. The locations of nonzero coefficients are chosen uniformly at random, and the nonzero entries are independent and identically distributed with zero mean and unit variance. The length of measurement vector is chosen as $m = 20$. The measurement matrices $\boldsymbol{A}_k$ are generated with independent and Gaussian distributed entries with zero mean, and the columns are normalized to have unit Euclidean norm.

We study the properties of the algorithm for both uncorrelated and highly correlated cases in the following subsections. For the uncorrelated case, we consider the improved online algorithm given by (4.41). We evaluate the performance of our algorithm using the same three metrics used in the last section. We consider two methods to initialize the hyperparameter vector $\boldsymbol{\gamma}$ for the online schemes, which we term *proper* initialization and *fixed* initialization. Proper initialization refers to initializing $\boldsymbol{\gamma}$ with its estimate obtained from the first $\Delta+1$ measurements using the offline KM-SBL algorithm. Fixed initialization refers to initializing $\boldsymbol{\gamma}$ with a fixed vector (which we take as $4 \cdot \mathbf{1}$).

Uncorrelated Case

Figures 4.2a-4.3c show the performance of the different schemes when $\boldsymbol{D} = \mathbf{0}$. The curves labeled `Offline` correspond to the performance of the offline M-SBL algorithm, which is our benchmark, and all other curves correspond to the improved online sawtooth lag scheme discussed in Section 4.4.3. The curves labeled `Init` $\bar{\Delta} = 1$, `Init` $\bar{\Delta} = 3$ and `Init` $\bar{\Delta} = 5$ correspond to the online algorithm with proper initialization, while the curves

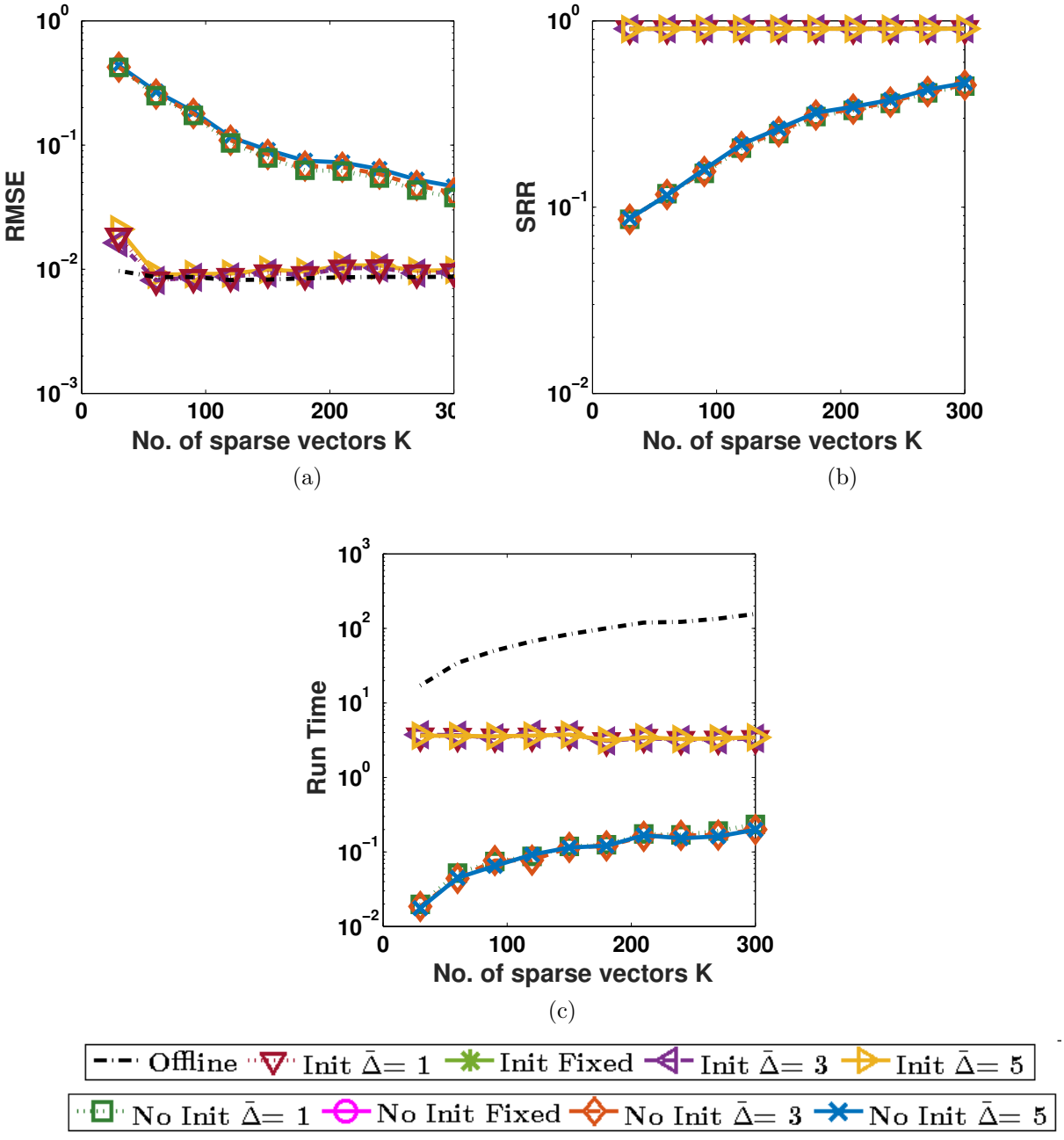


Figure 4.2: Performance of our algorithms relative to the offline algorithm for $\boldsymbol{D} = \mathbf{0}$ (uncorrelated case, where we use the M-SBL based algorithm). Other paramters are $\Delta = 5$ and SNR $= 20$ dB.

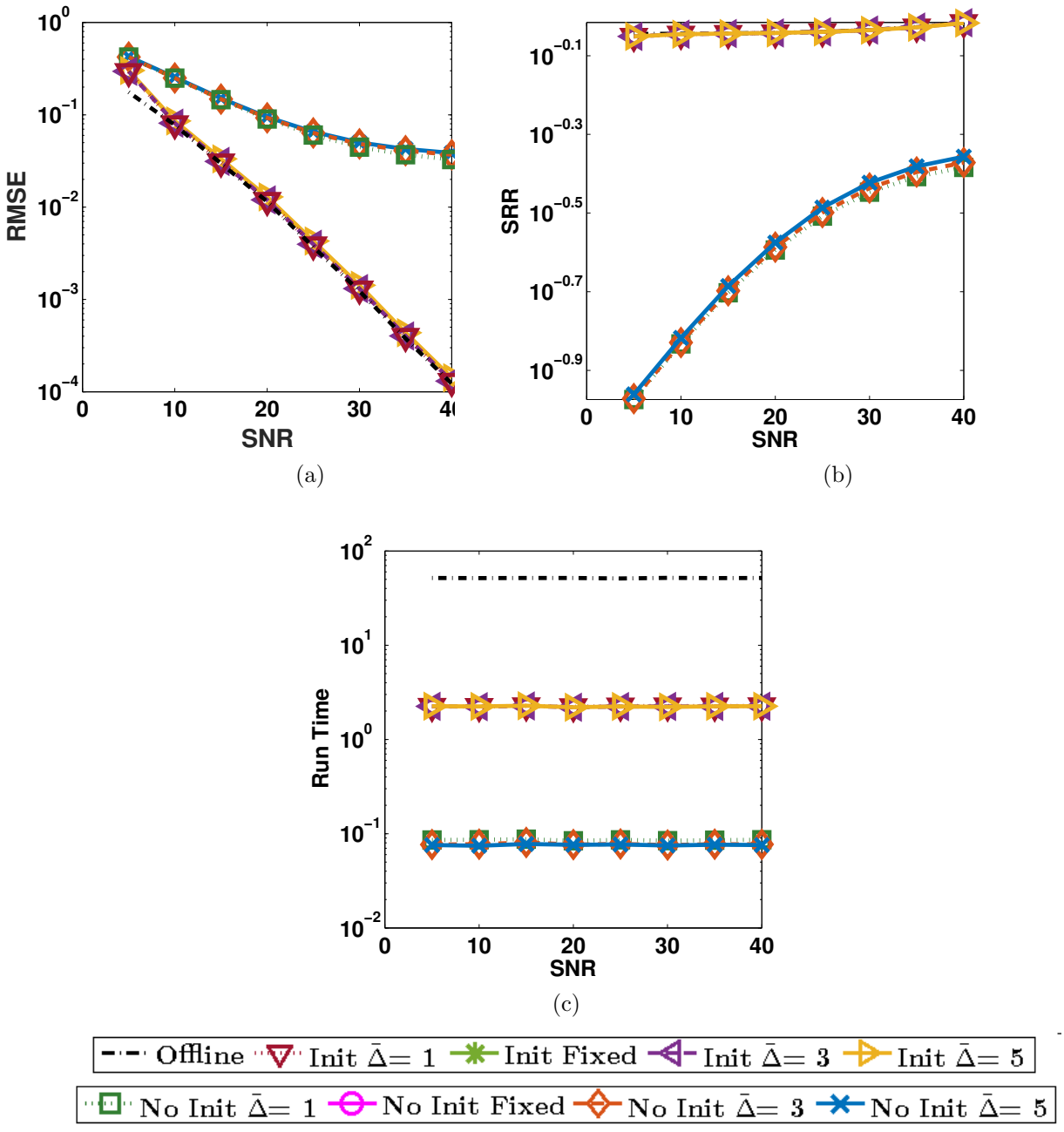


Figure 4.3: Performance of our algorithms relative to the offline algorithm for $\boldsymbol{D} = \mathbf{0}$ (uncorrelated case, where we use the M-SBL based algorithm). Other paramters are $\Delta = 5$ and $K = 120$.

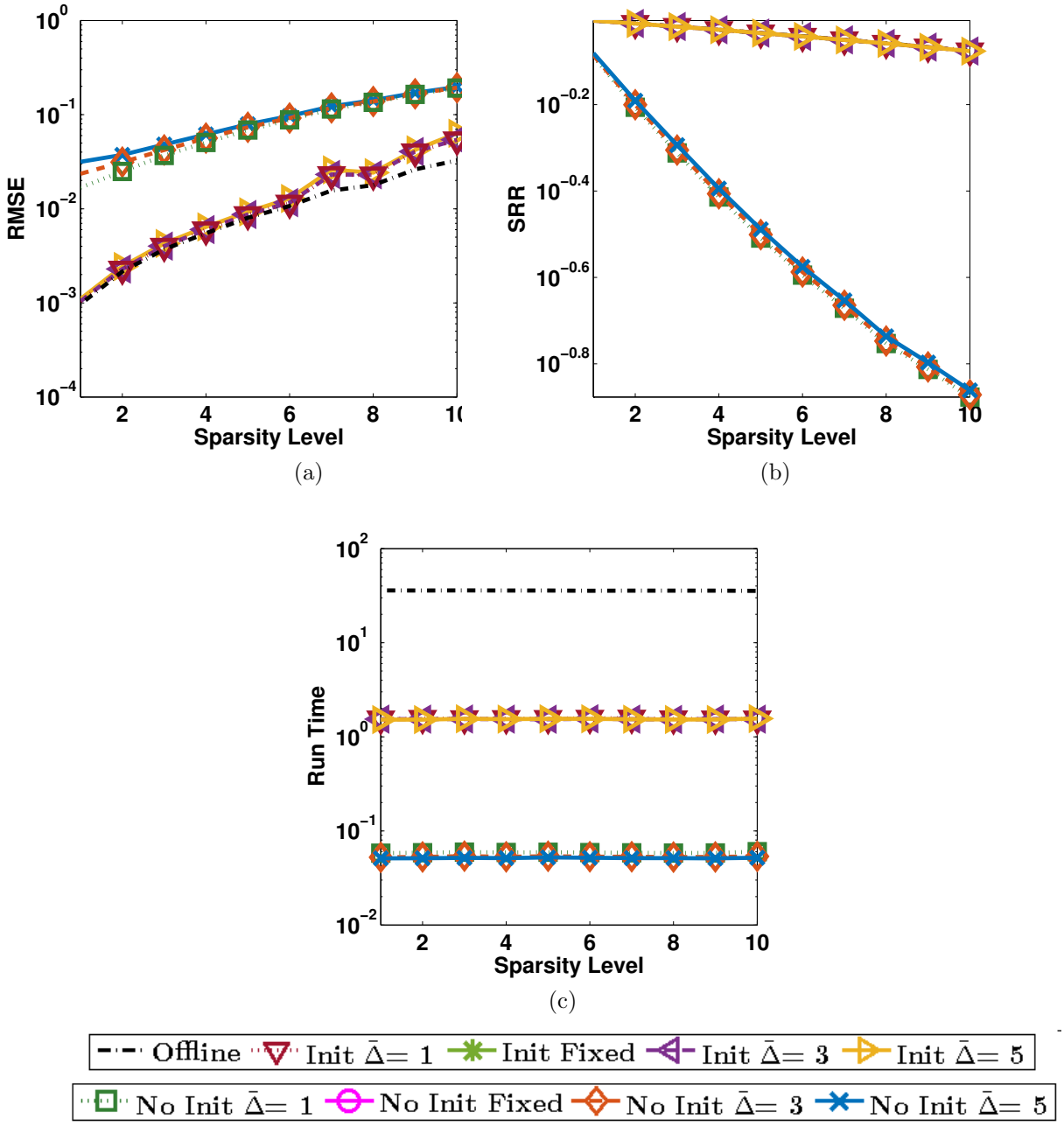


Figure 4.4: Performance of our algorithms relative to the offline algorithm for $\boldsymbol{D} = \mathbf{0}$ (uncorrelated case, where we use the M-SBL based algorithm). Other paramters are $K = 150$, $\Delta = 5$ and SNR $= 20$ dB.

labeled `No Init` $\bar{\Delta} = 1$, `No Init` $\bar{\Delta} = 3$ and `No Init` $\bar{\Delta} = 5$ correspond to the online algorithm with fixed initialization. Our observations from the results are as follows:

- *Initialization:* The online scheme with proper initialization closely matches with the offline scheme in terms of the recovery performance. On the other hand, the online scheme with fixed initialization requires significantly smaller time for execution, but the convergence is slower.

- *Number of sparse vectors K:* As K increases, the quality of the covariance estimate improves, and this, in turn, leads to better recovery performance; see Figures 4.2a and 4.2b. From Figure 4.2c, we see that the run time increases almost linearly with K for the offline scheme and the online scheme with fixed initialization. With proper initialization, the run time is roughly constant with K, as most of execution time is spent in computing the initialization of $\boldsymbol{\gamma}$.

- *SNR:* The recovery performance of all algorithms improve with increase in SNR, see Figures 4.3a and 4.3b. Also, the gap between the online scheme with proper initialization and the offline scheme virtually closes beyond an SNR of 10 dB. From Figure 4.3c, the run time remains almost constant with SNR, even though the offline scheme and the online scheme with proper initialization use an iterative step to estimate $\boldsymbol{\gamma}$.

- *Sparsity level:* The recovery performance of all algorithms degrade with increase in sparsity level (number of non-zero entries), see Figures 4.4a and 4.4b. However, the SRR performance of the algorithm with fixed initialization degrades significantly with the increase in the sparsity level. From Figure 4.4c, the run time remains almost

constant with sparsity level, since the complexity does not depend on the sparsity level.

- *Output batch-size* $\bar{\Delta}$*:* The performance of online schemes do not vary much with $\bar{\Delta}$, as can be seen from Figures 4.2a-4.3c. However, the recovery performance is slightly better and the run time is slightly worse for smaller values of $\bar{\Delta}$, as $\boldsymbol{\gamma}$ is updated more frequently.

- *Maximum delay* Δ*:* The performance of the algorithm with varying maximum delay Δ is similar to that of the highly correlated case as shown in Figure 4.5a-Figure 4.5c, and hence omitted. The performance of the online schemes improve as Δ increases, and the proper initialization can greatly improve the recovery performance compared to fixed initialization. The run time of the online scheme with proper initialization increases with Δ, because the number of measurement vectors used to initialize $\boldsymbol{\gamma}$ increases. However, the behavior the run time of the online schemes for the uncorrelated case is different from that of the highly correlated case, as discussed in Section 4.4.4. This is because the online algorithms use Kalman smoothing in the correlated case, and the complexity of Kalman smoothing increases with Δ. In the uncorrelated case, the complexity is independent of Δ, thus the run time remains constant for all values of Δ.

Highly Correlated Case

Figures 4.5a-4.6a show the performance of the different algorithms when the sparse vectors are highly correlated ($\boldsymbol{D} \neq \mathbf{0}$). The curves labeled `Init Fixed` and `No Init Fixed` correspond to the fixed lag scheme with proper and fixed initialization, respectively, while

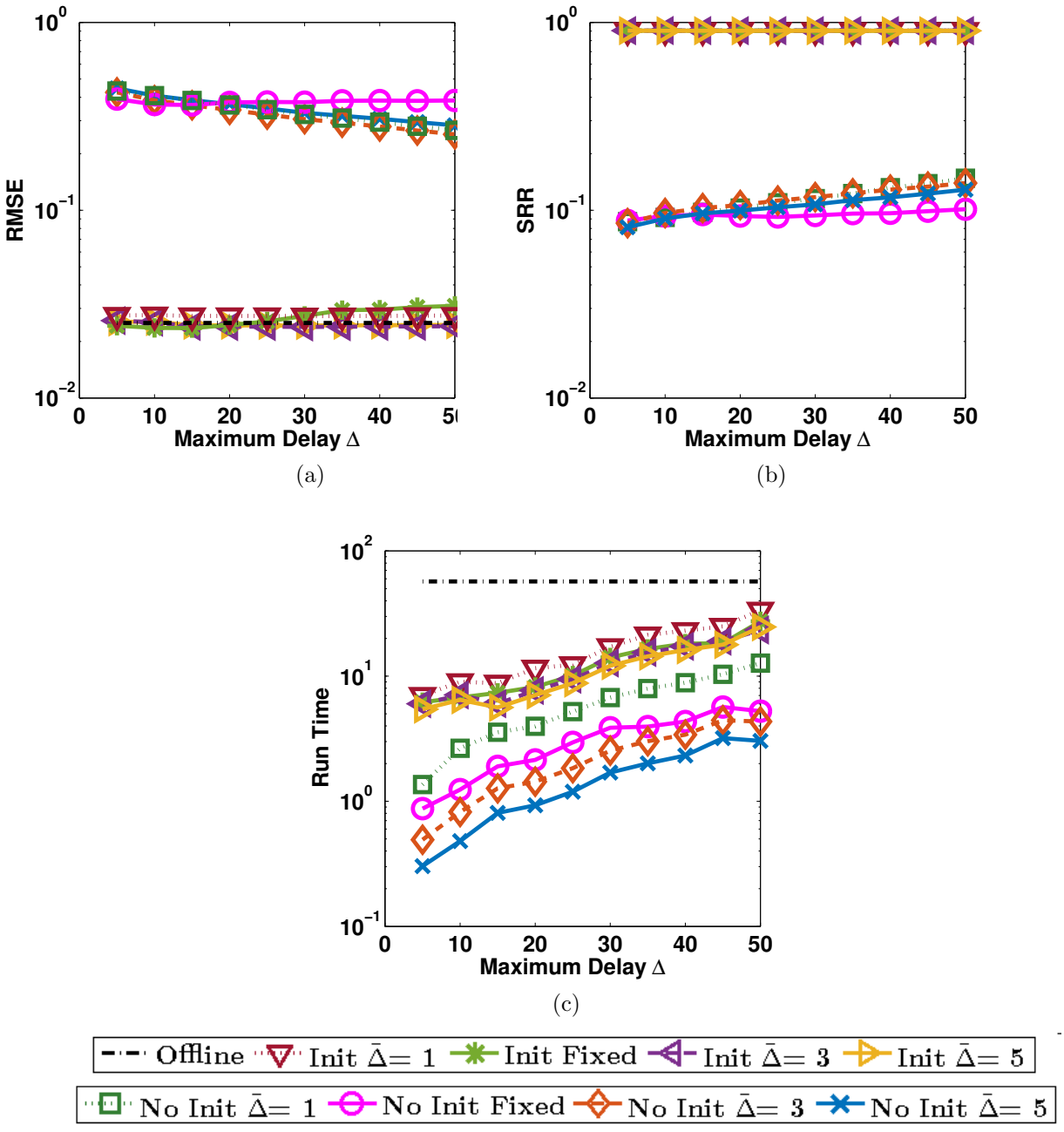


Figure 4.5: Performance of our algorithms relative to the offline algorithm for $\boldsymbol{D} = 0.9\boldsymbol{I}$ (correlated case, where we use the KM-SBL algorithm). Other paramters are $\Delta = 5$ and SNR $= 20$ dB.

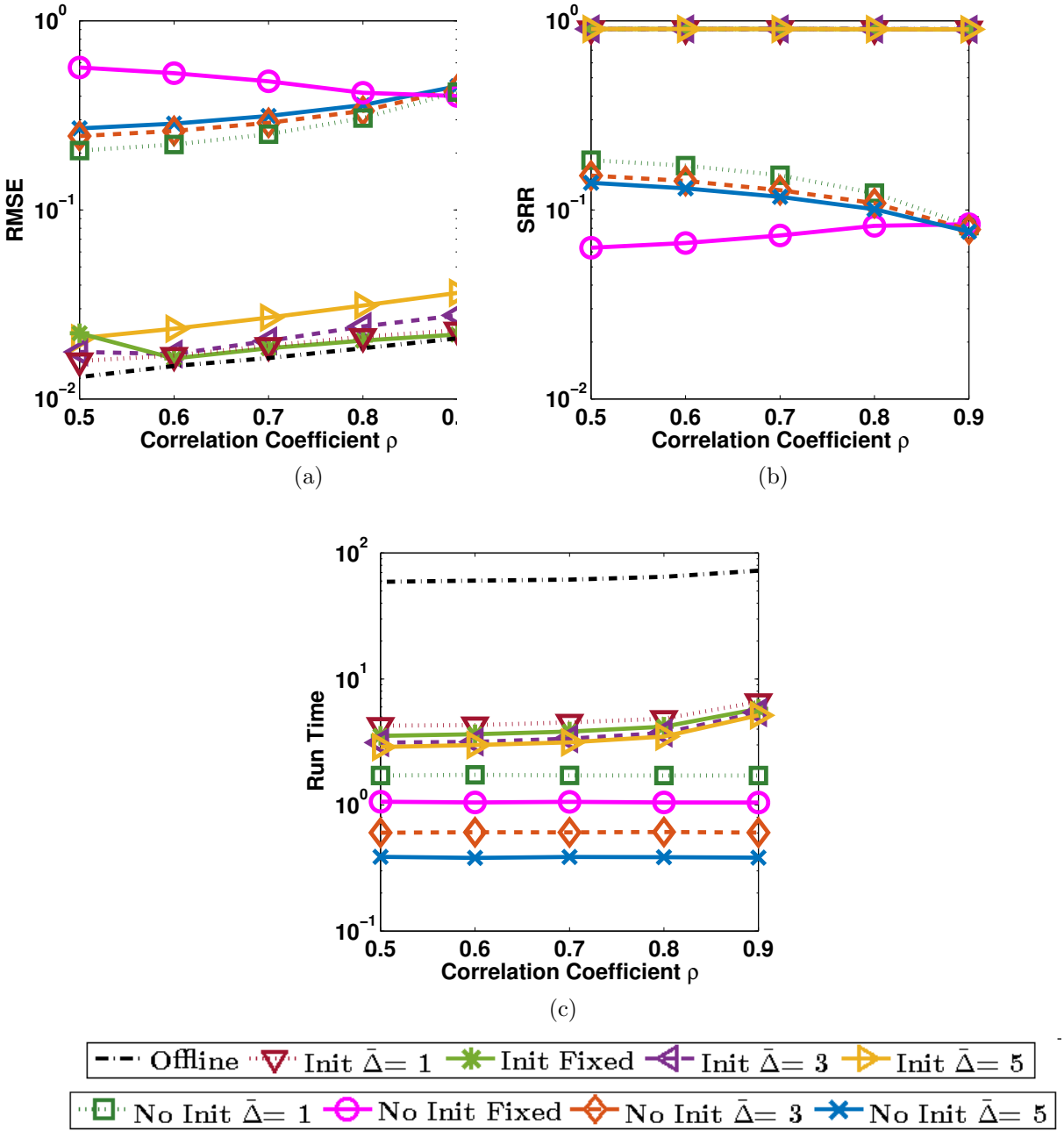


Figure 4.6: Performance of our algorithms relative to the offline algorithm for $\boldsymbol{D} = \rho \boldsymbol{I}$ (correlated case, where we use the KM-SBL algorithm). Other paramters are $K = 150$, $\Delta = 5$ and SNR $= 20$ dB.

the other labels are as in the previous plots. Our observations from the results are as follows:

- *Implementation schemes:* As discussed in Section 4.4.1, for the same output batch-size of $\bar{\Delta} = 1$, the sawtooth lag scheme outperforms the fixed lag scheme, at the cost of a higher run time. This is because the sawtooth lag scheme uses all the available measurements for updating the hyperparameters, while the fixed lag scheme uses only the latest available measurement. Comparing the fixed lag scheme with the sawtooth lag scheme with higher output batch-sizes ($\bar{\Delta} = 3$ and 5), the fixed lag scheme is slower but more accurate, as it updates the hyperparameters more frequently.

- *Correlation coefficient ρ:* The performance of the algorithms with varying correlation coefficient ρ (recall $\boldsymbol{D} = \rho \boldsymbol{I}$) is shown in Figures 4.6a-4.6c. As ρ increases, the recovery performance of the sawtooth lag scheme decreases, while that of the fixed lag scheme improves. This seemingly counterintuitive behavior can be explained as follows. In the offline case, an increase in ρ can worsen the support recovery of the sparse vectors, but helps the estimation of the amplitude of the nonzero entries. A combination of these effects determine the overall performance of the algorithm, and we see that the recovery performance slightly degrades as the ρ increases. A similar trend was observed in the SRR for the temporal M-SBL algorithm for recovering correlated sparse vectors [121, Figure 2]. In case of the sawtooth lag scheme, in addition to the above, an increase in ρ also makes the approximation in (4.14) loose. Due to this, the degradation in the recovery performance of the sawtooth lag scheme is large compared to the offline algorithm. In case of the fixed lag scheme, apart from the effects discussed above, an increase in ρ also improves ρ^{Δ}, the correlation

between the state and the observation in the new state space model (described by (4.1) and (4.17)). This improves the quality of the estimate output by the Kalman filter, and in turn helps the recovery. The overall effect of these is an improvement in the recovery performance of the fixed lag scheme. A more rigorous study of the effect of ρ an interesting topic for future work.

The run time of the algorithm remains the same for all values of ρ for the fixed initialization case, as its complexity is independent of ρ. However, the run time of the online schemes with proper initialization is higher in the highly correlated case. This is because, when data is highly correlated, the initialization phase using the offline scheme takes more iterations to converge. We can see a similar slight increase in the run time of the offline scheme in the highly correlated case.

- *Maximum delay* Δ*:* As the delay increases, the recovery performance of the online schemes increases for both methods of initialization. The change is more evident for the fixed initialization case, as the recovery performance of with proper initialization is very close to that of the offline scheme. We also observe that the improvement in recovery performance is small for the fixed lag scheme compared to the sawtooth lag scheme. This is because of the reduced correlation $\left(\boldsymbol{D}^{\Delta}\right)$ between the state and the observation of the new state space model given by (4.1) and (4.17). Also as pointed out earlier, the run time of the online schemes increases with Δ.

- *Output batch-size* $\bar{\Delta}$*:* The performance of the online algorithms remains constant with $\bar{\Delta}$ for both the correlated and uncorrelated case. However, the gap between the run time curves is wider for the correlated case. This is because each update of $\boldsymbol{\gamma}$ is computationally more expensive due to the Kalman smoothing in the correlated

case.

The performance of the online algorithms with K and SNR in the highly correlated case is similar to that observed in the uncorrelated case, and hence omitted.

In the next subsection, we compare the performance of our scheme with other existing online algorithms found in the literature as mentioned in Section 4.1.

4.5.1 Comparison with Existing Algorithms

In Figure 4.7a-Figure 4.7c, we compare our algorithm, labeled `Non-iterative KMSBL`, with the following algorithms (labels in brackets):

(i) Offline KM-SBL [3] (`Offline KMSBL`)

(ii) Reweighted l_1 dynamic filtering [111] (`RL1-DF`)

(iii) Iterative online KM-SBL (`Iterative KMSBL`)

(iv) Standard l_1 norm based algorithm on each measurement vector [122] (`Regular` l_1 `Norm`)

(v) Kalman compressed sensing [112] (`KF-CS`)

(vi) Least squares compressed sensing [109] (`LS-CS`)

Here, we choose $\Delta = 0$, as the other online schemes except the iterative online KM-SBL algorithm are not designed for $\Delta > 0$. We also note that we extended the Kalman compressed sensing algorithm in [112] to handle a first-order AR process with correlation matrix $\boldsymbol{D} \in [0,1]^N$, while the original algorithm only considers $\boldsymbol{D} = \boldsymbol{I}$. The recovery performance of our scheme is comparable with the other online schemes algorithms, and

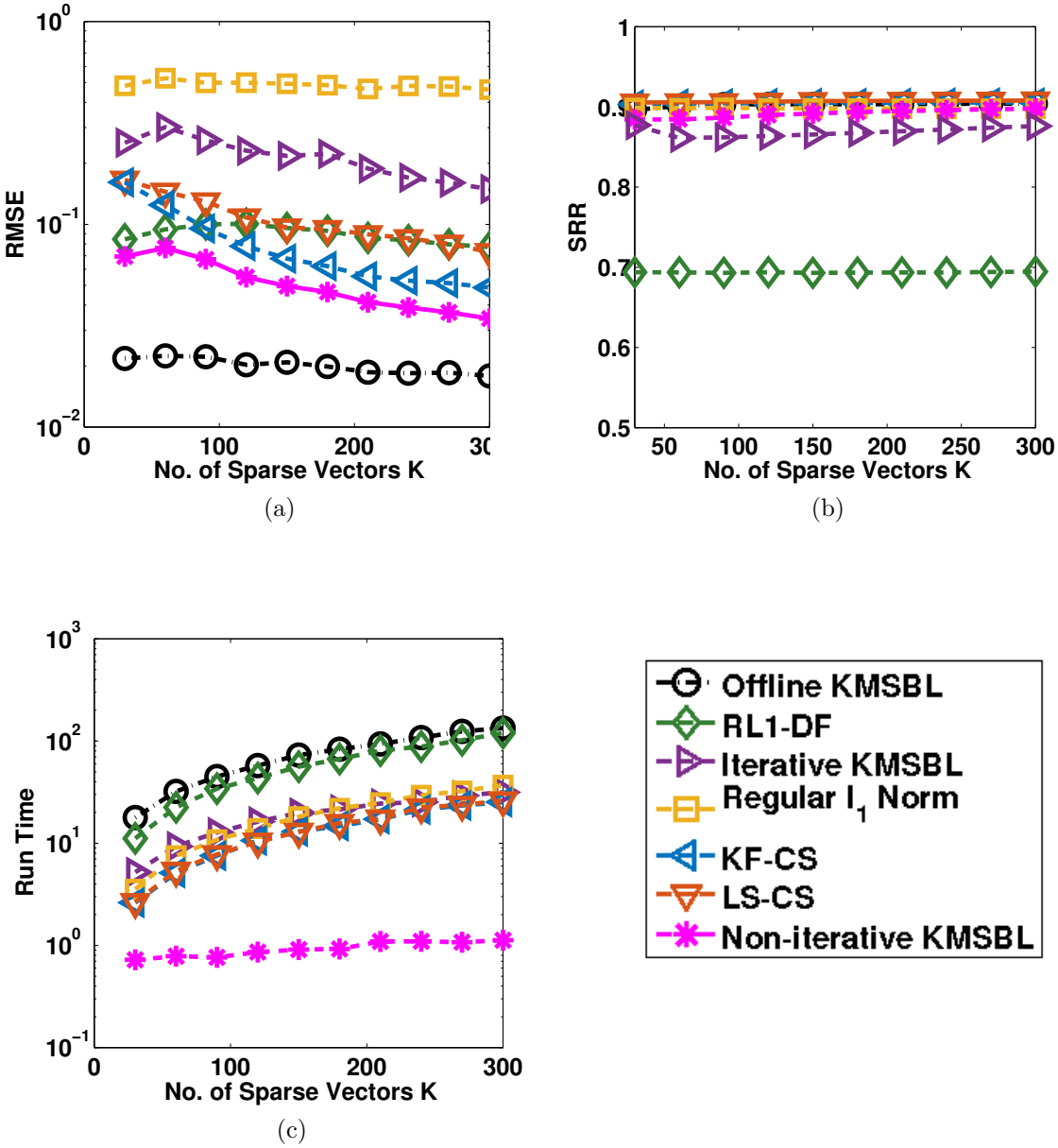


Figure 4.7: Comparison of RMSE, SRR and run time of our algorithm with the existing online schemes.

approaches the offline performance as K increases. However, the run time of our scheme is significantly lower than all the other schemes. Moreover, the rate of increase of the run time of our scheme with K is much smaller than the other schemes. The significant reduction in the run time is primarily due to the non-iterative nature of our scheme. Since all other algorithms are iterative in nature, their complexity and hence run time depends linearly on the number of iterations which, in turn, depends on N, m, K, the threshold used for stopping the iterations, etc. This brings out the major difference between the other algorithms and our online non-iterative schemes. Thus, our scheme is both fast and accurate, as promised in Section 4.1.

4.5.2 Sparse OFDM Channel Estimation

Parameter		**Value**
OFDM (3GPP/LTE broadband standard [123])	Transmission bandwidth	2.5 MHz
	Sub-frame duration	0.5 ms
	Subcarrier spacing	15 kHz
	Sampling frequency	3.84 MHz
	FFT size	256
	No. of data subcarriers	200
	OFDM symbol/slot	6
	CP length	16.67 μs
Channel	Environment	Pedestrian B [124]
	Model	Jakes model [125]
	Norm. Doppler freq.	10^{-3}
Coding and modulation		rate 1/2 Turbo code and QPSK
Pulse shaping		Raised cosine with rolloff factor= 0.5 [126]

Table 4.3: Simulation parameters for OFDM channel estimation

In this subsection, we consider the sparse OFDM channel estimation problem as an application of our algorithm [3]. We list the simulation parameters in Table 4.3. The sparse channel is of length $N = 59$, which taken as the length of the cyclic prefixing (CP), with $s = 6$ nonzero entries for each channel instantiation (PedB channel model [124]). In each OFDM symbol, $m = 20$ pilot symbols are placed uniformly, and the number of OFDM symbols K is taken as 150. We assume that the algorithms estimate the channel once in every OFDM slot, which gives $\Delta = 6$. We consider both coded[3] and uncoded scenarios and three metrics for the performance comparison: BER, MSE in channel estimation, and run time per channel vector estimation. We estimate the channel using the pilot symbols, and decode the data using the channel estimate (for details, refer to [3]). In Figure 4.8a-Figure 4.8c, we compare the performance of our algorithm, labeled `Online Non-iterative`, with the following three schemes (labels in brackets):

(i) Offline KM-SBL [3] (`Offline`)

(ii) Iterative online KM-SBL (`Online Iterative`)

(iii) Receiver with perfect knowledge of channel (`Genie`)

As mentioned earlier, the other online schemes are not applicable here, as we take $\Delta > 0$. From the figure, we infer that the BER and the MSE performance of our algorithm is better than the offline algorithm which was originally proposed for the channel estimation problem [3]. This is because the offline algorithm processes the data in blocks of size 6, and does not reuse the past measurements blocks, whereas our algorithm uses information

[3]For the Turbo code generation, we use the publicly available software [127].

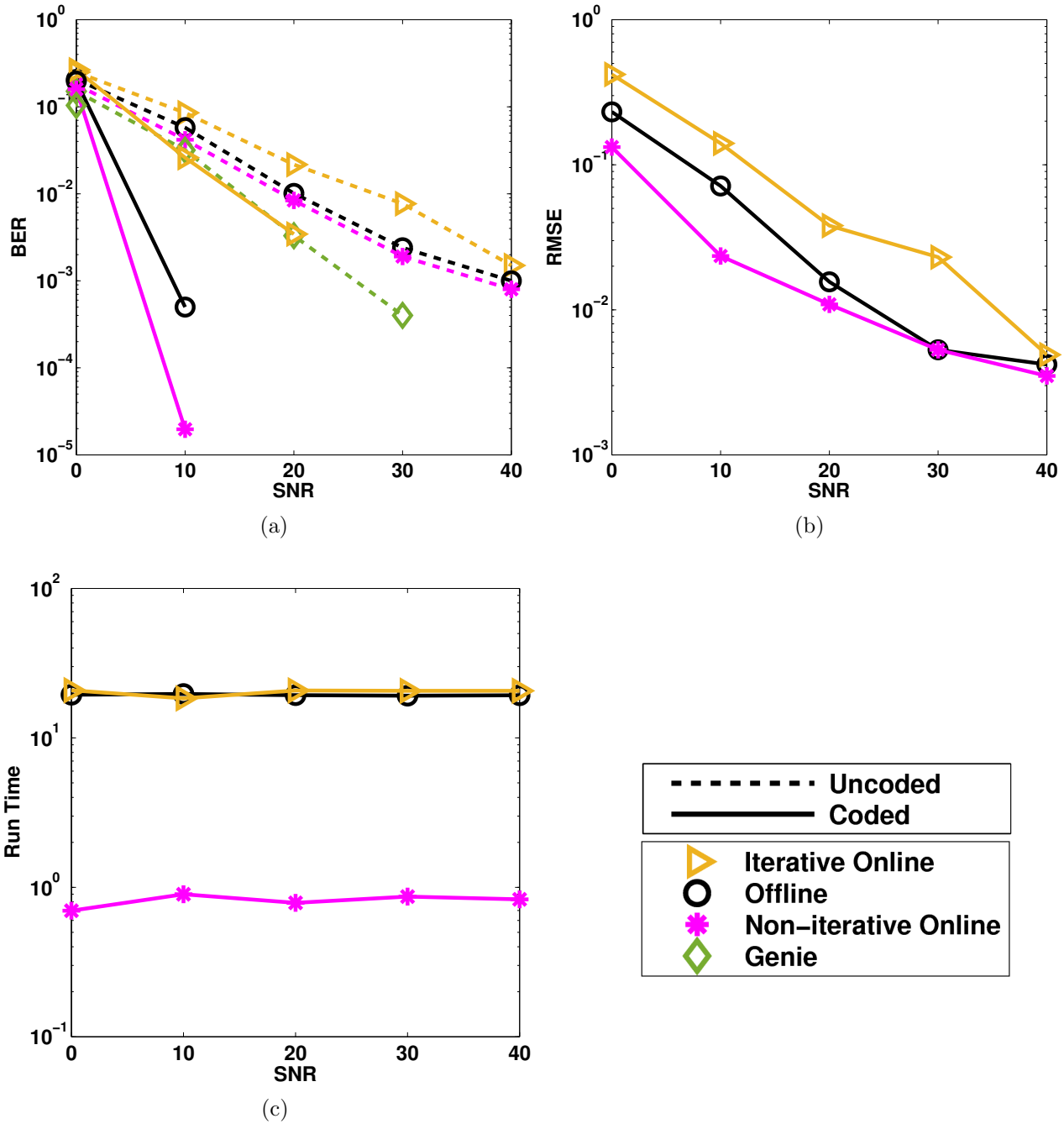


Figure 4.8: Comparison of the BER, RMSE and run time of our algorithm with existing schemes

from all past measurement blocks to estimate the channel vectors for the current block. Moreover, our algorithm has an added advantage of significantly reduced run time.

4.6 Summary

The chapter presented algorithms that are particularly useful in scenarios where noisy undetermined linear measurements of sparse state vectors arrive sequentially, and when one wants to exploit structure in the signal beyond (simultaneous) sparsity, specifically, the correlation introduced by the LDS. We developed two algorithms, namely, iterative and non-iterative, by combining the sequential EM procedure and the SBL framework, and presented two schemes for implementation: the fixed lag and sawtooth lag schemes. Our algorithms do not require any parameter tuning. Simulations showed that the performance of our algorithm is close to that of the offline algorithm, but it demands less memory and computational resources, both when the sparse vectors are uncorrelated and highly correlated. In short, in this chapter, we answered question Q2 for model SM2 by developing an online algorithm with good recovery properties. In the next chapter, we address question Q3 for the same model by providing some theoretical guarantees for the presented algorithms.

Chapter 5

Convergence Analysis of Online M-SBL Algorithm

Answering problem Q3 for SM2

In this chapter, we continue from the previous chapter and discuss the uniqueness of the solution obtained using the non-iterative online KM-SBL algorithm. The estimator for sparse unknowns is a MAP estimator, which enjoys the properties of being linear, unbiased and minimum variance. If the hyperparameter iterates converge to the true value, then the accuracy of estimating sparse unknowns is asymptotically equal to that of an oracle estimator that knows the variances of the entries of the unknowns. Thus, the convergence analysis has implications on the accuracy of the estimates, and provides insights to the uniqueness of the solution. Hence, we establish the uniqueness by examining the convergence properties of the presented algorithms and computing the limit point of the sequence of common hyperparameter iterates generated by the algorithm. However, the analysis for the arbitrary correlation case is hard, because the evolution of the every value in the sequence is a complicated function of the previous value. Therefore, we consider the

convergence results for the two extreme values of correlation: the uncorrelated case, and the perfectly correlated case. Intuitively, the algorithm should converge for all intermediate values of the correlation also. This is corroborated by our empirical results.

5.1 Uncorrelated Case

In the section, we study the convergence properties of the non-iterative online algorithm under the following assumptions:

(A1) The measurement matrices are identical, i.e., $\boldsymbol{A}_k = \boldsymbol{A}$, $\forall k$, and without loss of generality, $\mathsf{Rank}\,\{\boldsymbol{A}\} = m$.

(A2) The noise covariance matrix is the same for all measurements, i.e., $\boldsymbol{R}_k = \boldsymbol{R}$, $\forall k$.

(A3) The sparse vectors are uncorrelated, i.e., $\boldsymbol{D} = \boldsymbol{0}$.

The above assumptions are standard in the MMV literature, and are referred to as the joint sparsity model-2 (JSM-2) [37, 39–41]. The assumptions simplify the recursive algorithm, and make the analysis tractable. Since $\boldsymbol{D} = \boldsymbol{0}$, the fixed lag scheme discussed in Section 4.4.1 is not applicable, and we focus our analysis on the sawtooth lag implementation. We start with the case when $\bar{\Delta} = 1$. A similar analysis follows for $\bar{\Delta} > 1$, and we discuss this case later in the sequel.

When $\boldsymbol{A}_k = \boldsymbol{A}$ and $\boldsymbol{R}_k = \boldsymbol{R}$, (4.36)-(4.38) simplify to the following recursion:

$$\boldsymbol{\gamma}_k = \boldsymbol{\gamma}_{k-1} + \frac{1}{k}\mathsf{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma}_{k-1})\right\} + \frac{1}{k}\mathsf{Diag}\left\{\widehat{\boldsymbol{x}}(\boldsymbol{y}_k, \boldsymbol{\gamma}_{k-1})\widehat{\boldsymbol{x}}(\boldsymbol{y}_k, \boldsymbol{\gamma}_{k-1})^{\mathsf{T}} - \boldsymbol{\Gamma}_{k-1}\right\}, \tag{5.1}$$

where $\boldsymbol{P}(\boldsymbol{\gamma})$ and $\widehat{\boldsymbol{x}}(\boldsymbol{y}, \boldsymbol{\gamma})$ are as defined in (4.37) and (4.38), with $\boldsymbol{A}_t$ and $\boldsymbol{R}_t$ replaced by

$\boldsymbol{A}$ and $\boldsymbol{R}$, respectively. We can rewrite (5.1) as a stochastic approximation recursion as follows:

$$\boldsymbol{\gamma}_k = \boldsymbol{\gamma}_{k-1} + \frac{1}{k}\boldsymbol{f}(\boldsymbol{\gamma}_{k-1}) + \frac{1}{k}\boldsymbol{e}_k. \tag{5.2}$$

Here, $\boldsymbol{f}(\boldsymbol{\gamma})$ is the mean field function, given by

$$\boldsymbol{f}(\boldsymbol{\gamma}) \triangleq \mathsf{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma}) + \boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathsf{T}}\boldsymbol{R}^{-1}\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\right\}\boldsymbol{R}^{-1}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\} - \boldsymbol{\gamma}, \tag{5.3}$$

where the expectation is over the distribution of $\boldsymbol{y}$, and $\boldsymbol{e}_k$ is given by

$$\boldsymbol{e}_k \triangleq \mathsf{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma}_{k-1}) + \widehat{\boldsymbol{x}}(\boldsymbol{y}_k, \boldsymbol{\gamma}_{k-1})\widehat{\boldsymbol{x}}(\boldsymbol{y}_k, \boldsymbol{\gamma}_{k-1})^{\mathsf{T}}\right\} - \boldsymbol{\gamma}_{k-1} - \boldsymbol{f}(\boldsymbol{\gamma}_{k-1}). \tag{5.4}$$

Further, using $\boldsymbol{P}(\boldsymbol{\gamma})$ from (4.37),

$$\boldsymbol{P}(\boldsymbol{\gamma}) - \boldsymbol{\Gamma} = -\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma} \tag{5.5}$$

$$\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathsf{T}}\boldsymbol{R}^{-1} = \boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}. \tag{5.6}$$

Thus, we get the following:

$$\boldsymbol{f}(\boldsymbol{\gamma}) = \mathsf{Diag}\left\{\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\left(\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\right\} - \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} - \boldsymbol{R}\right)\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}\right\} \tag{5.7}$$

$$\boldsymbol{e}_k = \mathsf{Diag}\left\{\boldsymbol{\Gamma}_{k-1}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}_{k-1}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\left(\boldsymbol{y}_k\boldsymbol{y}_k^{\mathsf{T}} - \mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\right\}\right)\left(\boldsymbol{A}\boldsymbol{\Gamma}_{k-1}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}_{k-1}\right\}. \tag{5.8}$$

We next present the convergence results of the algorithm. We begin with a proposition which shows that the sequence of $\boldsymbol{\gamma}_k$ generated by the algorithm is bounded.

Proposition 5.1. *If $\boldsymbol{\gamma}_0$ is a nonnegative vector, the sequence $\boldsymbol{\gamma}_k$ generated by* (5.1) *remains in a compact subset of $\mathbb{R}_+^N$ almost surely (*a.s.*).*

Proof. See Appendix C.1. □

The next question to be answered is about the values to which the sequence $\boldsymbol{\gamma}_k$ could converge. The following theorem characterizes the asymptotic behavior of the algorithm.

Theorem 5.1. *Assume that the nonzero entries of $\boldsymbol{x}$ are orthogonal, and the diagonal matrix $\boldsymbol{\Gamma}_{\mathsf{opt}} \triangleq \mathbb{E}\left\{\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}\right\}$. If $\boldsymbol{\gamma}_0$ is a nonnegative vector, then the sequence $\boldsymbol{\gamma}_k$ of the online M-SBL algorithm given by* (5.1) *converges to an element in the union set: $\{\mathbf{0}\} \cup \{\boldsymbol{\gamma} \in \mathbb{R}_+^N : \boldsymbol{A}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathsf{opt}}\right)\boldsymbol{A}^{\mathsf{T}} = \mathbf{0}\}$* a.s. *Further, if $\mathsf{Rank}\left\{\boldsymbol{A} \odot \boldsymbol{A}\right\} = N$, the sequence $\boldsymbol{\gamma}_k$ converges to a point in the two-element set $\{\mathbf{0}, \boldsymbol{\gamma}_{\mathsf{opt}}\}$* a.s.

Proof. See Appendix C.2. □

We make the following observations from Theorem 5.1.

- The results are independent of the following parameters:
  - (a) sparsity level of the unknown vectors
  - (b) initialization of the algorithm (however, $\boldsymbol{\gamma}_0 \in \mathbb{R}_+^N$)
  - (c) distribution of the sparse vectors (even though the algorithm is designed assuming a Gaussian distribution), as long as the entries are orthogonal
  - (d) properties of $\boldsymbol{A}$, such as its restricted isometry constant or mutual coherence
  - (e) construction of $\boldsymbol{A}$, i.e., it can be deterministic or random, with normalized or unnormalized columns.

- The convergence guarantee of the original M-SBL algorithm in [41] holds only in the noiseless case. However, our generalized result applies whether noise is present or not. Hence, the result is practically more useful.

- The condition that the nonzero entries of $\boldsymbol{x}$ should be orthogonal is similar to the orthogonality condition required for the convergence guarantee of the original M-SBL algorithm in the noiseless case [41]. In fact, the orthogonality condition in [41] is hard to achieve since the number of sparse vectors to be estimated is finite. In that sense, ours is a more reasonable assumption.

- The M-SBL cost function [41] is defined as

$$\begin{aligned} V_{\text{M-SBL}}(\boldsymbol{\gamma}) &= \lim_{k\to\infty}\left[\frac{1}{k}\sum_{t=1}^{k}\boldsymbol{y}_t^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{y}_t+\log\left|\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right|\right] \\ &= \mathsf{Tr}\left\{\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)^{-1}\left(\boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)\right\}-\log\left|\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)^{-1}\right|. \end{aligned} \tag{5.9}$$

  We note that $V_{\text{M-SBL}}(\boldsymbol{\gamma})-\log\left|\boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right|-m$ is the Kullback-Leibler (KL) divergence between two distributions: $\mathcal{N}(\boldsymbol{0},\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R})$ and $\mathcal{N}(\boldsymbol{0},\boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R})$. The global minimum of $V_{\text{M-SBL}}(\boldsymbol{\gamma})$ is therefore achieved at $\{\boldsymbol{\gamma}\in\mathbb{R}_+^N:\boldsymbol{A}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}_{\mathsf{opt}}\right)\boldsymbol{A}^{\mathsf{T}}=\boldsymbol{0}\}$. Hence, the set to which our algorithm converges contains all the points achieving the global minimum of $V_{\text{M-SBL}}(\boldsymbol{\gamma})$.

- Since $V_{\text{M-SBL}}(\boldsymbol{\gamma})$ is a function of $\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}$, the smallest set to which M-SBL can converge is $\{\boldsymbol{\gamma}\in\mathbb{R}_+^N:\boldsymbol{A}\left(\boldsymbol{\Gamma}-\boldsymbol{\Gamma}_{\mathsf{opt}}\right)\boldsymbol{A}^{\mathsf{T}}=\boldsymbol{0}\}$. The $\boldsymbol{\gamma}_k$ output by our algorithm converges to the union of this set with $\boldsymbol{0}$.

- It can be shown that the algorithm is guaranteed to converge to a sparse solution,

where, by sparse solution, we mean one with no more than m nonzero entries. Given any s-sparse vector $\boldsymbol{\gamma}_{\mathsf{opt}}$ and sensing matrix $\boldsymbol{A}$, we can always construct a pair $(\boldsymbol{x}_c, \boldsymbol{y}_c)$ such that $\boldsymbol{y}_c = \boldsymbol{A}\boldsymbol{x}_c$ and $\boldsymbol{x}_c = \boldsymbol{\Gamma}_{\mathsf{opt}}^{1/2}(\boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}^{1/2})^{\dagger}\boldsymbol{y}_c$. By [23, Theorem 1], $\boldsymbol{\gamma}_{\mathsf{opt}}$ is the global minimizer of the SBL cost function constructed under a noiseless measurement model using $\boldsymbol{y}_c$ and $\boldsymbol{A}$. Further, from [23, Theorem 2], it is known that every local minimum of the SBL cost function is achieved at a sparse solution (even in the presence of noise). Now, the SBL cost is a function of $\boldsymbol{\Gamma}$ only through $\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}$. Hence, the set $\{\boldsymbol{\gamma} \in \mathbb{R}_+^N : \boldsymbol{A}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathsf{opt}}\right)\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{0}\}$ consists of local minima of this SBL cost function, which implies that the elements of the set are all sparse. Therefore, the algorithm is guaranteed to converge to a sparse solution.

We can extend the above convergence results to the refined algorithm given by (4.40) using the following corollary.

Corollary 5.1. *Consider the modified online M-SBL algorithm given by* (4.40) *and having learning rates satisfying* (4.39)*. Under the assumptions of Theorem 5.1, the sequence* $\boldsymbol{\gamma}_k$ *converges to a point in the set* $\{\boldsymbol{0}\} \cup \{\boldsymbol{\gamma} \in \mathbb{R}_+^N : \boldsymbol{A}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathsf{opt}}\right)\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{0}\}$ a.s. *Further, if* $\mathsf{Rank}\{\boldsymbol{A} \odot \boldsymbol{A}\} = N$*, the sequence* $\boldsymbol{\gamma}_k$ *converges to a point in the set* $\{\boldsymbol{0}, \boldsymbol{\gamma}_{\mathsf{opt}}\}$ a.s.

The proof of the above is similar to that of Theorem 5.1 because the only properties of the sequence $1/k$ (in (4.16)) that are used in Theorem 5.1 are the ones listed in (4.39).

We now consider the more general case where $\bar{\Delta} \geq 1$. As in the previous case, the algorithm can be rewritten as a stochastic approximation recursion as follows:

$$\boldsymbol{\gamma}_l = \boldsymbol{\gamma}_{l-1} + \frac{1}{l}\boldsymbol{f}(\boldsymbol{\gamma}_{l-1}) + \frac{1}{l}\tilde{\boldsymbol{e}}_l, \tag{5.10}$$

where $\boldsymbol{f}(\boldsymbol{\gamma})$ is as defined in (5.3), and

$$\tilde{\boldsymbol{e}}_l \triangleq -\boldsymbol{f}(\boldsymbol{\gamma}_{l-1}) + \frac{1}{\bar{\Delta}} \sum_{t=k_l+1}^{k_l+\bar{\Delta}} \mathrm{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma}_{l-1}) + \widehat{\boldsymbol{x}}(\boldsymbol{y}_t, \boldsymbol{\gamma}_{l-1})\widehat{\boldsymbol{x}}(\boldsymbol{y}_t, \boldsymbol{\gamma}_{l-1})^{\mathsf{T}}\right\}. \tag{5.11}$$

The following theorem characterizes the asymptotic behavior of the above algorithm. Using the theorem, we can also derive a corollary similar to Corollary 5.1. However, we omit the statement to avoid repetition.

Theorem 5.2. *Under the assumptions of Theorem 5.1, the sequence $\boldsymbol{\gamma}_l$ output by the online M-SBL algorithm given by* (5.10) *converges to a point in the set $\{\mathbf{0}\} \cup \{\boldsymbol{\gamma} \in \mathbb{R}_+^N : \boldsymbol{A}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathsf{opt}}\right)\boldsymbol{A}^{\mathsf{T}} = \mathbf{0}\}$* a.s. *Further, if $\mathsf{Rank}\left\{\boldsymbol{A} \odot \boldsymbol{A}\right\} = N$, the sequence $\boldsymbol{\gamma}_l$ converges to a point in the set $\{\mathbf{0}, \boldsymbol{\gamma}_{\mathsf{opt}}\}$* a.s.

Proof. The algorithm given by (5.10) differs from the algorithm given by (5.2) only in the last term. The only place where this term plays a role in the proof in Appendix C.2 is via Lemma C.1. Hence, it suffices to show that $\lim_{l\to\infty} \sum_{i=1}^{l} \frac{1}{i}\tilde{\boldsymbol{e}}_i$ exists and is finite. From (5.11), we get

$$\tilde{\boldsymbol{e}}_l = \mathrm{Diag}\Bigg\{\boldsymbol{\Gamma}_{l-1}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}_{l-1}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1} \left(\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\right\} - \frac{1}{\bar{\Delta}} \sum_{t=k_i+1}^{k_{i+1}} \boldsymbol{y}_t\boldsymbol{y}_t^{\mathsf{T}}\right)\left(\boldsymbol{A}\boldsymbol{\Gamma}_{l-1}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}_{l-1}\Bigg\}.$$

Now the result follows by replacing $\boldsymbol{e}_k$ in the proof of Lemma C.1 with $\tilde{\boldsymbol{e}}_l$. □

We can also get similar convergence results for the improved M-SBL algorithm given by (4.41), as follows.

Corollary 5.2. *Under the assumptions of Theorem 5.1, the sequence $\boldsymbol{\gamma}_l$ output by the improved online M-SBL algorithm given by* (4.41) *converges to a point in the set $\{\mathbf{0}\} \cup \{\boldsymbol{\gamma} \in$*

$\mathbb{R}_+^N : \boldsymbol{A}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathsf{opt}}\right)\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{0}\}$ a.s. *Further, if* $\mathsf{Rank}\left\{\boldsymbol{A} \odot \boldsymbol{A}\right\} = N$, *the sequence* $\boldsymbol{\gamma}_l$ *converges to a point in the set* $\{\boldsymbol{0}, \boldsymbol{\gamma}_{\mathsf{opt}}\}$ a.s.

Proof. Under the assumptions of Theorem 5.1, the improved online algorithm given by (4.41) is equivalent to the original algorithm given by (4.36) except that it uses $\bar{\Delta}$ measurement vectors $\left\{\boldsymbol{y}_t, t = \check{k}_l - \bar{\Delta} + 1, \check{k}_l - \bar{\Delta} + 2, \ldots, \check{k}_l\right\}$ instead of $\bar{\Delta}$ measurement vectors $\{\boldsymbol{y}_t, t = k_l + 1, k_l + 2, \ldots, k_{l+1}\}$ used by the original version. Since the measurement vectors are independent and identically distributed, the rest of the proof is the same as that of Theorem 5.1. □

5.2 Perfectly Correlated Case

In this section, we consider the convergence results for the other extreme value of correlation, i.e., $\boldsymbol{D} = \boldsymbol{I}$. We note that when $\boldsymbol{D} = \boldsymbol{I}$, from (4.1), we get $\boldsymbol{x}_k = \boldsymbol{x}_1 \triangleq \boldsymbol{x}$ for all values of k. This is because the covariance of $\boldsymbol{z}_k$ is assumed to be $\boldsymbol{I} - \boldsymbol{D} = \boldsymbol{0}$. Further, the Kalman filtering/smoothing equations for estimating the sparse vectors (4.19)-(4.24) and (4.26)-(4.31) become independent of $\boldsymbol{\gamma}$. Thus, when $\Delta = 1$, the algorithm simplifies to

$$\boldsymbol{J}_k = \boldsymbol{P}_{k-1}\boldsymbol{A}_k^{\mathsf{T}}\left(\boldsymbol{A}_k\boldsymbol{P}_{k-1}\boldsymbol{A}_k^{\mathsf{T}} + \boldsymbol{R}_k\right)^{-1} \tag{5.12}$$

$$\widehat{\boldsymbol{x}}_k = (\boldsymbol{I} - \boldsymbol{J}_k\boldsymbol{A}_k)\widehat{\boldsymbol{x}}_{k-1} + \boldsymbol{J}_k\boldsymbol{y}_k \tag{5.13}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{J}_k\boldsymbol{A}_k)\boldsymbol{P}_{k-1}, \tag{5.14}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{P}_k$ are the estimates of $\boldsymbol{x}$ and its covariance, respectively, at time k.

When $\boldsymbol{D} = \boldsymbol{I}$, (4.27) becomes $\boldsymbol{P}_{k|k-1} = \boldsymbol{P}_{k-1}$, and hence, here we analyze the converge of $\boldsymbol{P}_k$. Further, when $\boldsymbol{D} = \boldsymbol{0}$, we showed that $\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}$ converges to $\boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}}$. Similarly,

here we show the convergence of $\boldsymbol{A}\boldsymbol{P}_k\boldsymbol{A}^{\mathsf{T}}$.

As before, we first prove that the algorithm converges and then characterize the limit points.

Proposition 5.2. *Under assumptions A1 and A2, the algorithm given by* (5.12)-(5.14) *converges as* $k \to \infty$*, provided* $\boldsymbol{P}_0$*, the initialization of the covariance matrix, has full rank.*

Proof. See Appendix C.3. □

Now that we know the algorithm converges, and the next important question is whether the algorithm converges to the right solution. This is addressed in the following theorem.

Theorem 5.3. *Under assumptions A1 and A2, the sequence* $\widehat{\boldsymbol{x}}_k$ *of the algorithm given by* (5.12)-(5.14) *converges to the true solution almost surely, if the initialization* $\boldsymbol{P}_0$ *is a full rank matrix.*

Proof. See Appendix C.4. □

Remark 1: Theorem 5.3 is very general, and holds under a variety of settings. In particular, it is independent of:

- the sparsity level of the unknown vector $\boldsymbol{x}$
- initializations of $\widehat{\boldsymbol{x}}_0$ and $\boldsymbol{P}_0$, provided $\boldsymbol{P}_0$ has full rank
- measurement noise level and noise correlation, i.e., $\boldsymbol{R}$

5.3 Simulation Results

We use the following setup to evaluate the performance of the algorithm and corroborate the theoretical results. We generate sparse signals of length $N = 60$, each with $s = 6$ nonzero entries. The locations of nonzero coefficients are chosen uniformly at random, and the nonzero entries are independent and identically distributed with zero mean and unit variance. The length of measurement vector is chosen as $m = 20$. The measurement matrices $\boldsymbol{A}_k$ are generated with independent and Gaussian distributed entries with zero mean, and the columns are normalized to have unit Euclidean norm.

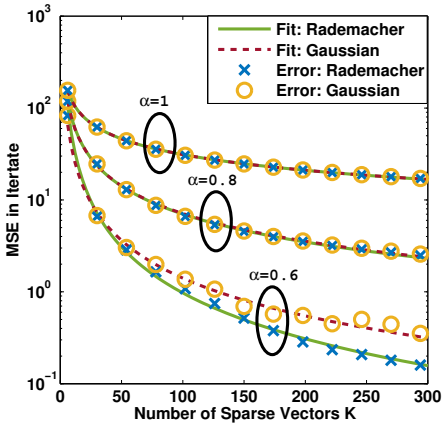
We study the properties of the algorithm for both uncorrelated and highly correlated cases in the following subsections. For the uncorrelated case, we consider the improved online algorithm given by (4.41).

5.3.1 Convergence

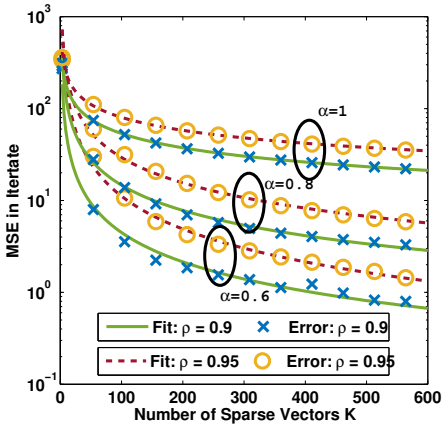
We consider three different learning rates $b_k = 1/k^\alpha$: $\alpha = 0.6, 0.8$ and 1. The maximum delay between the measurement and estimation is taken as $\Delta = 5$. To highlight the convergence behavior, we initialize the hyperparameters with a fixed value $4 \cdot \mathbf{1}$, irrespective of the measurements. The SNR is chosen as 20 dB for all the results in this subsection.

Uncorrelated Case

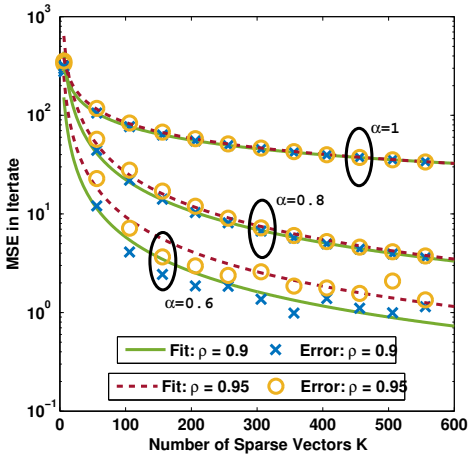
We generate the sparse vectors from two distributions: Gaussian and Rademacher distribution. The mean squared error (MSE) in the estimated hyperparameters when $\bar{\Delta} = 3$ are plotted in Figure 5.1a. The curves labeled `Fit` are the fitted curves on the error using the function: $f(x) = ax^{-p}$ where a and p are parameters. The result for other values of $\bar{\Delta}$ is similar, and we summarize the values of p in Table 5.1. Our observations from the



(a)



(b)



(c)

Figure 5.1: Convergence of the hyperparameters to the true value.

Algo.	**Rademacher Dist.**			**Gaussian Dist.**		
	$\bar{\Delta} = 1$	$\bar{\Delta} = 3$	$\bar{\Delta} = 5$	$\bar{\Delta} = 1$	$\bar{\Delta} = 3$	$\bar{\Delta} = 5$
$\alpha = 0.6$	1.69	1.30	1.17	1.18	1.09	0.96
$\alpha = 0.8$	0.87	0.79	0.72	0.86	0.78	0.71
$\alpha = 1.0$	0.49	0.47	0.43	0.49	0.47	0.43

Table 5.1: Value of error-fit power function parameter p when $\boldsymbol{D} = \mathbf{0}$.

results are as follows:

- *Convergence:* The algorithm converges to the true $\boldsymbol{\gamma}$, and not to the other equilibrium point, $\boldsymbol{\gamma} = \mathbf{0}$, in all cases. This happens even if we initialize the algorithm with very small values such as $10^{-2} \cdot \mathbf{1}$.

- *Sparse vector distribution:* The algorithm works equally well for both Gaussian (which is continuous) and Rademacher distribution (which is discrete), as guaranteed by Theorem 5.2. In particular, it works for the Rademacher distribution even though it was developed by imposing a Gaussian distribution on the nonzero coefficients of the sparse vectors.

- *Learning rate:* The smaller the α, the larger the learning rate b_k, and hence the larger the weightage given to the update term $\mathsf{Diag}\left\{\left(\boldsymbol{I} - \boldsymbol{D}^2\right)\boldsymbol{T}_{k|k+\Delta} - \boldsymbol{\Gamma}_{k-1}\right\}$ in (4.40), leading to faster convergence. Since $1/2 < \alpha \leq 1$ is required for theoretical convergence guarantee, a value of α close to $1/2$ ensures the fastest convergence. However, we have also observed from our experiments that $\alpha \leq 1/2$ leads to even faster convergence. Hence, in practice, one could try using $\alpha \leq 1/2$, but the convergence would not be guaranteed by our analysis.

- *Value of $\bar{\Delta}$:* As $\bar{\Delta}$ increases, the exponent p slightly decreases. This is because when

$\bar{\Delta}$ increases, the hyperparameter $\boldsymbol{\gamma}$ gets updated less frequently. Hence, a lower $\bar{\Delta}$ improves the convergence rate and estimation accuracy, but at the cost of higher average latency and computational complexity. This is further illustrated in the following subsections.

Highly Correlated Case

Next, we study the convergence of our algorithm in the highly correlated case. Figures 5.1b and 5.1c show the MSE in the hyperparameter estimates when $\bar{\Delta} = 3$, for the fixed lag and sawtooth lag schemes, respectively. A few interesting observations from the figures are as follows:

- *Correlation coefficient:* As the correlation coefficient increases, the convergence becomes slower. This is because the approximation in (4.14) becomes loose as the correlation increases, as discussed in Section 4.4.2.

- *Implementation scheme:* We see that the convergence behavior of the fixed lag and sawtooth lag schemes are similar. However, the gap between the curves when the correlation coefficient is 0.9 and 0.95 is smaller for the fixed lag scheme compared to the sawtooth lag scheme. Further discussion about this is provided in Section 4.5.

- *Learning rates:* As observed in the uncorrelated case, the convergence is faster for small values of α. However, the gap between the curves for the two correlation coefficients is wider for smaller values of α. This is because as α decreases, the weightage given to the update term in (4.40) increases, and thus, it becomes more sensitive to the approximation in (4.14).

5.4 Summary

In this chapter, we provided a rigorous convergence analysis of the algorithm presented in Chapter 4. Using empirical simulations we showed that the algorithm output converges to the true value, for the regimes which are not covered by the theoretical results. Hence, the chapter throughly answered the question on uniqueness of sparse solution for the model and algorithms under consideration. With this, we completed the mathematical theory for the second model in this thesis. In the next chapter, we advance to the last sparsity model considered in the thesis, model SM3. 3

Chapter 6

Bayesian Learning Algorithm for Sparse Control Inputs and Observation Matrix

Answering problem Q2 and Q3 for SM3

In this chapter, we consider at the most sophisticated sparsity model where the initial state, sparse inputs and the measurement matrix are unknown (model SM3). The unknown measurement matrices are assume be identical for all the measurements. Further, we also assume that the transition matrix is an all zero matrix, i.e, the states are independent of each other. For this setting, we learn a measurement matrix or a dictionary from a finite set of noisy measurement vectors, such that the measurement vectors admit a sparse representation over the dictionary. This problem is referred to as the dictionary learning (DL) problem. While several solutions are available in the literature, relatively little is known about their convergence and optimality properties. We make progress on this problem by analyzing a Bayesian algorithm for DL. Specifically, we cast the DL problem into the sparse Bayesian learning framework by imposing a hierarchical Gaussian

prior on the sparse vectors. This allows us to simultaneously learn the dictionary as well as the parameters of the prior on the sparse vectors using the expectation-maximization algorithm. The dictionary update step turns out to be a nonconvex optimization problem, and we present two solutions, namely, an alternating minimization (AM) procedure and an Armijo line search (ALS) method. We rigorously analyze the convergence properties of the solution, and show that the ALS procedure is globally convergent. We also analyze the stability of the solution by characterizing its limit points. Further, we prove the convergence and stability of the overall DL-SBL algorithm, and we show that the minima of the cost function of the overall algorithm are achieved at sparse solutions. As a concrete example, we consider the application of the SBL-based DL algorithm to image denoising, and demonstrate the efficacy of the algorithm relative to existing DL algorithms.

6.1 Background

In sparse coding, the signal of interest is represented as a linear combination of a relatively small number of columns of a properly chosen over-complete *dictionary*. The dictionary can be of two types: first, non-adaptive or predefined dictionaries like Fourier, Gabor, discrete cosine transform and wavelet [128]; and second, an adaptive or learned dictionary that is specific to the given class of signals. The use of adaptive dictionaries often leads to more compact representations and better performance in many signal processing applications ranging from image denoising [129–131], audio processing [132,133], and classification tasks [134–140], to name a few. Therefore, we are interested in the dictionary learning problem, where the objective is to find a dictionary over which a set of training signals admits a sparse representation.

Several dictionary learning algorithms for sparse coding have been proposed in the literature such as method of optimal directions (MOD) [141], K-singular value decomposition (K-SVD) [142], dictionary learning with the majorization method (DL-MM) [143], simultaneous codeword optimization (SimCO) [144], parallel atom-updating dictionary learning (PAU-DL) [145], sequential generalization of K-means (SGK) [146], and iterative thresholding and K means (ITKM) [147]. Most of the algorithms involve an iterative procedure, alternately updating the dictionary and the sparse representation, and differ in the cost function used in the dictionary update step. To update the sparse representation, an existing standard sparse signal recovery algorithm is used.

Although the aforementioned algorithms achieve good performance, they require the knowledge of the sparsity level of the system and hand-tuning of various sensitive algorithm parameters. These limitations are handled to some extent by Bayesian algorithms [148–150]. Bayesian algorithms come with an added advantage of not requiring the knowledge of the measurement noise variance. However, the posterior distributions proposed in [148, 150] cannot be derived analytically, and a Gibbs sampler is used for Bayesian inference. The Gibbs sampling based algorithms are computationally demanding as they involve ensemble learning. To overcome this difficulty, [150] also proposes a variational Bayes' based algorithm for dictionary learning by imposing a Gaussian prior on the dictionary elements. The Gaussian prior intuitively models the boundedness of the dictionary elements and helps to obtain closed form expressions for the dictionary update. The closed form expressions results in faster convergence than the Gibbs sampling based Bayesian algorithms. Nonetheless, imposing a Gaussian prior (on a dictionary with no special structure) results in low accuracy and requires a large number of iterations to

converge. Therefore, the choice of Gaussian prior still leaves room for improvement. This motivates us to develop an improved Bayesian dictionary learning algorithm which does not require the knowledge of the sparsity level, or fine-tuning of parameters, while at the same time improving on the recovery performance.

Our proposed dictionary learning algorithm is based on the sparse Bayesian learning (SBL) framework [22, 23]. In the context of sparse signal recovery, SBL is known to offer superior performance compared to algorithms based on convex relation and greedy approaches, and does not require one to tune the algorithm parameters. The basic idea of SBL is to incorporate a parameterized prior on the unknown sparse vectors that encourages sparsity. Specifically, a fictitious Gaussian prior is imposed on the sparse vectors, and the so-called hyperparameters of the Gaussian distribution are determined using Type-II maximum likelihood (ML) estimation. Our approach is different from other Bayesian dictionary learning algorithms as we impose *no prior on the dictionary elements*. Instead, we estimate the dictionary as a deterministic matrix with unit norm columns. The estimation method uses the expectation-maximization (EM) algorithm to simultaneously learn the parameters of the prior and the sparsifying dictionary. The dictionary update step in the EM algorithm turns out to be a quadratic optimization problem with unit norm constraints, which is a nonconvex problem because of the constraint. Since a closed form solution is not available, we propose to employ the alternating minimization (AM) procedure or Armijo line search (ALS) to solve it.

6.2 SBL Based Dictionary Learning

We consider a special case of LDS presented in Chapter 1, where $\boldsymbol{D} = \boldsymbol{0}$. Hence, we have a set of K training signals $\boldsymbol{y}^K = \{\boldsymbol{y}_k \in \mathbb{R}^m\}_{k=1}^K$ such that $\boldsymbol{y}^K$ admits a sparse representation $\boldsymbol{x}^K = \{\boldsymbol{x}_k \in \mathbb{R}^N\}_{k=1}^K$ over an unknown dictionary $\boldsymbol{A} \in \mathbb{R}^{m \times N}$ and is corrupted by noise, i.e.,

$$\boldsymbol{y}_k = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{w}_k, \tag{6.1}$$

where the noise term $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. Our goal is to estimate the K sparse vectors and the measurement matrix $\boldsymbol{A}$, using the knowledge of N. To resolve the ambiguity in amplitude, we assume $\boldsymbol{A}$ has unit norm columns. That is, $\boldsymbol{A} \in \mathbb{O}$, where

$$\mathbb{O} \triangleq \left\{ \boldsymbol{A} \in \mathbb{R}^{m \times N} : \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{A}_i = 1, i = 1, 2, \ldots, N \right\}. \tag{6.2}$$

Motivated by the SBL framework [22, 23], we impose a Gaussian prior on the unknown sparse vectors $\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{0}, \mathsf{Diag}\{\boldsymbol{\gamma}_k\})$, where $\boldsymbol{\gamma}_k \in \mathbb{R}_+^N$. Using this hierarchical model, we first compute the ML estimates $\hat{\boldsymbol{\gamma}}_k$ and $\hat{\boldsymbol{A}}$ of $\boldsymbol{\gamma}_k$ and $\boldsymbol{A}$, respectively. These estimates, in turn, can be used to estimate the sparse vectors as $\hat{\boldsymbol{x}}_k = \mathbb{E}\left\{ \boldsymbol{x}_k | \boldsymbol{y}_k, \hat{\boldsymbol{\gamma}}_k, \hat{\boldsymbol{A}} \right\}$.

We do not assume any structure in $\boldsymbol{A}$ apart from the unit norm columns, and thus we do not impose any prior on $\boldsymbol{A}$. To obtain the ML estimates $\hat{\boldsymbol{\gamma}}_k$ and $\hat{\boldsymbol{A}}$, we need to maximize $p(\boldsymbol{y}^K; \boldsymbol{\Lambda})$, where $\boldsymbol{\Lambda} = \{\boldsymbol{A}, \boldsymbol{\gamma}_k;\, k = 1, 2, \ldots K\} \in \mathbb{O} \times \mathbb{R}_+^{NK}$ is the tuple of unknown parameters.

We now develop an EM procedure to solve the ML estimation problem, equivalently, for minimizing the negative log likelihood $-\log p(\boldsymbol{y}^K; \boldsymbol{\Lambda})$. Thus, the optimization problem to

be solved is $\underset{\boldsymbol{\Lambda}\in\mathbb{O}\times\mathbb{R}_+^{NK}}{\arg\min}\ T(\boldsymbol{\Lambda})$, where the cost function[1] is

$$T(\boldsymbol{\Lambda}) \triangleq \sum_{k=1}^{K} \log\left|\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right| + \boldsymbol{y}_k^{\mathsf{T}}\left(\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{y}_k. \tag{6.3}$$

The EM algorithm treats the unknowns $\boldsymbol{x}^k$ as the hidden data and the observations $\boldsymbol{y}^K$ as known data. It is an iterative procedure with two steps: an expectation step (E-step) and a maximization step (M-step). Let $\boldsymbol{\Lambda}^{(r)}$ be the estimate of $\boldsymbol{\Lambda}$ at the r^{th} iteration. The E-step computes the marginal log-likelihood of the observed data $Q^{(r-1)}$, and the M-step computes the parameter tuple $\boldsymbol{\Lambda}$ that maximizes $Q^{(r-1)}$.

$$\begin{aligned}\textbf{E-step: } Q\left(\boldsymbol{\Lambda};\boldsymbol{\Lambda}^{(r-1)}\right) &= \mathbb{E}_{\boldsymbol{x}^K|\boldsymbol{y}^K;\boldsymbol{\Lambda}^{(r-1)}}\left\{\log p\left(\boldsymbol{y}^K,\boldsymbol{x}^K;\boldsymbol{\Lambda}\right)\right\}\\ \textbf{M-step: } \boldsymbol{\Lambda}^{(r)} &= \underset{\boldsymbol{\Lambda}\in\mathbb{O}\times\mathbb{R}_+^{NK}}{\arg\max}\ Q\left(\boldsymbol{\Lambda};\boldsymbol{\Lambda}^{(r-1)}\right).\end{aligned} \tag{6.4}$$

Simplifying $Q\left(\boldsymbol{\Lambda},\boldsymbol{\Lambda}^{(r-1)}\right)$ we get,

$$\begin{aligned}Q\left(\boldsymbol{\Lambda};\boldsymbol{\Lambda}^{(r-1)}\right) = c_K - \frac{1}{2}\sum_{k=1}^{K}\Big[\log|\boldsymbol{\Gamma}_k| + \mathsf{Tr}\left\{\boldsymbol{\Gamma}_k^{-1}\mathbb{E}\left\{\boldsymbol{x}_k\boldsymbol{x}_k^{\mathsf{T}}|\boldsymbol{y}^K;\boldsymbol{\Lambda}^{(r-1)}\right\}\right\}\Big]\\ - \frac{1}{2\sigma^2}\sum_{k=1}^{K}\mathbb{E}\left\{\left(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{x}_k\right)^{\mathsf{T}}\left(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{x}_k\right)|\boldsymbol{y}^K;\boldsymbol{\Lambda}^{(r-1)}\right\},\end{aligned} \tag{6.5}$$

where c_K is a constant independent of $\boldsymbol{\Lambda}$. We notice that the optimization in the M-step is separable in its variables $\boldsymbol{\Gamma}_k$ and $\boldsymbol{A}$. We get the update of $\boldsymbol{\gamma}_k$ in the M-step as follows (see [22, 23] for the detailed derivation):

$$\boldsymbol{\gamma}_k^{(r)} = \mathsf{Diag}\left\{\boldsymbol{\mu}_k\boldsymbol{\mu}_k^{\mathsf{T}} + \boldsymbol{\Sigma}_{(k)}\right\}, \tag{6.6}$$

[1]With a slight abuse of notation, we define $\boldsymbol{\Gamma}_k = \mathsf{Diag}\{\boldsymbol{\gamma}_k\}$, and not the k^{th} column of the matrix $\boldsymbol{\Gamma}$.

where we define the following quantities:

$$\boldsymbol{\mu}_k \triangleq \mathbb{E}\left\{\boldsymbol{x}_k|\boldsymbol{y}_k;\boldsymbol{\Lambda}^{(r-1)}\right\} \in \mathbb{R}^N \tag{6.7}$$

$$\boldsymbol{\Sigma}_{(k)} \triangleq \mathbb{E}\left\{(\boldsymbol{x}_k-\boldsymbol{\mu}_k)(\boldsymbol{x}_k-\boldsymbol{\mu}_k)^{\mathsf{T}}|\boldsymbol{y}_k;\boldsymbol{\Lambda}^{(r-1)}\right\} \in \mathbb{R}^{N\times N} \tag{6.8}$$

The optimization problem corresponding the dictionary update reduces to

$$\underset{\boldsymbol{A}\in\mathbb{O}}{\arg\min}\ \sum_{k=1}^{K}\mathbb{E}\left\{(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{x}_k)^{\mathsf{T}}(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{x}_k)\Big|\boldsymbol{y}_k;\boldsymbol{\Lambda}^{(r-1)}\right\}. \tag{6.9}$$

The objective function above can be equivalently written as

$$g(\boldsymbol{A}) = -\mathsf{Tr}\left\{\boldsymbol{M}\boldsymbol{Y}^{\mathsf{T}}\boldsymbol{A}\right\} + \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{A}\left(\boldsymbol{\Sigma}-\mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\boldsymbol{A}^{\mathsf{T}}\right\}, \tag{6.10}$$

where $\boldsymbol{M}\in\mathbb{R}^{N\times K}$ has $\boldsymbol{\mu}_k$ as its k^{th} column, $\boldsymbol{Y}\in\mathbb{R}^{m\times K}$ has $\boldsymbol{y}_k$ as its k^{th} column, and $\boldsymbol{\Sigma}\triangleq\sum_{k=1}^{K}\left(\boldsymbol{\Sigma}_{(k)}+\boldsymbol{\mu}_k\boldsymbol{\mu}_k^{\mathsf{T}}\right)\in\mathbb{R}^{N\times N}$. We note that there is no closed form solution to the quadratic optimization with the unit norm column constraints in (6.9). Therefore, we solve the optimization problem using two iterative schemes: AM and ALS.

6.2.1 Alternating Minimization (AM)

The AM procedure updates one column of $\boldsymbol{A}$ at a time, keeping the other columns fixed. If we fix all columns of $\boldsymbol{A}$ except the i^{th} column, the optimization problem reduces to

$$\underset{\boldsymbol{A}_i:\boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{A}_i=1}{\arg\min}\left(\sum_{k=1}^{K}-\boldsymbol{\mu}_k[i]\boldsymbol{y}_k+\sum_{j=1;j\neq i}^{N}\boldsymbol{\Sigma}[i,j]\boldsymbol{A}_j\right)^{\mathsf{T}}\boldsymbol{A}_i. \tag{6.11}$$

Interestingly, the above reduced optimization problem admits a unique closed form solution provided $\sum_{k=1}^{K}\boldsymbol{\mu}_k[i]\boldsymbol{y}_k-\sum_{j=1;j\neq i}^{N}\boldsymbol{\Sigma}[i,j]\boldsymbol{A}_j\neq\boldsymbol{0}$. If otherwise, we skip the update of that particular column and continue with the update of the next column. Therefore,

Algorithm 3 Dictionary Learning via SBL using AM

Input: $\boldsymbol{Y} = \boldsymbol{y}^K$, N and σ^2

Parameters: ϵ_1 and ϵ_2 (stopping thresholds)

Initialize: $r = 0, \boldsymbol{A}^{(0)} = \boldsymbol{1}, \boldsymbol{\gamma}_k^{(0)} = \boldsymbol{1}, k = 1, 2, \ldots, K$

repeat

for $k = 1, 2, \ldots, K$ **do**

#E-Step:

$$\tilde{\boldsymbol{\Phi}} = \left(\sigma^2 \boldsymbol{I} + \boldsymbol{A}^{(r)} \boldsymbol{\Gamma}_k^{(r)} \boldsymbol{A}^{(r)\mathsf{T}}\right)^{-1}$$

$$\boldsymbol{\Sigma}_{(k)} = \boldsymbol{\Gamma}_k^{(r)} - \boldsymbol{\Gamma}_k^{(r)} \boldsymbol{A}^{(r)\mathrm{T}} \tilde{\boldsymbol{\Phi}} \boldsymbol{A}^{(r)} \boldsymbol{\Gamma}_k^{(r)}$$

$$\boldsymbol{\mu}_k = \sigma^{-2} \boldsymbol{\Sigma}_{(k)} \boldsymbol{A}^{(r)\mathrm{T}} \boldsymbol{y}_k$$

$r \leftarrow r + 1$

#M-Step:

$$\boldsymbol{\gamma}_k^{(r)} = \mathsf{Diag}\left\{\boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathsf{T}} + \boldsymbol{\Sigma}_{(k)}\right\}$$

end for

#Update of $\boldsymbol{A}$ (also part of the M-Step)

Initialize AM: $u = 0$, $\boldsymbol{A}^{(r,0)} = \boldsymbol{A}^{(r-1)}$

$$\boldsymbol{\Sigma} = \sum_{k=1}^{K} \left(\boldsymbol{\Sigma}_{(k)} + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathsf{T}}\right), \ \boldsymbol{M} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K]$$

repeat

$u \leftarrow u + 1$

for $i = 1, 2, \ldots, N$ **do**

$$\boldsymbol{v}_i^{(r,u)} = \left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}}\right)_i - \sum_{j=1}^{i-1} \boldsymbol{\Sigma}[i,j] \boldsymbol{A}_j^{(r,u)} - \sum_{j=i+1}^{N} \boldsymbol{\Sigma}[i,j] \boldsymbol{A}_j^{(r,u-1)}$$

$$\boldsymbol{A}_i^{(r,u)} = \begin{cases} \frac{1}{\left\|\boldsymbol{v}_i^{(r,u)}\right\|} \boldsymbol{v}_i^{(r,u)} & \text{if } \boldsymbol{v}_i^{(r,u)} \neq \boldsymbol{0} \\ \boldsymbol{A}_i^{(r,u-1)} & \text{otherwise.} \end{cases}$$

end for

until $\|\boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)}\| < \epsilon_2$

$\boldsymbol{A}^{(r)} = \boldsymbol{A}^{(r,u)}$

until $\|\boldsymbol{A}^{(r)} - \boldsymbol{A}^{(r-1)}\| + \sum_{k=1}^{K} \|\boldsymbol{\gamma}_k^{(r)} - \boldsymbol{\gamma}_k^{(r-1)}\| < \epsilon_1$

Output: $\{\boldsymbol{\mu}_k, k = 1, 2, \ldots, K\}$ and $\boldsymbol{A}^{(r)}$

the dictionary update in the r^{th} iteration of the EM algorithm reduces to the following recursions for $i = 1, 2, \ldots, N$:

$$\boldsymbol{v}_i^{(r,u)} \triangleq \sum_{k=1}^{K} \boldsymbol{\mu}_k[i]\boldsymbol{y}_k - \sum_{j=1}^{i-1} \boldsymbol{\Sigma}[i,j]\boldsymbol{A}_j^{(r,u)} - \sum_{j=i+1}^{N} \boldsymbol{\Sigma}[i,j]\boldsymbol{A}_j^{(r,u-1)} \tag{6.12}$$

$$\boldsymbol{A}_i^{(r,u)} = \begin{cases} \frac{1}{\left\|\boldsymbol{v}_i^{(r,u)}\right\|}\boldsymbol{v}_i^{(r,u)} & \text{if } \boldsymbol{v}_i^{(r,u)} \neq \boldsymbol{0} \\ \boldsymbol{A}_i^{(r,u-1)} & \text{otherwise.} \end{cases} \tag{6.13}$$

where u denotes the AM procedure iteration index. We stop the AM iterations when $\boldsymbol{A}^{(r,u)}$ converges, i.e., its change in successive iterations is small. The pseudo-code for this algorithm, which we call *dictionary learning via SBL (DL-SBL) using AM*, is provided in Algorithm 3.

Remark: For the special case when $\boldsymbol{\Sigma}$ is a diagonal matrix and $\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} \neq \boldsymbol{0}$, the optimization problem (6.9) is separable in the columns of $\boldsymbol{A}$. Then, the AM procedure returns the global minimum of (6.10) in one iteration.

6.2.2 Armijo Line Search (ALS)

The ALS procedure updates the entire matrix $\boldsymbol{A}$ in every iteration instead of updating one column at a time [151–153]. The idea here is to translate the constrained optimization problem into an unconstrained convex optimization problem using Riemannian geometry. The algorithm continuously translates a test point in the direction of a tangent vector at the point, while staying on the manifold, until a reasonable decrease in objective function is obtained, and finally reaches a stationary point. Such a mapping is called a *retraction*,

Algorithm 4 Dictionary Learning SBL using ALS

Input: $\boldsymbol{Y} = \boldsymbol{y}^K$, N and σ^2

Parameters: ϵ_1 and ϵ_2 (stopping thresholds)

Initialize: $r = 0, \boldsymbol{A}^{(0)} = \boldsymbol{1}, \boldsymbol{\gamma}_k^{(0)} = \boldsymbol{1}, k = 1, 2, \ldots, K$

repeat

for $k = 1, 2, \ldots, K$ **do**

#E-Step:

$\tilde{\boldsymbol{\Phi}} = \left(\sigma^2 \boldsymbol{I} + \boldsymbol{A}^{(r)} \boldsymbol{\Gamma}_k^{(r)} \boldsymbol{A}^{(r)\mathsf{T}}\right)^{-1}$

$\boldsymbol{\Sigma}_{(k)} = \boldsymbol{\Gamma}_k^{(r)} - \boldsymbol{\Gamma}_k^{(r)} \boldsymbol{A}^{(r)\mathrm{T}} \tilde{\boldsymbol{\Phi}} \boldsymbol{A}^{(r)} \boldsymbol{\Gamma}_k^{(r)}$

$\boldsymbol{\mu}_k = \sigma^{-2} \boldsymbol{\Sigma}_{(k)} \boldsymbol{A}^{(r)\mathrm{T}} \boldsymbol{y}_k$

$r \leftarrow r + 1$

#M-Step:

$\boldsymbol{\gamma}_k^{(r)} = \mathsf{Diag}\left\{\boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathsf{T}} + \boldsymbol{\Sigma}_{(k)}\right\}$

end for

#Update of $\boldsymbol{A}$ *(also part of the M-Step)*

Initialize ALS: $u = 0$, $\boldsymbol{A}^{(r,0)} = \boldsymbol{A}^{(r-1)}$

$\boldsymbol{\Sigma} = \sum_{k=1}^{K} \left(\boldsymbol{\Sigma}_{(k)} + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\mathsf{T}}\right)$, $\boldsymbol{M} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K]$

repeat

$u \leftarrow u + 1$

$\boldsymbol{Z}^{(r,u-1)} = P_{\boldsymbol{A}^{(r,u-1)}}\left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}^{(r,u-1)} \boldsymbol{\Sigma}\right)$

Compute the smallest integer $p > 0$ such that

$$g\left(R_{\boldsymbol{A}^{(r,u-1)}}\left(\beta^p \alpha \boldsymbol{Z}^{(r,u-1)}\right)\right) - g\left(\boldsymbol{A}^{(r,u-1)}\right) \leq -c\beta^p \alpha \left\|\boldsymbol{Z}^{(r,u-1)}\right\|^2$$

$\boldsymbol{A}^{(r,u)} = R_{\boldsymbol{A}^{(r,u-1)}}\left(\beta^p \alpha \boldsymbol{Z}^{(r,u-1)}\right)$

until $\|\boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)}\| < \epsilon_2$

$\boldsymbol{A}^{(r)} = \boldsymbol{A}^{(r,u)}$

until $\|\boldsymbol{A}^{(r)} - \boldsymbol{A}^{(r-1)}\| + \sum_{k=1}^{K} \|\boldsymbol{\gamma}_k^{(r)} - \boldsymbol{\gamma}_k^{(r-1)}\| < \epsilon_1$

Output: $\{\boldsymbol{\mu}_k, k = 1, 2, \ldots, K\}$ and $\boldsymbol{A}^{(r)}$

is denoted by $R_{\boldsymbol{A}}$. For Riemannian manifolds, the line search method takes the form

$$\boldsymbol{A}^{(r,u)} = R_{\boldsymbol{A}^{(r,u-1)}}\left(\beta^p \alpha \boldsymbol{Z}^{(r,u-1)}\right), \tag{6.14}$$

where $\boldsymbol{Z}^{(r,u-1)}$ is the tangent direction of the cost function at $\boldsymbol{A}^{(r,u-1)}$ and $\beta^p\alpha$ is the Armijo step size. The constants β and α are the parameters of the algorithm. The step size is chosen so that p is the smallest nonnegative integer that satisfies

$$g\left(R_{\boldsymbol{A}^{(r,u-1)}}\left(\beta^p \alpha \boldsymbol{Z}^{(r,u-1)}\right)\right) - g\left(\boldsymbol{A}^{(r,u-1)}\right) \leq -c\beta^p\alpha \left\|\boldsymbol{Z}^{(r,u-1)}\right\|^2, \tag{6.15}$$

where the scalar parameter $c \in (0,1)$. The interested readers are referred to [151] for more details on ALS procedure.

We first note that the feasible set $\mathbb{O}$ is the Cartesian product of N unit spheres in $\mathbb{R}^m$ which are submanifolds of the Euclidean space $\mathbb{R}^m$. Since the Cartesian product of Riemmanian manifolds is a Riemmanian manifold, $\mathbb{O}$ is a Riemmanian manifold. We define the Riemannian metric for $\mathbb{O}$ as $\langle \boldsymbol{A}, \boldsymbol{B}\rangle = \mathsf{Tr}\left\{\boldsymbol{A}^\mathsf{T}\boldsymbol{B}\right\}$ for $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{O}$. The gradient of the objective function g in the Euclidean space is as follows:

$$\nabla g\left(\boldsymbol{A}\right) = -\boldsymbol{Y}\boldsymbol{M}^\mathsf{T} + \boldsymbol{A}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right). \tag{6.16}$$

The tangent space of the Cartesian product of manifolds is the Cartesian product of the tangent spaces. Therefore, we get the tangent space as

$$T_{\boldsymbol{A}} = \left\{\boldsymbol{B} : \boldsymbol{A}_i^\mathsf{T}\boldsymbol{B}_i = 0, \forall i\right\}. \tag{6.17}$$

The i^{th} column of the orthogonal projection onto the tangent space is

$$P_{\boldsymbol{A}}\left(\boldsymbol{Z}\right)_i = \left(\boldsymbol{I} - \boldsymbol{A}_i \boldsymbol{A}_i^{\mathsf{T}}\right) \boldsymbol{Z}_i. \tag{6.18}$$

Thus, the gradient of the restriction of g to $\mathbb{O}$ is $P_{\boldsymbol{A}}\left(\nabla g\left(\boldsymbol{A}\right)\right)$, and we can choose the i^{th} column of the retraction as

$$R_{\boldsymbol{A}}\left(\boldsymbol{Z}\right)_i = \frac{\boldsymbol{A}_i + \boldsymbol{Z}_i}{\|\boldsymbol{A}_i + \boldsymbol{Z}_i\|}. \tag{6.19}$$

We note that the denominator $\|\boldsymbol{A}_i + \boldsymbol{Z}_i\| \neq 0$ when $\boldsymbol{Z}_i$ is the orthogonal projection onto the tangent space from (6.18). We call this algorithm DL-SBL using ALS, and summarize its pseudo-code in Algorithm 4.

6.2.3 Comparison of the two optimization procedures

In this subsection, we compare the AM and the ALS procedures to get insights on how to choose between them.

- *Computational complexity:* We assume that the multiplication of a $p \times q$ matrix with a $q \times r$ matrix requires $\mathcal{O}(pqr)$ flops [120]. Each iteration of the AM procedure has a complexity $\mathcal{O}(mKN + mN^2)$. Typically, $K \gg N$ for accurate estimation, and therefore the complexity order is $\mathcal{O}(mKN)$. Thus, the complexity is linear in m, N and K. On the other hand, the computational complexity of the ALS procedure is also of the order $\mathcal{O}(mKN)$, except for the computation of the step-size parameter m. The complexity of this step depends on c, β and α, and it is hard to determine the precise dependence. However, we have observed in our simulations that the ALS algorithm requires a larger number of iterations and a longer run time to converge compared to the AM procedure for the same initialization. Hence, the AM procedure

is faster than the ALS procedure.

- *Memory Requirements:* Both AM and ALS procedures require $\mathcal{O}(N^2)$ sized memory, as the largest matrix we keep track of has size $N \times N$.

- *Parameter tuning:* The AM procedure does not require tuning of any sensitive parameters. However, the ALS procedure has scalar parameters c, β and α which determine the rate of convergence, but these parameters do not affect the recovery performance of the overall algorithm. Hence, the tuning of the parameters of ALS is not very critical.

Thus, for practical applications, we prefer AM to ALS as it is computationally less expensive and does not require tuning of any parameters. However, ALS has better theoretical convergence guarantees compared to AM algorithm, which we discuss in Section 6.3.

6.2.4 Comparison with other Bayesian techniques

The main differences between our algorithm and the other Bayesian algorithms in the literature are as follows:

1. Our algorithm does not use Gibbs sampling, unlike the algorithms in [148, 150]. Instead, we use a variational evidence framework which obviates the need for generating posterior samples, and thus our algorithm is faster. Moreover, the ensemble learning based algorithms come with no convergence guarantees. We provide rigorous convergence guarantees for our algorithm in Section 6.3.

2. Our algorithm is similar to the Sparse Bayesian dictionary learning with a Gaussian

hierarchical model proposed in [150] except for the prior on the dictionary. The algorithm in [150] uses a Gaussian prior on the dictionary elements to obtain a closed form expression for the EM updates. However, the choice of Gaussian prior was heuristically motivated by the fact that the entries of the dictionary are bounded. Since the dictionary is an arbitrary matrix with unit norm columns, the ideal choice of prior on the dictionary columns is a uniform distribution on the unit $m-$dimensional sphere. Hence, we propose to use no prior (which is equivalent to a uniform prior) on the dictionary and learn the dictionary as a deterministic unknown. Due to the better prior model used, our algorithm outperforms the one in [150] in terms of the reconstruction accuracy. The cost paid for this approach is the extra iterative procedure that is nested within the EM algorithm. Using an optimization procedure within the EM framework may appear to be more computationally demanding than an approach with closed form expressions. Nonetheless, from our simulations, we find that our algorithm requires far fewer number of iterations compared to the algorithm in [150]. Hence, the overall run time of the algorithm is much smaller.[2] In other words, the algorithm in this chapter is an improved version of Gaussian hierarchical model based SBL algorithm with reduced run time and higher accuracy. We corroborate these arguments through numerical simulations in Section 6.5.2 (See Figure 6.2c).

3. Another Bayesian algorithm for dictionary learning is known as multimodal sparse

[2] A similar observation can be found, in the context of sparse signal recovery, in [154]. Iterative reweighted ℓ_2 algorithms are typically slower than iterative reweighted ℓ_1 algorithms, even though the former admits closed form expressions in the iterations.

Bayesian dictionary learning [155]. This algorithm is same as the Gaussian hierarchical model based SBL algorithm with a non-informative prior on the dictionary columns, except that it includes an additional projection step. This step projects the columns of the dictionary to the unit norm sphere to avoid instabilities due to the ambiguity in the amplitude. As in the case of the Gaussian hierarchical model based SBL algorithm, this algorithm has a closed form expression for the M-step. As explained above, the algorithm effectively uses a non-informative prior on the dictionary atoms instead of using a uniform distribution on the $m-$dimensional unit sphere. Further, the convergence guarantees in [155] do not apply to the algorithm that involves the projection step, which is crucial to the success of the algorithm. Since our cost function is carefully designed to handle the amplitude ambiguity, our algorithm outperforms the multimodal sparse Bayesian dictionary learning algorithm. We illustrate this through numerical simulations in Section 6.5.2 (See Figure 6.2c).

6.3 Convergence of Optimization Procedures

In this section, we discuss the convergence properties of the AM and ALS procedures proposed to solve (6.9).

Proposition 6.1 (Function value convergence)**.** *The sequences of cost function values* $\left\{g\left(\boldsymbol{A}^{(r,u)}\right)\right\}_{u\in\mathbb{N}}$ *generated by the AM and the ALS procedures are non-increasing and convergent.*

Proof. See Appendix D.1. □

While above proposition guarantees that the cost function converges, it does not establish the convergence of the iterates. Hence, we study the convergence behavior of the iterates in the next subsections. We start with the following definition.

Definition 6.1 (Nash equilibrium)**.** *The matrix $\boldsymbol{A}$ with unit norm columns is said to be a Nash equilibrium point of* (6.9) *if*

$$g\left(\boldsymbol{A}\right) \leq g\left(\left[\boldsymbol{A}_1, \ldots, \boldsymbol{A}_{i-1}, \boldsymbol{a}, \boldsymbol{A}_{i+1}, \ldots, \boldsymbol{A}_N\right]\right), \tag{6.20}$$

for any unit-norm vector $\boldsymbol{a}$ and for $i = 1, 2, \ldots, N$.

Every column of a Nash equilibrium is optimal when other columns of the dictionary are held fixed, that is, one cannot unilaterally improve the cost function in (6.9) by updating any single column. We now proceed with our analysis of the convergence of the AM procedure in the next subsection.

6.3.1 AM Procedure

The iterative AM procedure can be viewed as a fixed point iteration with the update mapping dictated by the function whose stationary point is sought. The following result shows that the fixed points of the updates generated by the AM procedure are Nash equilibria of (6.9).

Proposition 6.2 (Nash Equlibrium)**.** *Let $G : \mathbb{O} \to \mathbb{O}$ be the update mapping of AM procedure, i.e., $\boldsymbol{A}^{(r,u+1)} = G(\boldsymbol{A}^{(r,u)})$. Then, a matrix $\boldsymbol{A}^*$ is a fixed point of G if and only if $\boldsymbol{A}^*$ is a Nash equilibrium point of the objective function of* (6.9)*. Further, all Nash equilibrium points are stationary points of the cost function.*

Proof. See Appendix D.2. □

Corollary 6.1. *A matrix $\boldsymbol{A}$ with unit norm columns is a Nash equilibrium point of the objective function in* (6.9) *if and only if $\boldsymbol{A}$ satisfies the relation:*

$$\boldsymbol{A}\boldsymbol{L} = \boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right), \tag{6.21}$$

for some diagonal psd matrix $\boldsymbol{L}$.

Proof. The result directly follows from the form of the fixed points shown in the proof of Proposition 6.2. □

We note that the update mapping of the AM procedure does not have a closed form expression owing to the sequential, column-wise update of the dictionary. Due to this, although the above theorem characterizes its fixed points, it is hard to establish the convergence of the iterates. On the other hand, it is possible to show several interesting convergence properties of the iterates in the ALS procedure. We discuss this next.

6.3.2 ALS Procedure

We begin by noting that establishing convergence guarantees for the ALS procedure is challenging because the optimization problem in (6.9) is nonconvex in $\boldsymbol{A}$. In particular, since $\boldsymbol{A}$ is constrained to lie in the set $\mathbb{O}$, the set of all matrices with unit-norm columns, establishing convergence requires analyzing the convergence behavior over Riemann manifolds. Existing results in this direction, e.g., [156–160], consider convex optimization problems, and very few results are known for the nonconvex case. In [161], the authors studied the convergence of the so-called proximal algorithm applied to nonsmooth functions that satisfy the Łojasiewicz inequality around their generalized critical points. Based on this, convergence of iterative solvers for quadratic optimization of a matrix valued variable over

the space of orthogonal matrices was shown in [162]. In [163], quadratic optimization over the space of unit norm vectors was studied. These results, when extended to a matrix setting, lead to a unit norm constraint on the rows of the matrix, and hence are not applicable in our case. Thus, the convergence of an ALS type procedure for a quadratic optimization problem under *unit-norm column* constraints has not been studied in the literature, and requires new analysis.

To discuss the convergence properties of the ALS procedures, we consider an equivalent unconstrained version of the optimization problem in (6.9) as follows:

$$\arg\min_{\boldsymbol{A}} \mathsf{Tr}\left\{-\boldsymbol{M}\boldsymbol{Y}^{\mathsf{T}}\boldsymbol{A} + \frac{1}{2}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right\} + \delta_{\text{norm}}(\boldsymbol{A}). \tag{6.22}$$

Here, we define δ_{norm} as a barrier function corresponding to the feasible region of (6.9):

$$\delta_{\text{norm}}(\boldsymbol{A}) \triangleq \begin{cases} 0, & \text{if } \boldsymbol{A} \in \mathbb{O} \\ \infty, & \text{otherwise.} \end{cases} \tag{6.23}$$

Also, let $\tilde{g} : \mathbb{R}^{m\times N} \to \mathbb{R}$ denote the objective function of (6.22). The critical points of (6.9) are the points where the subgradient of $\tilde{g}$ vanishes. [3]

Theorem 6.1 (Convergence of iterates)**.** *The sequence output by the ALS procedure,* $\{\boldsymbol{A}^{(r,u)}\}_{u\in\mathbb{N}}$*, is globally convergent.*

Proof. See Appendix D.3. □

The above result guarantees that the iterates of the ALS procedure converge irrespective of the initial point. However, it does not ensure that the algorithm converges to the same point irrespective of the initialization. Such a guarantee exists only if the cost function

[3] We note that we use an extended definition of sub-gradient as the function $\tilde{g}$ is non-convex.

has only one limit point. Hence, we next characterize the properties of the limits points of the sequence of iterates.

Proposition 6.3 (Characterization of limits)**.** *The limit* $\boldsymbol{A}^{(r)}$ *of the sequence* $\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$ *generated by the ALS procedure satisfies the relation:*

$$\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}^{(r)}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right) = \boldsymbol{A}^{(r)}\boldsymbol{L}, \tag{6.24}$$

for some diagonal matrix $\boldsymbol{L}$*. Moreover,*

1. $\boldsymbol{A}^{(r)}$ *is a Nash equilibrium point of* (6.9) *if and only if* $\boldsymbol{L}$ *is a positive semidefinite matrix.*

2. $\boldsymbol{A}^{(r)}$ *is a local minimum if and only if* $\boldsymbol{L} + \boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}$ *is a positive semidefinite matrix. Further,* $\boldsymbol{A}^{(r)}$ *is a strict local minimum if and only if* $\boldsymbol{L} + \boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}$ *is a positive definite matrix.*

Proof. See Appendix D.4. □

We make the following observations from the above results:

- As in the case of the AM procedure, the update mapping of ALS is not available in closed form because of the step size selection process. However, the results characterize the fixed points of the mapping.

- The initialization $\boldsymbol{A}^{(r,0)}$ need not be a feasible point of (6.9). Because of the retraction step which projects the iterates to the feasible set, the algorithm can be initialized from any bounded matrix.

- The results are independent of the estimates from the outer iteration loop of the EM algorithm and the dimension of the dictionary. Thus, the results are applicable to any quadratic cost function of the form (6.9).

- Given $\boldsymbol{A}^{(r)}, \boldsymbol{M}, \boldsymbol{Y}$ and $\boldsymbol{\Sigma}$, the conditions for the Nash equilibrium and local minimum are easily verifiable.

Now, for any first order method such as the ALS procedure, the best guarantees one can obtain are that it converges to a stationary point. Further, we can determine whether the stationary point is a local minimum using the test in step 2 of Proposition 6.3. Beyond this, the only guarantee one can provide for first order methods is that of stability of the limit points. Stability implies that the algorithm converges to a limit point whenever it is initialized close enough to it. Formally, we define stability as follows:

Definition 6.2 (Stability)**.** *Let $G : \mathbb{O} \to \mathbb{O}$ be the update mapping of an iterative algorithm, i.e., $\boldsymbol{A}^{(r,u+1)} = G(\boldsymbol{A}^{(r,u)})$. Also, we let $G^{(u)}(\cdot)$ denote the result of u applications of G:*

$$G^{(1)}(\boldsymbol{A}) = G(\boldsymbol{A}); \quad G^{(u+1)}(\boldsymbol{A}) = G\left(G^{(u)}(\boldsymbol{A})\right). \tag{6.25}$$

The matrix $\boldsymbol{A}^*$ said to be a stable point of the iterative algorithm if, for every neighborhood $\mathcal{U}$ of $\boldsymbol{A}^*$, there exists a neighborhood $\mathcal{V}$ of $\boldsymbol{A}^*$ such that, for all $\boldsymbol{A} \in \mathcal{V}$ and any positive integer u, it holds that $G^{(u)}(\boldsymbol{A}) \in \mathcal{U}$.

We have the following characterization of the stability of the fixed points of the ALS procedure, based on whether the fixed point is a local minimum or not.

Theorem 6.2 (Stability)**.** *Let $\boldsymbol{A}^{(r)}$ be a limit point of the sequence $\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$ generated by the ALS procedure. Then,*

(i) If $\boldsymbol{A}^{(r)}$ is not a local minimum of $\tilde{g}$, then $\boldsymbol{A}^{(r)}$ is not a stable point of the ALS procedure.

(ii) If $\boldsymbol{A}^{(r)}$ is a strict local minimum of $\tilde{g}$, then the algorithm converges to $\boldsymbol{A}^{(r)}$ if the initial point $\boldsymbol{A}^{(r,0)}$ is sufficiently close to $\boldsymbol{A}^{(r)}$.

Proof. See Appendix D.5. □

An implication of Theorem 6.2 is that the ALS procedure converges to a local minimum of the cost function, except when the initial condition is carefully constructed to be adversarial in nature. Also, as in the previous case, the results are independent of the estimates from the outer iteration loop of the EM algorithm and the dimension of the dictionary. Thus, Theorem 6.2 is applicable to any optimization of the form (6.9).

In this section, we have analyzed the convergence properties of the inner loop in the M-step of EM algorithm. Our analysis guarantees that the optimization procedure has good converge properties. As a consequence, and by virtue of the well-known properties of the EM algorithm, DL-SBL is globally convergent. Next, we formally prove the convergence of the overall DL-SBL algorithm and analyze the minima of the DL-SBL cost function given by (6.3).

6.4 Analysis of DL-SBL Algorithm

The DL-SBL algorithm is not an EM algorithm in the strict sense because the M-step of the DL-SBL is not guaranteed to converge to the global minimizer, unlike the conventional EM. However, DL-SBL inherits many good properties of EM such as a monotonic reduction of the cost function. In this section, we build on the results in Section 6.3 and study the

characteristics of the DL-SBL algorithm and the cost function.

6.4.1 Convergence of DL-SBL

We start by stating the following result, which asserts that the DL-SBL cost converges.

Proposition 6.4. *Suppose that $\sigma^2 > 0$. The sequence $\left\{T(\mathbf{\Lambda}^{(r)})\right\}_{r\in\mathbb{N}}$ generated by the DL-SBL algorithm via ALS procedure converges to $T(\mathbf{\Lambda}^*)$ for some $\mathbf{\Lambda}^* \in \mathbb{O} \times \mathbb{R}_+^{KN}$.*

Proof. See Appendix D.6. □

Next, we characterize the properties of the iterates generated by the algorithm.

Theorem 6.3. *Suppose that $\sigma^2 > 0$. The iterates $\left\{\mathbf{\Lambda}^{(r)}\right\}_{r\in\mathbb{N}}$ of the outer loop of the DL-SBL algorithm converge to the set of stationary points of the DL-SBL cost function given by (6.3). Moreover, if a limit point $\mathbf{\Lambda}^*$ of the sequence $\left\{\mathbf{\Lambda}^{(r)}\right\}_{r\in\mathbb{N}}$ is not a local minimum of T, then $\mathbf{\Lambda}^*$ is not a stable point of the ALS procedure.*

Proof. See Appendix D.7. □

The above results guarantee that the cost function values $\left\{T(\mathbf{\Lambda}^{(r)})\right\}$ converge to $T(\mathbf{\Lambda}^*)$ for some stationary point $\mathbf{\Lambda}^*$. They also guarantee that the sequence of iterates converges to a compact and connected subset of a level set of the cost function, although it does not necessarily converge to a single point. Theorem 6.3 also gives insights to the stability of the fixed points of the algorithm, similar to Theorem 6.2. Further, as in the case of the results in Section 6.3, the above results hold for any values of system dimensions: m, N, and K, and sparsity level s.

The next question that we address is on how good the final solution of DL-SBL is. We answer this question by analyzing the minima of the DL-SBL cost function given by (6.3).

6.4.2 Analysis of Minima of The Cost Function

First, note that, in the context of dictionary learning, the problem of finding the sparse representation of a given set of vectors $\boldsymbol{y}^K$, uniqueness of the solution is defined up to an unavoidable permutation of the unit-norm columns of $\boldsymbol{A}$ and rows of $\boldsymbol{X}$, where $\boldsymbol{X} \in \mathbb{R}^{N \times K}$ is the matrix obtained by stacking the sparse vectors $\boldsymbol{x}_k$. We now present necessary conditions for the uniqueness of the solution:

Proposition 6.5. *Consider the dictionary learning problem under noiseless condition $\sigma^2 = 0$, i.e, for any given $\boldsymbol{Y}$, the problem of finding matrices $\boldsymbol{A}$ and $\boldsymbol{X}$ such that $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$, the columns of $\boldsymbol{A}$ have unit norm and the columns of $\boldsymbol{X}$ have at most s non-zero entries. The solution to the problem is unique only if the following conditions are satisfied:*

$$\mathsf{Rank}\{\boldsymbol{X}\} = N \tag{6.26}$$

$$\mathsf{Rank}\{\boldsymbol{A}_{\mathcal{S}_k}\} = |\mathcal{S}_k| < m, \tag{6.27}$$

where $\mathcal{S}_k$ is the support of $\boldsymbol{x}_k$ and $\boldsymbol{A}_{\mathcal{S}_k} \in \mathbb{R}^{m \times |\mathcal{S}_k|}$ is the submatrix of $\boldsymbol{A}$ formed by the columns indexed by $\mathcal{S}_k$. Further, for the special case of $\max\limits_{k=1,2,\ldots,K} \|\boldsymbol{x}_k\|_0 = 1$, the conditions are sufficient.

Proof. See Appendix D.8. □

We note that the necessary conditions required to ensure the uniqueness of the solution of the dictionary learning problem is applicable for any dictionary learning algorithm, and in particular, DL-SBL. Next, we establish that the cost function in (6.3), when minimized, has the desired global minima.

Theorem 6.4. *Suppose the tuple* $(\boldsymbol{A}^*, \boldsymbol{X}^*)$ *satisfies the necessary conditions* (6.26) *and* (6.27). *Also, let* $\left\{\boldsymbol{\Gamma}_k^* \in \mathbb{R}^{N\times N}\right\}_{k=1}^K$ *be a set of nonnegative diagonal matrices denoting the covariance matrix of the sparse vectors such that*

$$\boldsymbol{x}_k^* = \boldsymbol{\Gamma}_k^{*1/2}\left(\boldsymbol{A}^*\boldsymbol{\Gamma}_k^{*1/2}\right)^\dagger \boldsymbol{y}_k \qquad \text{and} \qquad 0 < c < \min_{k=1,2,\dots K} \gamma_k^*, \tag{6.28}$$

where γ_k^* *is the smallest nonzero entry of* $\boldsymbol{\Gamma}_k^*$ *and* c *is a universal constant. Then, as the noise variance* $\sigma^2 \to 0$, *the global minimum of* (6.3) *is achieved at* $\left(\boldsymbol{A}^*\boldsymbol{P}, \{\boldsymbol{P}\boldsymbol{\Gamma}_k^*\boldsymbol{P}\}_{k=1}^K\right)$ *where* $\boldsymbol{P}$ *is a signed permutation matrix.*

Proof. See Appendix D.9. □

We note that the sparsest solution of (6.3) is $(\boldsymbol{A}^*, \boldsymbol{X}^*)$ due to (6.27). Although we assume that the necessary conditions (6.26) and (6.27) hold, the theorem holds true under the mild condition that

$$\max_{k=1,2,\dots,K} \|\boldsymbol{x}_k\|_0 < m. \tag{6.29}$$

However, under the above condition, uniqueness is not guaranteed, i.e., solutions with suboptimal sparsity may also globally minimize the cost function.

We know that the DL problem is NP-hard [164]. Thus, it is not surprising that the cost function obtained using SBL framework may have multiple local minima. Nonetheless, extending the results of the original SBL algorithm on sparse recovery [23], we can show that all the local minima of the function are achieved at sparse solutions.

Theorem 6.5. *Every* $\boldsymbol{\gamma}_k$ *corresponding to the local minimum of the DL-SBL cost function* (6.3) *is at most* $m-$*sparse, regardless of the value of noise variance* σ^2.

Proof. See Appendix D.10. □

Table 6.1: Comparison of ALS convergence behaviour with varying step size parameters β and α

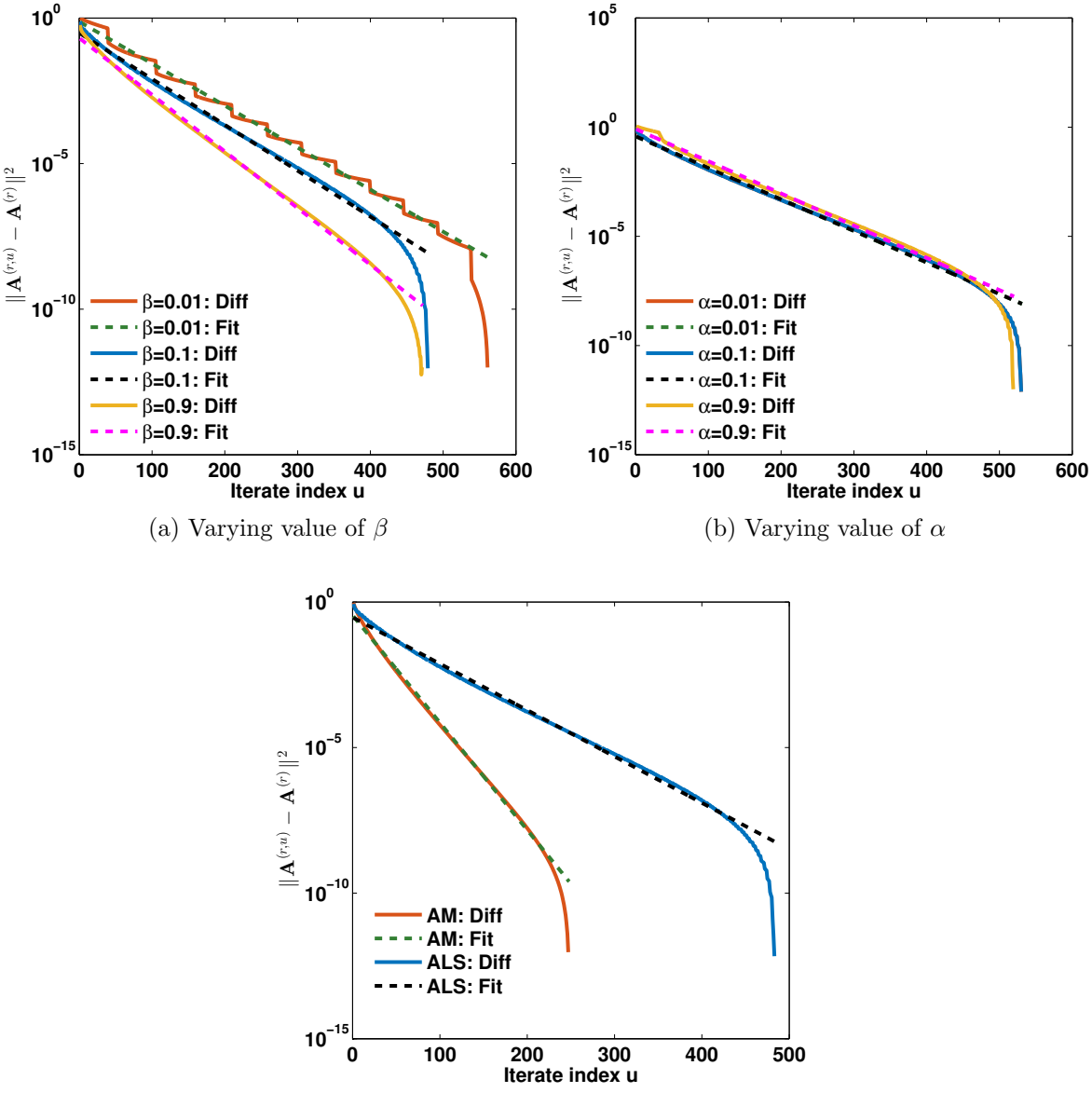
Setting		**Fit parameters**		**no. of iterations**	**run time (s)**
		a	b		
$\alpha = 0.1$	$\beta = 0.01$	-0.034	-0.093	565.04	1.33
	$\beta = 0.1$	-0.036	-1.102	490.09	1.5
	$\beta = 0.9$	-0.044	-1.554	480.63	13.68
$\beta = 0.1$	$\alpha = 0.01$	-0.036	-1.118	494.26	1.55
	$\alpha = 0.1$	-0.036	-1.102	490.09	1.50
	$\alpha = 0.9$	-0.037	-0.226	486.60	1.51

6.5 Simulation Results

We use the following simulation setup to evaluate the performance of the algorithms and validate the theoretical convergence results in Section 6.5.1 and Section 6.5.2. The locations of nonzero coefficients are chosen uniformly at random, and the nonzero entries are independent and identically Gaussian distributed with zero mean and unit variance. The length of measurement vector is chosen as $m = 20$ and SNR = 20 dB. The columns of dictionary matrix $\boldsymbol{A}$ are drawn uniformly from the surface of the m-dimensional unit hypersphere [165].

6.5.1 Convergence

To study the convergence of the AM procedure, we take size of training data set as $K = 1000$. We generate sparse signals of length $N = 60$, each with $s = 6$ nonzero entries. We look at the first iteration ($r = 1$) of the EM algorithm because that requires the maximum number of inner iterations to converge, and thus illustrates the convergence behavior well.



(a) Varying value of β

(b) Varying value of α

(c) Comparison of AM and ALS

Figure 6.1: Convergence of ALS procedure ((a), (b)) and comparison with AM (c), with $K = 1000$, $m = 20$, $N = 60$, $s = 6$, and SNR $= 20$ dB, for the first iteration of EM algorithm.

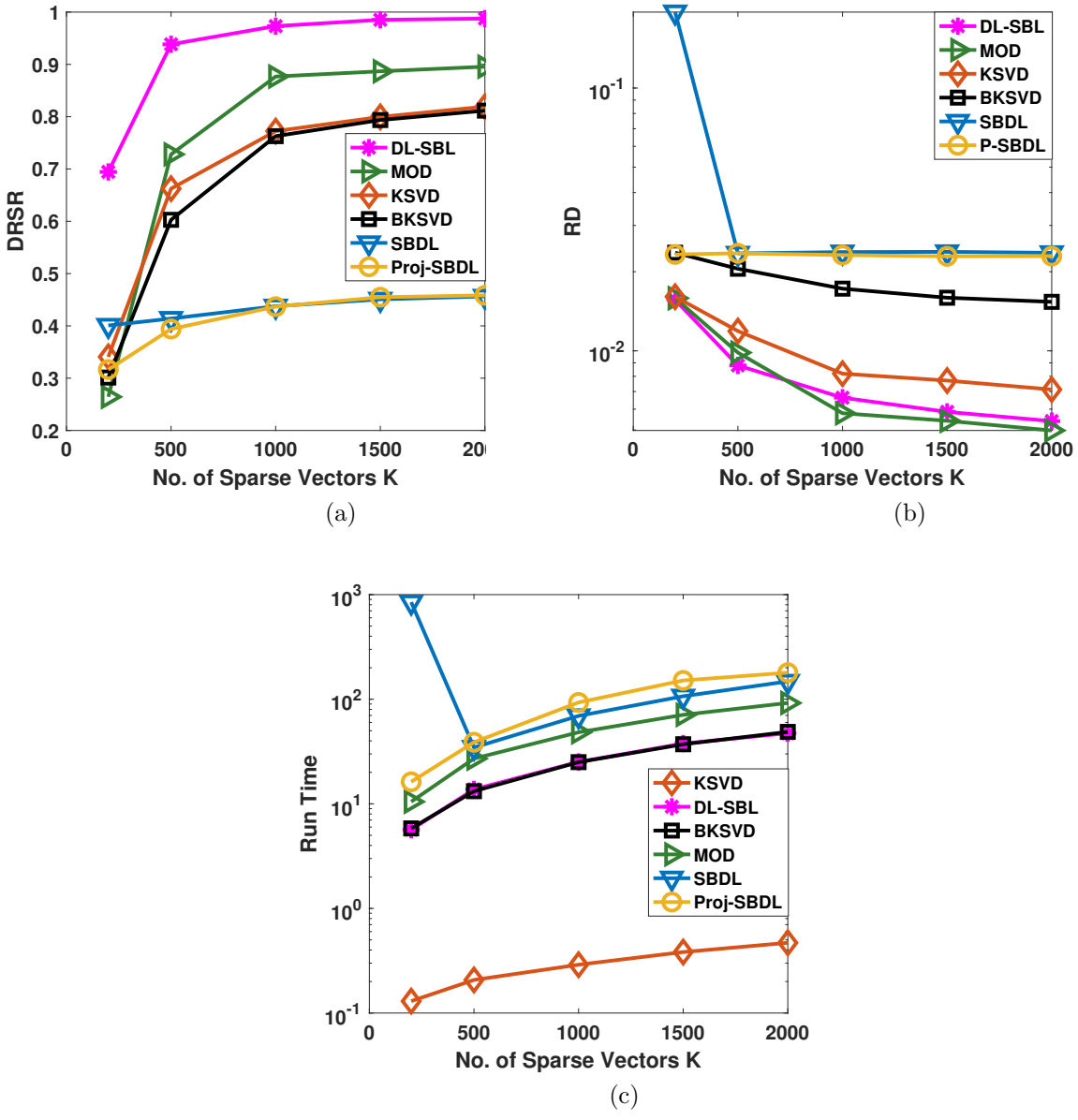


Figure 6.2: Comparison of DL-SBL with KSVD, MOD, Gaussian hierarchical model based SBL algorithm, multimodal sparse Bayesian dictionary learning, and Bayesian KSVD, when the number of input vectors is varied. The performance of DL-SBL is superior to the other three algorithms.

Table 6.2: Comparison of ALS and AM convergence behavior

Algo.	**Fit parameters**		**no. of iterations**	**run time (s)**
	a	b		
AM	-0.0427	-0.4603	248.95	0.5828
ALS	-0.0361	-1.1022	490.09	1.5020

6.5.2 Performance of the Algorithms

In this subsection, we compare the performance of our algorithms with other popular algorithms in literature. Here, we do not show separate curves for DL-SBL using the ALS and AM algorithms, as their performances are virtually identical.

For fairness of comparison, the noise level information is provided to all algorithms. For SimCo, KSVD and MOD, it is used to set the error threshold in the orthogonal matching pursuit (OMP) step of the algorithm; the threshold is set to be 1.15 times the noise variance. For DL-SBL, GAMP, Gaussian hierarchical model based SBL, and Bayesian KSVD, the noise variance is an input to the algorithm. We use the version of Gaussian hierarchical model based SBL and Bayesian KSVD which do not learn the noise level, but take the noise level as an input.

Synthetic Data

We use the same setup as in [142]. We generate sparse signals of length $N = 50$, each with $s = 3$ nonzero entries. We let $\hat{\boldsymbol{x}}_k$ and $\boldsymbol{x}_k$ denote the estimate and true value of the sparse vector, respectively and $\hat{\boldsymbol{A}}$ and $\boldsymbol{A}$ denote the estimate and true value of the dictionary, respectively. We use the following metrics evaluating the performance.

(i) Dictionary recovery success rate (DRSR) [142], which is the fraction of successfully recovered columns of the dictionary. A column is said to be successfully recovered if

the magnitude inner product between the column in the true dictionary and any of the estimated dictionary columns exceeds 0.99.

(ii) Relative distortion (RD) [144], defined as:

$$\text{RD} \triangleq \frac{\sum_{k=1}^{K} \|\hat{\boldsymbol{A}}\hat{\boldsymbol{x}}_k - \boldsymbol{A}\boldsymbol{x}_k\|^2}{\sum_{k=1}^{K} \|\boldsymbol{A}\boldsymbol{x}_k\|^2}. \tag{6.30}$$

(iii) Run time, which is the time required to complete the computations. It measures the computational complexity.

We refer to the DRSR and RD metrics jointly as the recovery performance of the algorithm. These two metrics are equally important due to non-uniqueness of the solution. Any solution of the form $\{\boldsymbol{A}\boldsymbol{P}, \boldsymbol{P}\boldsymbol{x}_k, k = 1, 2, \ldots, K\}$, where $\boldsymbol{P}$ is a signed permutation matrix[4] is a solution to the dictionary learning problem. Thus, the error metric $\frac{\|\boldsymbol{A}-\hat{\boldsymbol{A}}\|^2}{\|\boldsymbol{A}\|^2}$ does not account for the inherent non-uniqueness of the solution. Hence, we use DRSR as a measure of how well the dictionary is recovered, and RD is a measure of how well the recovered solution matches with the measurements.

Figure 6.2 compares the proposed algorithm with the following algorithms:

- KSVD [142]
- MOD [141]
- Gaussian hierarchical model based SBL algorithms [150] (labeled as `SBDL`)
- Multimodal sparse Bayesian dictionary learning [155, 166] (labeled as `Proj-SBDL`)

[4]A matrix is said to be a signed permutation matrix if it has exactly one nonzero entry which is either 1 or -1 in each row and each column.

- Bayesian KSVD [167].

For the Gaussian hierarchical model based SBL, the best performance is achieved when the prior imposed is non-informative, and therefore, we use that version of the algorithm for comparison.

The performance of all the algorithms improve with K, as more information about the dictionary is available to the algorithm. The DL-SBL algorithm outperforms the other algorithms in terms of both DRSR and RD. The run time demanded by our algorithm is larger than K-SVD, but it is lower than the other two algorithms.

The Gaussian hierarchical model based SBL and multimodal sparse Bayesian dictionary learning have similar performance except for $K = 200$. When the number of measurements is very small $K = 200$, the Gaussian hierarchical model based SBL algorithm fails to converge and thus, the run time is higher and the recovery performance is poorer. The extra projection step used in the multimodal sparse Bayesian dictionary learning eliminates such instabilities. As the value of K increases, the algorithm converges, and the recovery performance improves. However, in the regime shown in Figure 6.2, the performance of both the algorithms is inferior to the other algorithms in the literature. This observation agrees with the intuitive explanation presented in Section 6.2.4 that the Gaussian hierarchical model based SBL algorithm requires a larger number of measurements compared to the DL-SBL algorithm to achieve good performance.

Image Denoising

We next consider the application of DL to the problem of image denoising. Here, the goal is to remove zero-mean white and homogeneous Gaussian additive noise from a given image. We adopt the same simulation setup as in [142], and use 10 randomly chosen gray

Table 6.3: Comparison of PSNR values of different algorithms with varying noise variance

Noise Standard Deviation	5	10	15	25
SimCo	38.9843	33.7205	30.8103	27.3856
KSVD	**39.0861**	33.8418	30.8928	27.3751
MOD	38.8720	33.8818	**31.0586**	27.5354
DL-SBL	39.0680	**33.9115**	31.0513	**27.6371**
GAMP	38.7975	33.7574	30.9353	27.4408
BKSVD	39.0317	33.8861	31.0124	27.6041

Table 6.4: Comparison of SSIM values of different algorithms with varying noise variance

Noise Standard Deviation	5	10	15	25
SimCo	0.9643	0.8936	0.8289	0.7396
KSVD	0.9648	0.8946	0.8297	0.7393
MOD	0.9646	**0.8959**	**0.8324**	0.7425
DL-SBL	**0.9650**	0.8958	0.8320	**0.7440**
GAMP	0.9600	0.8876	0.8252	0.7384
BKSVD	0.9644	0.8953	0.8317	0.7439

scale images from the Berkeley segmentation database. The noise standard deviations used in this benchmark are 5, 10, 15, and 25 gray levels. For every image, we learn the dictionary using $K = 6000$ uniformly randomly chosen blocks of size $m = 8 \times 8 = 64$ pixels. The length of the sparse vectors N is taken as 256.

For all the algorithms, once the dictionary is learned, the complete image is reconstructed using the OMP algorithm with the corrupted image and the learned dictionary as inputs and error threshold as 1.15 times the noise variance. We reconstruct the image as 8×8 overlapping blocks which are then combined by averaging the overlapping pixels. The peak SNR (PSNR) and structural similarity index (SSIM) values of the images reconstructed by the following algorithms are shown in Table 6.3 and Table 6.4, respectively. The tables show the median values of the corresponding measures for each noise levels.

- Simultaneous codeword optimization (SimCo) [144];
- K-singular value decomposition (K-SVD) [142];
- Method of optimal directions (MOD) [141].
- Bilinear generalized approximate message passing algorithm (GAMP) [149];
- Bayesian K-SVD (BKSVD) [167]

The results show that the performance of DL-SBL matches that of the other algorithms at all noise levels, and it offers the best performance at a noise level of 25. At smaller noise levels (5 and 10), there is no clear winner as the best PSNR value and the best SSIM value correspond to different algorithms including DL-SBL. At noise level 15, MOD has the best performance. However, the performance of DL-SBL is close to the best performing

algorithm for both metrics for all noise levels. Therefore, the performance of our algorithm is similar to the state-of-the-art algorithms.

6.6 Summary

In this chapter, we analyzed a Bayesian algorithm for jointly recovering a dictionary matrix and a set of sparse vectors from a noisy linear underdetermined training set. We developed the algorithm using the SBL framework, and implemented it using the EM algorithm, with the dictionary matrix and the variances of the entries of the sparse vectors as unknown parameters. The EM algorithm requires one to solve a non-convex optimization problem in the M-step, which we tackled using an AM or ALS procedure. We compared the AM and ALS procedures in terms of their computational complexity and memory requirements. We also provided a rigorous convergence analysis of the proposed optimization procedures. Further, by direct analysis of the cost function involved, we showed that the DL-SBL algorithm is likely to output the sparsest representation of the input vectors. We empirically showed the efficacy of our algorithm compared to existing algorithms, when applied to the image denoising problem. This chapter dealt with the questions Q2 and Q3 on algorithm development and recovery guarantees for the model SM3. So, we have looked at all questions posed in Chapter 1 in Chapter 2 to Chapter 6. Before we conclude the thesis, in the next chapter, we shift gears and provide a new application of structured sparse signal recovery in the context of structural health monitoring.

Chapter 7

Anomaly Imaging for Structural Health Monitoring

Exploiting clustered sparsity

In this chapter, we take a small deviation from the main theme of the thesis and present a structured sparse signal recovery algorithm to a signal model which is not coupled with LDS. The goal here is to develop an algorithm for anomaly imaging for structural health monitoring. Under this model, we are given a set of sensor measurements which are linearly related to an unknown anomaly map. We use a Bayesian framework to explicitly account for both sparsity and cluster pattern structures that are typical of structural anomalies. Hence, the algorithm of this chapter provides excellent reconstruction accuracy by incorporating the available prior information on the anomaly map. Experimental results on a unidirectional composite plate confirms that the algorithm of this chapter outperforms two competing methods in terms of reconstruction accuracy.

7.1 Background

Many critical infrastructures like aircraft, load bearing walls and oil pipelines use fiber reinforced laminate composite materials. Although composite materials are lightweight, strong, and possess excellent fatigue and corrosion resistance, many inter-laminar defects may show no visible evidence [168, 169]. To ensure the integrity of the structure for safe and efficient operation, it is desirable to embed an inspection system within the structure [170]. An active structural health monitoring (SHM) system consists of an array of transducers that can excite and sense wave propagation within the thickness of the structure. The system periodically excites the structure using the transducers sequentially. The resulting waveforms are collected at the remaining transducers which act as sensors. The collected signal is compared to a set of baseline measurements acquired from the structure prior to deployment. The differences between the two signals are characterized using an anomaly metric. The anomaly metric for all actuator-sensor pairs are used to detect and characterize structural damage.

Several algorithms for anomaly mapping have been presented in the literature. Malyarenko and Hinders [171] described a tomography-based approach to image flaws and corrosion on metallic structures. This approach employs the time difference of arrival of the wave between an actuator and a sensor as the measure of the average properties of the actuator-sensor path. A similar approach is studied for composite plates in [172]. Later, Prasad *et al.* [173] successfully located holes on crossply and quasi-isotropic plates using an iterative algebraic reconstruction technique (ART). The algorithm uses the root-mean-square value between the sensor signals and the corresponding baseline signals as the basis for the reconstruction algorithm. Gao *et al.* [174] proposed the reconstruction

algorithm for probabilistic inspection of damage (RAPID) for damage mapping. Although low in complexity, the algorithm design does not consider any particular signal structure associated with the anomaly map. Recently, Zoubi and Mathews [175, 176] developed an anomaly mapping algorithm that uses the sparse nature of damage distribution on structures. However, the sparsity model in the anomaly map is significantly more structured and exhibits a two-dimensional clustered pattern. Therefore, we present a new solution exploiting the two-dimensional clustered sparsity pattern as a prior information to the anomaly map reconstruction problem.

This chapter presents a new algorithm for anomaly imaging, based on ART and the two-dimensional pattern coupled sparse Bayesian learning algorithm. The algorithm takes a set of Lamb wave measurements collected on the structure as input, and outputs an anomaly map from which one can estimate the boundaries of damage on the structure. To the best of our knowledge, this chapter is the first to exploit the two-dimensional clustered sparsity pattern of the anomaly map to improve damage mapping accuracy. Experimental results on a unidirectional composite plate show that the method of this chapter provides more accurate estimates of the damage boundaries than two competing algorithms.

7.2 System Model

We consider an SHM system that uses a set of m transducers where each transducer can act as a wave sensor or a wave actuator, as needed. The structure is excited sequentially by the transducers to obtain an anomaly metric (index) for each signal path connecting the actuator-sensor pairs. The goal is to estimate the anomaly map of the structure, using the $K = m(m-1)$ damage indices thus obtained.

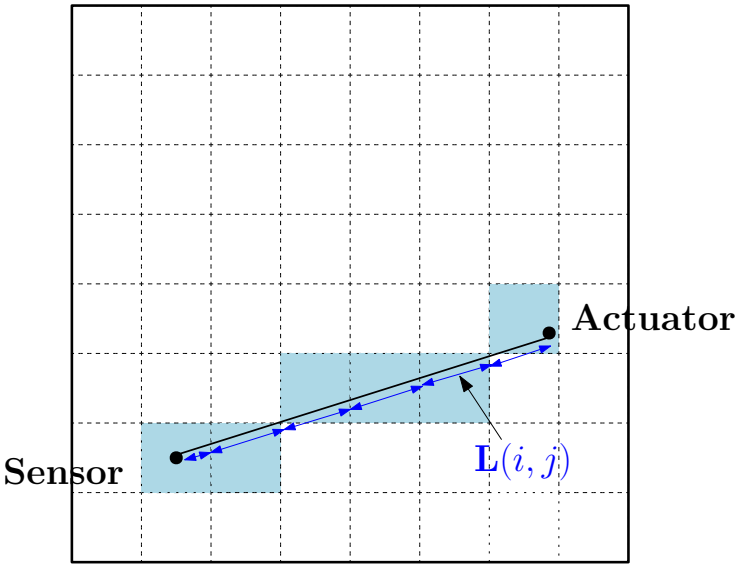


Figure 7.1: The figure shows i^{th} sensor-actuator pair and the direct path between them. The pixels in blue correspond to the nonzero entries of j^{th} row of $\boldsymbol{L}$, and the non-zero value equals the length of the path overlapping the pixel.

To reconstruct the anomaly map, we employ a grid architecture where the spatially continuous map is discretized into N cells or pixels using a grid as shown in Figure 7.1. The damage value associated with each pixel indicates the state of the corresponding part of the structure. Thus, our goal reduces to computing the map value at each pixel using the measured anomaly indices. The mathematical model relating anomaly map values and the damage indices is adopted from the ART framework. Here, the damage indices are assumed to be a linear combination of the pixel values weighted with the length of the direct path between the actuator-sensor pair that crosses the pixel [177]. For instance, in Figure 7.1, the damage index for the signal path between the sensor-actuator pair depends on the pixels which are marked in blue. This linear relationship can be written as

$$\boldsymbol{y} = \boldsymbol{L}\boldsymbol{x}, \tag{7.1}$$

where $\boldsymbol{y} \in \mathbb{R}^K$ is obtained by stacking the damage metrics into a column vector. The column vector $\boldsymbol{x} \in \mathbb{R}^N$ is the vector of the pixel values or the vectorized version of the discretized anomaly map. The $(i,j)^{\mathsf{th}}$ entry of the matrix $\boldsymbol{L} \in \mathbb{R}^{K \times N}$ is the length of i^{th} line segment that overlaps pixel j, as illustrated using Figure 7.1. Hence, the map recovery problem is equivalent to the recovery of $\boldsymbol{x}$ from (7.1) when $\boldsymbol{y}$ and $\boldsymbol{L}$ are known. In the next section, we present the algorithm to recover the discretized anomaly map which utilizes the sparse and clustered structures associated with the unknown map.

7.3 Map Recovery Algorithm

We recover the unknown $\boldsymbol{x}$ by exploiting two underlying structures in the signal:

1. Anomaly areas on the structure are usually small compared to its overall size, which makes $\boldsymbol{x}$ naturally sparse.

2. The anomaly areas occupy a small continuous region of the structure. Therefore, the anomaly map exhibits two-dimensional cluster patterns, also known as block-sparsity.

Several recovery algorithms that exploit block-sparsity have been proposed in the literature. Examples include block-OMP [178], mixed ℓ_2/ℓ_1 norm-minimization [179], group LASSO [180] and block-sparse Bayesian learning [181]. These algorithms require *apriori* knowledge of the block boundaries. However, in our case, the two-dimensional cluster pattern is not known as it depends on the unknown location and shape of the anomaly area. Recently, a new approach has been proposed to tackle the difficulty of unknown block boundaries using the sparse Bayesian learning (SBL) framework [182, 183]. Moreover, the

SBL-based algorithms are known to have superior performance compared to convex relaxation or greedy approaches. Hence, we use the *pattern-coupled (PC) SBL* algorithm to exploit the two-dimensional block-sparse structure.

In the SBL framework, we use a fictitious prior on the unknown signal which promotes the underlying signal structures. To account for the two-dimensional block-sparse structure, a pattern-coupled Gaussian hierarchical prior is imposed on $\boldsymbol{x}$. The use of hyper-parameters associated with each entry of $\boldsymbol{x}$ in the hierarchical Gaussian prior is known to promote sparsity. In addition, two-dimensional block-sparse structure is captured by imposing dependency between the hyper-parameters associated with each entry and that of its neighboring entries: $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}; \mathbf{0}, \boldsymbol{\Gamma})$, where $\boldsymbol{\Gamma} \in \mathbb{R}^{N \times N}$ is a diagonal matrix with diagonal entries:

$$\boldsymbol{\gamma}_i^{-1} = \boldsymbol{\alpha}_i + \beta \sum_{j \in \mathcal{B}(i)} \boldsymbol{\alpha}_j, \tag{7.2}$$

Here, $\boldsymbol{\alpha} \in \mathbb{R}^N$ is a vector of non-negative hyperparameters, $\beta \in [0, 1]$ is the coupling parameter, and $\mathcal{B}(i)$ is the set of neighboring entries of $\boldsymbol{x}_i$ in the two-dimensional signal. Due to the interdependence on the priors, the entry $\boldsymbol{x}_i$ is driven to zero if α_i or any of its neighboring hyperparameters goes to infinity. The shared hyperparameters enables the prior to flexibly model any block-sparse structure, without pre-specifying the block boundaries.

Using the model in (7.2), as in conventional SBL, we use type II maximum likelihood estimation for $\boldsymbol{x}$. In other words, we first estimate the hyperparamters of the imposed prior which in turn yields an estimate of the sparse $\boldsymbol{x}$. The hyperparameters are obtained using the expectation-maximization (EM) algorithm, where the sparse vectors are treated as hidden variables. We summarize the pseudo-code for anomaly mapping in Algorithm 5.

For detailed derivation of the PC-SBL algorithm, please refer to [182, 183].

Algorithm 5 The PC-SBL Recovery Algorithm

Input: $\boldsymbol{y}$ and $\boldsymbol{L}$
  Parameters: Coupling coefficient β, Tolerance ϵ
  Initialize: $\boldsymbol{\alpha}^{(0)}$, $\sigma^{2(0)}$, $c = d = 10^{-4}$
  while $\left\|\boldsymbol{\alpha}^{(r)} - \boldsymbol{\alpha}^{(r-1)}\right\| \leq \epsilon$ **and** $\left|\sigma^{2(r)} - \sigma^{2(r-1)}\right| \leq \epsilon$ **do**
    for $r = 1, 2, \ldots$ **do**
      $\boldsymbol{\gamma}_i = \left(\boldsymbol{\alpha}_i^{(r-1)} + \beta \sum_{j \in \mathcal{B}(i)} \boldsymbol{\alpha}_j^{(r-1)}\right)^{-1}$, $i = 1,2,\ldots,N$
      $\boldsymbol{\Sigma} = (\sigma^{-2(r-1)} \boldsymbol{L}\boldsymbol{L}^{\mathsf{T}} + \mathsf{Diag}\{\boldsymbol{\gamma}\})^{-1}$
      $\boldsymbol{\mu} = \sigma^{-2(r-1)} \boldsymbol{\Sigma} \boldsymbol{L}^{\mathsf{T}} \boldsymbol{y}$
      $\sigma^{2(r)} = K + 2c \left(2d + \|\boldsymbol{y} - \boldsymbol{L}\boldsymbol{\mu}\|^2 + \mathsf{Tr}\{(\} \boldsymbol{L}\boldsymbol{L}^{\mathsf{T}} \boldsymbol{\Sigma})\right)^{-1}$
      for $i = 1, 2, \ldots, N$ **do**
        $\boldsymbol{\alpha}_i^{(r)} = 2 \left(\boldsymbol{\mu}_i^2 + \boldsymbol{\Sigma}_{ii} + \beta \sum_{j \in \mathcal{B}(i)} \boldsymbol{\mu}_j^2 + \boldsymbol{\Sigma}_{jj}\right)^{-1}$
      end for
    end for
  end while
Output: $\boldsymbol{x} = \boldsymbol{\mu}$

Although (7.1) does not assume any model mismatch, PC-SBL can handle noisy measurements. The PC-SBL based reconstruction can also be applied to other ART-based tomographic imaging methods such as MRI, for cancer detection.

7.4 Experimental Results

The experiments described here were conducted on a 41″ wide, 40″ long and 0.1″ thick, unidirectional composite panel made out of 8 IM7/8552 carbon fiber plies. Thirty two piezoelectric transducers were attached to the plate covering the middle $33'' \times 32''$ region of the plate. The excitation signal used was a linear chirp with bandwidth [150,300 kHz] and the resulting waveforms were acquired with a 2×10^6 samples/second sampling rate. First, the baseline signals were collected before impact damage was introduced into the structure. Then, we impacted the structure on different locations to create damage, and

the test signals were acquired after each impact experiment. Other computational details are as follows:

Choice of damage index: We first applied a mode decomposition algorithm based on cross Wigner-Ville-distribution of the received signal. The anomaly indices were computed using the extracted first arriving mode of the measured signal and the baseline signal, as proposed in [175, 176, 184].

Multi-grid architecture: The virtual grid on the structure is assumed to be rectangular with 22×22 pixels. Since the choice of the grid structure is arbitrary, we used the multi-grid architecture to improve the reconstruction accuracy. We reconstructed the map using 20 different grids on the structure, then, interpolated them to obtain a map on a high-resolution grid. The interpolated grids were 200×200 pixels. The final estimate of the anomaly map was obtained by averaging over these 20 maps. Further details on the multi-grid averaging approach can be found in [185].

Algorithm tuning: From our experiments, we have seen that the choice of parameters β and ϵ of Algorithm 5 is not critical. For the results presented here, we choose $\beta = 1$ and $\epsilon = 10^{-6}$. Also, in the algorithm, we adopt a pruning operation for faster convergence. At each iteration, we pruned those small coefficients associated with hyperparameters $\boldsymbol{\alpha}_i$ greater than 10^{11} times the minimum value.

Estimation of anomaly boundaries: The damage area was estimated as the locations on the structure where the estimated map value was greater than some threshold, and the threshold is calculated using training data.

Recovery accuracy metric: We used the Sørenson-Dice index (also known as F1 score), which computes the correlation between two data sets A and B as $\frac{2|A \cap B|}{|A|+|B|}$. Here, the

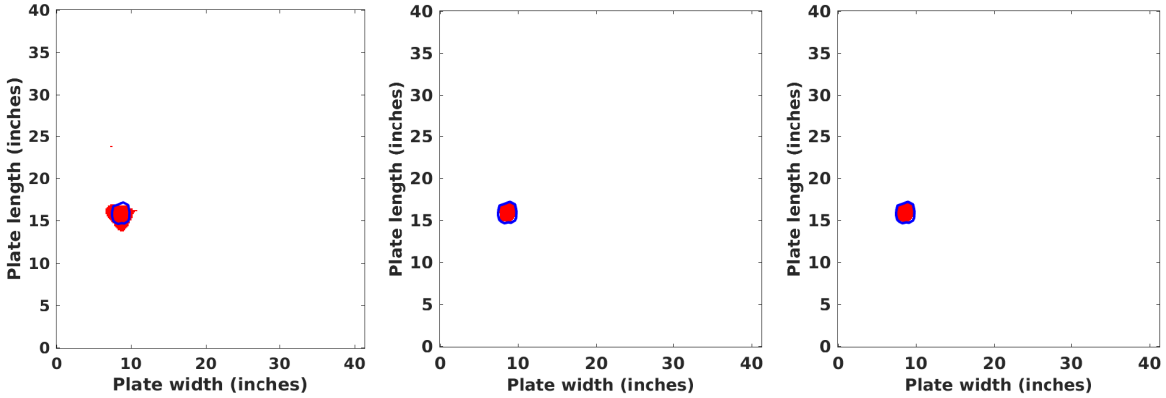
anomaly map obtained using an A-scan device (manual non-destructive evaluation technique) was used as the ground truth.

To illustrate the performance of our algorithm, we compare the algorithm of this chapter based on PC-SBL with two state-of-the-art algorithms: a least-squares (LS) based damage mapping algorithm [185] and a LASSO based damage mapping algorithm [175, 176]. Figure 7.2 shows the reconstructed map of the composite plate obtained using three algorithms after impact experiments. Each row corresponds to an experiment, and each column corresponds to an algorithm. The blue outlines in the maps represent the boundaries of the anomaly estimated using A-scan. The extent of the anomaly estimated by each algorithm is shown in red. We also provide the Sørenson-Dice similarity index of the estimated boundaries in the caption of each figure.

From Figure 7.2, we see that, compared to the LS-based method (first column), the algorithm of this chapter (last column) has fewer false alarms. Also, compared to the LASSO based method (middle column), the algorithm of this chapter gives better estimate of the damage boundaries, which is evident from the Sørenson-Dice index. Overall, the results clearly indicate that the map reconstructed by the PC-SBL algorithm identifies the true anomalies in the structure more closely compared to the other approaches.

7.5 Summary

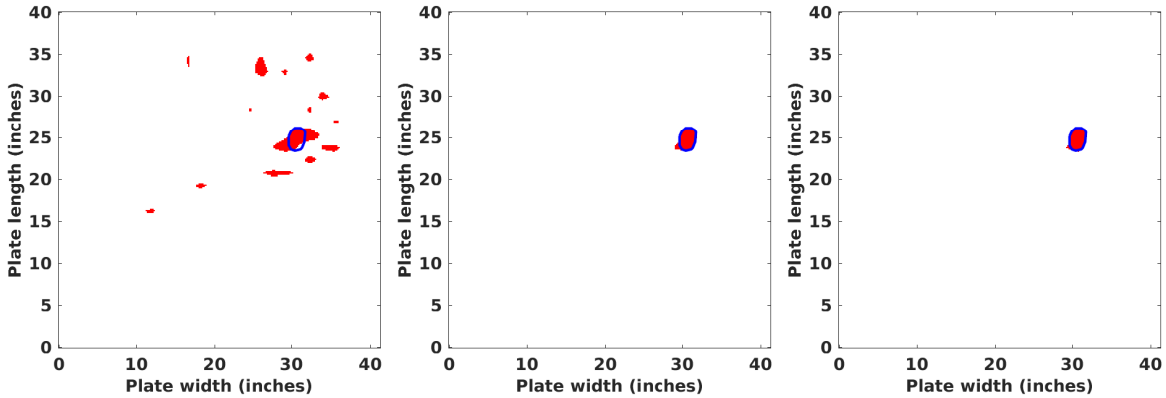
This chapter presented a new algorithm for anomaly map reconstruction for health monitoring of composite structures. We utilized the two-dimensional clustered sparse structure associated with structural damage to present a new map reconstruction algorithm. Using a data set obtained from impact experiments, we demonstrated the superiority of our



(a) Experiment 1: Least-squares Sørenson-Dice index = 0.6827

(b) Experiment 1: LASSO Sørenson-Dice index = 0.9186

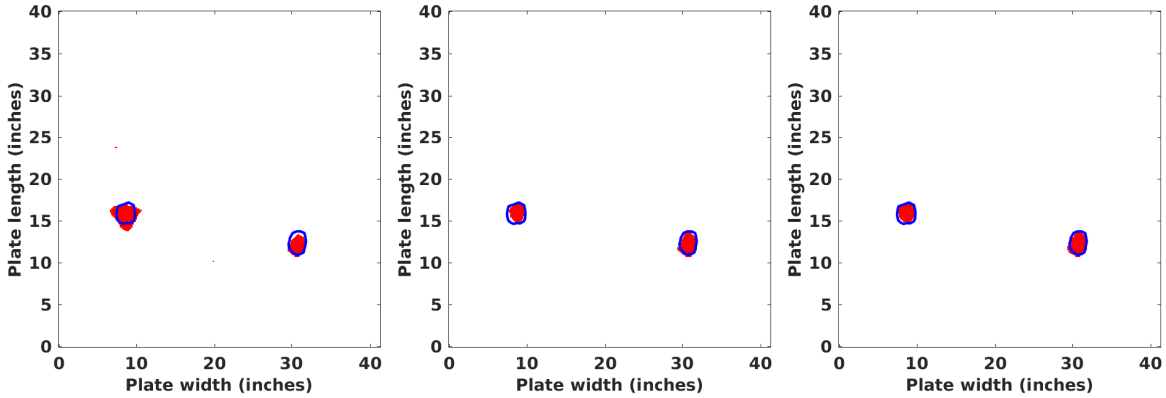
(c) Experiment 1: PC-SBL Sørenson-Dice index = 0.9402



(d) Experiment 2: Least-squares Sørenson-Dice index = 0.3427

(e) Experiment 2: LASSO Sørenson-Dice index = 0.9005

(f) Experiment 2: PC-SBL Sørenson-Dice index = 0.9377



(g) Experiment 3: Least-squares Sørenson-Dice index = 0.7491

(h) Experiment 3: LASSO Sørenson-Dice index = 0.8379

(i) Experiment 3: PC-SBL Sørenson-Dice index = 0.8729

Figure 7.2: Comparison of the damage outlines estimated by three different algorithms along with corresponding Sørenson-Dice similary index. The method of this chapter provides the best results out of the three methods.

algorithm compared to two competing algorithms available in the literature. The results showed that exploiting any underlying structure of the damage improves the map reconstruction accuracy. Hence, this chapter covered a problem which is not related to LDS, but connected the sparsity property of a linear system, which is the central theme of the thesis. Finally, in the next and the final chapter of the thesis, we summarize the main points and discuss some broader implications of the research presented.

Chapter 8

Conclusions

Summarizing the key takeaways and looking ahead

The thesis presented new theoretical results and algorithms concerning the estimation of state vectors in LDS with sparsity constraints. This final chapter summarizes all the findings presented so far, and the new insights the thesis has contributed. We also provide some exciting questions that the results raise and directions that seem to be promising for future work.

8.1 Summary of Contributions

We studied the sparse signal recovery problem under three different models associated with LDS. For each model, we investigated three important aspects: conditions for the existence of a solution, low-complexity recovery algorithm development, and recovery guarantees. We list the specific contributions associated with each model in the following subsections.

8.1.1 SM1: Known inputs and observation matrix

Under this model, we considered the recovery of a sparse initial state using the knowledge of measurements, inputs and other system matrices. We derived guarantees on recoverability of the sparse initial state of a linear dynamical system under a stochastic setting for two cases: (i) the observation matrices at different time instants are independent and identically distributed subgaussian random matrices; (ii) the observation matrices at all time instants are identical, and equal to a subgaussian random matrix. Our results revealed that when the system transfer matrix is arbitrary, the measurement bound for recovery depends on the inverse of the condition number of the matrix. Furthermore, our results are more general than existing results, and for the regime where they are comparable, our measurement bounds are tighter.

8.1.2 SM2: Unknown inputs and Known observation matrix

Under this model, we looked at the recovery of a set of sparse control inputs using the knowledge of the measurements and system transfer matrices. We first considered the conditions on the system for the existence of a solution. We developed a non-combinatorial, polynomial time test, called the PBH test, for determining the existence of a solution. Our procedure is equivalent to the existing Kalman rank based test, but it comes with the advantage of low complexity. It is interesting to note that such a non-combinational test is not available for a canonical sparse recovery problem, and the special structure in the measurement induced by the LDS made it possible to develop this simple test. We also derived bounds on the minimum number of input vectors required to ensure the existence of a solution, and an extension of the Kalman decomposition algorithm for sparse inputs.

In brief, the key contribution from this part of the work is the first-ever low-complexity controllability test for LDS with sparse inputs.

Next, we addressed the recovery of jointly sparse control inputs using the SBL framework. We developed a low-complexity, memory efficient algorithm that retained the good performance of SBL. Specifically, we presented a non-iterative online algorithm for recovering temporally correlated sparse vectors, which resulted in low computational complexity and memory requirements. We presented two schemes for implementation: a fixed lag scheme and a sawtooth lag scheme, and discussed an efficient method to initialize the algorithm. Further, we demonstrated the efficacy of the algorithm by applying it to the problem of OFDM wireless channel estimation.

Even though the proposed algorithm performed better than the existing algorithms in terms of recovery performance and run time, the key novelty of the work is in the analysis of the algorithm. To the best of our knowledge, none of the online algorithms for the sparse signal recovery come with theoretical guarantees. The offline counterpart of our algorithm, KM-SBL, is known to have theoretical guarantees only for the special cases when the sparse vectors are uncorrelated or perfectly correlated. We analyzed the proposed algorithms for these two special cases and established strong convergence guarantees. However, there is an important difference between the offline and online algorithm guarantees: offline algorithm analysis does not consider computational or memory limitations, and establishes recovery guarantees given a finite set of measurements; the online algorithm analysis accounts for the computational and memory limitations of the system, and establishes asymptotic guarantees describing limiting behavior as the number of measurements gets large. Therefore, although both type of algorithms have guarantees for

the same special cases, the flavor of results and the mathematical machinery used are completely different. In a few words, our algorithm stands out from the array of existing online sparse recovery algorithms owing to its strong theoretical guarantees.

8.1.3 SM3: Unknown inputs and observation matrix

Under this model, we considered the dictionary learning problem where the goal is to recover both the set of sparse control inputs and the observation matrix from the noisy measurements. We tackled the problem using the SBL framework by estimating the dictionary as a deterministic matrix with unit norm columns. Due to this, our algorithm outperforms existing Bayesian algorithms which use a prior on the dictionary elements, both in terms of the reconstruction accuracy and run time. The estimation method uses the expectation-maximization (EM) algorithm to simultaneously learn the parameters of the prior and the sparsifying dictionary. The dictionary update step in the EM algorithm is a quadratic optimization problem with unit norm constraints, which is a nonconvex problem because of the constraint. Since a closed form solution is not available, we proposed to employ the alternating minimization (AM) procedure or Armijo line search (ALS) to solve it. We illustrated the performance of the algorithms by comparing it with the other popular algorithms in the literature when applied to the image denoising problem.

Apart from the superior recovery performance, the main highlight of our dictionary learning algorithm is the associated theoretical guarantees. We showed that our formulation of the underlying cost function ensures that the algorithm converges to the sparsest possible representation. Further, we derived convergence guarantees of the dictionary update step using AM and ALS optimization procedures and established the stability of the limit points of the ALS procedure. Thus, the remarkable characteristic of the algorithm compared to

other dictionary algorithms is the powerful recovery guarantees.

8.1.4 Anomaly Imaging Exploiting Clustered Sparsity

We presented a new algorithm for anomaly imaging, based on algebraic reconstruction technique and the two-dimensional pattern coupled sparse Bayesian learning algorithm. The key features of this part of our work are as follows:

- *Exploiting Clustered Sparsity:* We exploit the two-dimensional clustered sparsity pattern of the anomaly map to improve damage mapping accuracy.

- *Experimental Validation:* Experimental results on a unidirectional composite plate show that our method provides more accurate estimates of the damage boundaries than two competing algorithms.

The major takeaway of this work is that exploiting any known structure in the map significantly improves the reconstruction accuracy of the anomaly map.

Overall, the thesis revolves around investigating the role of sparsity in linear systems. We looked at three different models of sparsity, and thoroughly examined some of the fundamental aspects related to sparse signal recovery in the context of LDS. The theoretical analysis presented here is deep-rooted in the rich and elegant mathematical theory of linear algebra, optimization, probability theory (in particular, concentration inequalities and random matrix theory), stochastic approximation, Riemannian matrix manifold, etc. The mathematical analysis presented in the thesis can lead to some interesting future research, and we discuss some ideas in the next section

8.2 Future Work

Using this thesis as a point of departure, one can consider new algorithm development for different but related sparsity models, and a deeper analysis of some of the problems, or explore theoretical aspects of the results out of simple curiosity. Some possible directions for future work are as follows:

1. **Stabilizability:** Similar to the analysis of observability and the controllability of LDS presented in Chapter 2 and Chapter 3, a similar theory on the stabilizability of an LDS under sparsity constraints can be developed. Further, it would be interesting to develop guarantees for the case where the measurement matrices are deterministic, possibly via the mutual coherence of the matrices.

2. **Constrained sparse-controllability:** Building upon our results in Chapter 3, further studies which impose a constraint on the maximum magnitude of the sparse inputs, or integer or lattice constraints on the sparse inputs, can be undertaken in the future.

3. **Online recovery algorithms:** Continuing the online algorithm development in Chapter 4, one can devise online algorithms where the measurements arrive sequentially, for the following scenarios:

   - Single sparse recovery problem ($\boldsymbol{D} = \boldsymbol{I}$).
   - Sparse input recovery in LDS for arbitrary $\boldsymbol{D}$ and $\boldsymbol{H}$.
   - Dictionary learning using linear projections of sparse data.

4. **Dictionary learning for LDS:** Following the ideas presented in Chapter 6, a

universal algorithm that can learn the observation matrix for a general LDS can be developed.

To conclude, the research presented in the thesis offered new mathematical theory and a bundle of algorithms connecting the areas of control theory, compressed sensing and online learning algorithms. We identified that sparse structures can arise in LDS in several practical scenarios. The results established that exploiting the sparsity along with any additional structure is intriguing and fascinating because of the beauty of its theoretical guarantees and the superior performance. Moreover, involving randomness in the measurement step enables one to establish strong theoretical guarantees. These realizations, together with their potential applications, have also triggered some important research questions that need to be looked at in the future.

Appendix A

Appendix to Chapter 2

A.1 Proof of Proposition 2.1

Proof. Using [32, Corollary 7.32] and [32, Theorem 7.30], we can show that, for $t > 0$,

$$\mathbb{P}\left\{\sum_{l=1}^{m}\left(A_l - \mathbb{E}\left\{A_l\right\}\right) \geq t\right\} \leq \exp\left(-\frac{c_2^2 t^2/2}{2c_1 m + c_2 t}\right) \tag{A.1}$$

$$\mathbb{P}\left\{\sum_{l=1}^{m}\left(-A_l + \mathbb{E}\left\{A_l\right\}\right) \geq t\right\} \leq \exp\left(-\frac{c_2^2 t^2/2}{2c_1 m + c_2 t}\right). \tag{A.2}$$

Therefore, for $t > m \max\left\{a_{\max}, -a_{\min}\right\}$,

$$\mathbb{P}\left\{\sum_{l=1}^{m} A_l \geq t\right\} \leq \exp\left(-\frac{c_2^2 (t - m a_{\max})^2/2}{2c_1 m + c_2 (t - m a_{\max})}\right)$$

$$\mathbb{P}\left\{\sum_{l=1}^{m} -A_l \geq t\right\} \leq \exp\left(-\frac{c_2^2 (t + m a_{\min})^2/2}{2c_1 m + c_2 (t + m a_{\min})}\right).$$

We get the desired result by combining the above inequalities using the union bound. □

A.2 Proof of Theorem 2.1

Proof. First, we note that an overall scaling does not affect the RIP of a matrix. Hence, without loss of generality, we assume that the largest and the smallest singular values of $\boldsymbol{D} \neq \boldsymbol{0}$ are 1 and λ, respectively. For any $\boldsymbol{z} \in \mathbb{R}^N$ such that $\|\boldsymbol{z}\|^2 = 1$ and $t \in (0,1)$, we have

$$\mathbb{P}\left\{\left|\frac{1}{Km}\left\|\tilde{\boldsymbol{A}}_{(K)}\boldsymbol{z}\right\|^2 - \|\boldsymbol{z}\|^2\right| \geq t\right\} = \mathbb{P}\left\{\left|\sum_{k=0}^{K-1}\sum_{l=1}^{m}\left(a_{k,l} + \left\|\boldsymbol{D}^k\boldsymbol{z}\right\|^2 - \|\boldsymbol{z}\|^2\right)\right| \geq Kmt\right\}, \quad \text{(A.3)}$$

where $a_{k,l} \triangleq \left|(\boldsymbol{A}_{(k)}^{\mathsf{T}})_l^{\mathsf{T}}\boldsymbol{D}^k\boldsymbol{z}\right|^2 - \left\|\boldsymbol{D}^k\boldsymbol{z}\right\|^2$, where $(\boldsymbol{A}_{(k)}^{\mathsf{T}})_l^{\mathsf{T}}$ is the l^{th} row of the matrix $\boldsymbol{A}_{(k)}$. Here, the term $(\boldsymbol{A}_{(k)}^{\mathsf{T}})_l^{\mathsf{T}}\boldsymbol{D}^k\boldsymbol{z}$ is the inner product between a row of $\tilde{\boldsymbol{A}}_{(K)}$ and $\boldsymbol{z}$. It is easy to see that $(\boldsymbol{A}_{(k)}^{\mathsf{T}})_l^{\mathsf{T}}\boldsymbol{D}^k\boldsymbol{z}$ is a subgaussian random variable with parameter $c\left\|\boldsymbol{D}^k\boldsymbol{z}\right\|^2$. Also, using the independence and unit variance property of the entries of $(\boldsymbol{A}_{(k)}^{\mathsf{T}})_l$, we have $\mathbb{E}\{a_{k,l}\} = 0$. Thus, from Lemma 2.1, for $|\theta| \leq \frac{1}{16c\left\|\boldsymbol{D}^k\boldsymbol{z}\right\|^2}$ and hence for $|\theta| \leq \frac{1}{16c}$, we have

$$\mathbb{E}\left\{\exp\left(\theta a_{k,l}\right)\right\} \leq \exp\left(128\theta^2c^2\left\|\boldsymbol{D}^k\boldsymbol{z}\right\|^4\right) \leq \exp\left(128\theta^2c^2\right),$$

which follows since the largest singular value of $\boldsymbol{D}$ is 1. Note that this holds true even if $\boldsymbol{D}$ is not invertible. Hence, using the Chernoff bound, for all $t > 0$,

$$\mathbb{P}\{|a_{k,l}| \geq t\} \leq 2\min_{0<\theta\leq\frac{1}{16c}}\exp\left(128\theta^2c^2\right)\exp\left(-\theta t\right) \quad \text{(A.4)}$$

$$\leq 2\exp\left(1/8\right)e^{-t/(32c)}, \quad \text{(A.5)}$$

where (A.5) is obtained by setting $\theta = 1/(32c)$. Further, independence of the rows of $\boldsymbol{A}_{(k)}$ for $k = 1, 2, \ldots, K$ implies that $a_{k,l}$ are independent. Therefore, $a_{k,l} + \left\|\boldsymbol{D}^k\boldsymbol{z}\right\|^2 - \|\boldsymbol{z}\|^2$

satisfies the conditions required to apply Proposition 2.1. Thus, (A.3), along with the fact $\lambda^{2(K-1)} - 1 \leq \left\|\boldsymbol{D}^k \boldsymbol{z}\right\|^2 - \|\boldsymbol{z}\|^2 \leq 0$ yields, for $t \in (1-\lambda^{2(K-1)}, 1)$

$$\begin{aligned}
&\mathbb{P}\left\{\left|\frac{1}{Km}\left\|\tilde{\boldsymbol{A}}_{(K)}\boldsymbol{z}\right\|^2 - \|\boldsymbol{z}\|^2\right| \geq t\right\} \\
&\quad\leq \exp\left(-\frac{c_2^2(Kmt)^2/2}{2c_1Km + c_2Kmt}\right) + \exp\left(-\frac{c_2^2\left[Kmt + Km\left(\lambda^{2(K-1)}-1\right)\right]^2/2}{2c_1Km + c_2Km\left(t+\lambda^{2(K-1)}-1\right)}\right) \quad \text{(A.6)} \\
&\quad\leq \exp\left(-\frac{c_2^2 Kmt^2}{2\left(2c_1 + c_2 t\right)}\right) + \exp\left(-\frac{c_2^2 Km\left(t+\lambda^{2(K-1)}-1\right)^2}{2\left(2c_1+c_2t\right)}\right) \quad \text{(A.7)} \\
&\quad\leq 2\exp\left(-\tilde{c}Km\left(t-1+\lambda^{2(K-1)}\right)^2\right), \quad \text{(A.8)}
\end{aligned}$$

where $c_1 = 2\exp(1/8)$, $c_2 = 1/(32c)$ and $\tilde{c} = \frac{c_2^2}{2(2c_1+c_2)}$. Also, the last step follows because when $t \in (1-\lambda^{2(K-1)}, 1)$, we have $t^2 \geq \left(t+\lambda^{2(K-1)}-1\right)^2$. Now, using the proof technique in [32, Theorem 9.11], we get that if (2.13) holds, the RIC δ_s of A satisfies $\delta_s < \delta$, for all $\delta > 1-\lambda^{2(K-1)}$, with probability at least $1-\epsilon$. This completes the proof. □

A.3 Proof of Theorem 2.3

Before we prove the theorem, we present a set of mathematical tools used in the proof.

A.3.1 Toolbox

Let $\mathcal{Z} \subset \mathbb{R}^{m\times N}$ be a set of matrices, and the set $\mathcal{T}_s$ denote the set of $s-$sparse vectors in $\mathbb{R}^N$: $\mathcal{T}_s = \left\{\boldsymbol{z} \in \mathbb{R}^N : \|\boldsymbol{z}\| = 1 \text{ and } \|\boldsymbol{z}\|_0 \leq s\right\}$. We need the following two definitions to state the results in this subsection.

Definition A.1 (Admissible sequence)**.** *An admissible sequence $\mathcal{U} = \{\mathcal{U}_i\}_{i=0}^{\infty}$ on $\mathcal{Z}$ is an increasing sequence of partitions of $\mathcal{Z}$ such that $|\mathcal{U}_i| = 2^{2^i}$ and $|\mathcal{U}_0| = 1$. Here, increasing sequence of partitions implies that every set of $\mathcal{U}_i$ is contained in one of the sets of $\mathcal{U}_{i-1}$*

for all i. Also, $\mathcal{U}_0 = \mathcal{Z}$, and every set of $\mathcal{U}_i$ is a subset of $\mathcal{Z}$. Given a matrix $\boldsymbol{Z} \in \mathcal{Z}$, we denote the unique set of $\mathcal{U}_i$ that contains $\boldsymbol{Z}$ by $\mathcal{U}_i(\boldsymbol{Z})$.

Definition A.2. *[Functionals on a set of matrices] We define three functionals on $\mathcal{Z}$ as follows:*

$$d_F(\mathcal{Z}) \triangleq \sup_{\boldsymbol{Z}\in\mathcal{Z}} \|\boldsymbol{Z}\|_F \tag{A.9}$$

$$d_2(\mathcal{Z}) \triangleq \sup_{\boldsymbol{Z}\in\mathcal{Z}} \|\boldsymbol{Z}\|_2 \tag{A.10}$$

$$\zeta(\mathcal{Z}) \triangleq \inf_{\mathcal{U}=\{\mathcal{U}_i\}_{i=0}^{\infty}} \sup_{\boldsymbol{Z}\in\mathcal{Z}} \sum_{i=0}^{\infty} 2^{i/2}\mathcal{D}(\mathcal{U}_i(\boldsymbol{Z})), \tag{A.11}$$

where the inf *is over all possible admissible sequences, and the term $\mathcal{D}$ is defined as follows:*

$$\mathcal{D}(\mathcal{U}_i(\boldsymbol{Z})) \triangleq \max_{\boldsymbol{U}_{(1)},\boldsymbol{U}_{(2)}\in\mathcal{U}_i(\boldsymbol{Z})} \left\|\boldsymbol{U}_{(1)} - \boldsymbol{U}_{(2)}\right\|. \tag{A.12}$$

It represents the diameter the set $\mathcal{U}_i(\boldsymbol{Z})$, which is a decreasing function of i.

Next, we state a result which is the main ingredient of our proof. It bounds the suprema of a chaos process indexed by the set $\mathcal{Z}$.

Theorem A.1. *[72, Theorem 3.1] Let $\boldsymbol{u}$ be a random vector whose entries are independent zero-mean, unit-variance subgaussian random variables with common parameter c. Let*

$$F_1 \triangleq \zeta(\mathcal{Z})\left[\zeta(\mathcal{Z}) + d_F(\mathcal{Z})\right] + d_F(\mathcal{Z})\, d_2(\mathcal{Z}) \tag{A.13}$$

$$F_2 \triangleq d_2^2(\mathcal{Z})\left[\zeta(\mathcal{Z}) + d_F(\mathcal{Z})\right]^2 \tag{A.14}$$

$$F_3 \triangleq d_2^2(\mathcal{Z}). \tag{A.15}$$

Then, for $t > 0$, it holds that

$$\mathbb{P}\left\{\sup_{\boldsymbol{Z}\in\mathcal{Z}}\left|\|\boldsymbol{Z}\boldsymbol{u}\|^2 - \mathbb{E}\left\{\|\boldsymbol{Z}\boldsymbol{u}\|^2\right\}\right| > c_1F_1 + t\right\} \leq 2\exp\left(-c_2\min\left\{\frac{t^2}{F_2}, \frac{t}{F_3}\right\}\right), \tag{A.16}$$

where $c_1, c_2 > 0$ are universal positive constants which depend only on c.

It is difficult to directly apply the above theorem due to the complicated form of the functional $\zeta(\mathcal{Z})$. We need a result that bounds the function $\zeta(\mathcal{Z})$ using the notion of the covering number. The covering number is defined as follows.

Definition A.3 (Covering number)**.** *Given $u > 0$, the covering number $\mathsf{CN}\{\mathcal{Z}, u\}$ is defined as the smallest integer p such that one can find a subset $\mathcal{Z}' \subset \mathcal{Z}$ satisfying $|\mathcal{Z}'| \leq p$ and*

$$\min_{\boldsymbol{Z}'\in\mathcal{Z}'}\|\boldsymbol{Z} - \boldsymbol{Z}'\|_2 \leq u, \forall \boldsymbol{Z} \in \mathcal{Z}. \tag{A.17}$$

Lemma A.1. *[186]. For the functional $\zeta(\mathcal{Z})$, it holds that*

$$\zeta(\mathcal{Z}) \leq \int_0^\infty \ln^{1/2}\left(\mathsf{CN}\{\mathcal{Z}, u\}\right) du. \tag{A.18}$$

The covering number is hard to compute in closed form. Hence, we use the following lemma to further bound the covering number of the set of interest to us.

Lemma A.2. *[74, Lemma 6] Let a linear map $\boldsymbol{A} : \mathbb{R}^N \to \mathbb{R}^m$ be such that*

$$\left\|\frac{1}{\sqrt{s}}\boldsymbol{A}(\boldsymbol{z})\right\| \leq \kappa\|\boldsymbol{z}\|_1, \forall \boldsymbol{z} \in \mathbb{R}^N, \kappa > 0. \tag{A.19}$$

Then, for the set $\mathcal{T}_s$, it holds that

$$\ln\left(\mathsf{CN}\left\{\boldsymbol{A}(\mathcal{T}_s), u\right\}\right) \quad \begin{cases} \leq Cs\min\left\{\frac{\kappa^2}{u^2}\ln^2 N, s\ln N + s\ln\left(1+\frac{2\kappa}{u}\right)\right\} \\ \qquad\qquad\qquad\qquad \textit{for } 0 < u < \kappa \\ = 0 \qquad\qquad\qquad\qquad \textit{for } u > \kappa, \end{cases} \tag{A.20}$$

where C is a universal positive constant.

We will also need the following result from elementary calculus in the proof to tackle the integral in Lemma A.1.

Lemma A.3. *For $a \leq b$, it holds that*

$$\int_0^a \sqrt{\ln\left(1+\frac{b}{u}\right)}du \leq \frac{3}{2}a\sqrt{\ln\left(1+\frac{b}{a}\right)}. \tag{A.21}$$

Proof. See Appendix A.4. □

We have now presented all the mathematical tools that are required to prove the theorem. In the next subsection, we formally prove the desired result.

A.3.2 Proof

As mentioned in Section 2.2, (2.6) is equivalent to (2.4). Therefore, without loss of generality, we assume that the largest and the smallest singular values of $\boldsymbol{D}$ are 1 and λ, respectively. We recall that our goal is to obtain a probabilistic bound on $\left\|\widetilde{\boldsymbol{A}}_{(K)}\boldsymbol{z}\right\|$ for $\boldsymbol{z} \in \mathcal{T}_s$, using Theorem A.1. At a high level, there are four main steps to the proof:

- First, we convert $\widetilde{\boldsymbol{A}}_{(K)}\boldsymbol{z}$ to the form given in Theorem A.1, i.e., the product of a matrix and a subgaussian vector.

- Second, we bound the three functionals d_2, d_F and ζ in Definition A.2.

- Third, using the bounds in the previous step, we bound F_1, F_2 and F_3 in Theorem A.1, since the three quantities are functions of d_2, d_F and ζ.

- Fourth, we apply Theorem A.1 with the upper bounds on F_1, F_2 and F_3. Here, we note that Theorem A.1 holds for upper bounds on F_1, F_2 and F_3. This yields a concentration inequality bounding the deviation of the random variable $\left\| \frac{1}{\sqrt{Km}} \widetilde{\boldsymbol{A}}_{(K)} \boldsymbol{z} \right\|^2$ from its mean $\frac{1}{K} \sum_{k=0}^{K-1} \left\| \boldsymbol{D}^k \boldsymbol{z} \right\|^2$. Finally, we establish the desired result by suitable algebraic manipulation of the concentration inequality.

In the remainder of this section, we provide the details of each of these steps.

For the first step, we consider the following:

$$\left\| \frac{1}{\sqrt{Km}} \widetilde{\boldsymbol{A}}_{(K)} \boldsymbol{z} \right\|^2 = \sum_{k=0}^{K-1} \left\| \frac{1}{\sqrt{Km}} \boldsymbol{A} \boldsymbol{D}^k \boldsymbol{z} \right\|^2 \tag{A.22}$$

$$= \left\| \frac{1}{\sqrt{Km}} \boldsymbol{A} \boldsymbol{Z}_{(K)}(\boldsymbol{z}) \right\|_F^2, \tag{A.23}$$

where the matrix function $\boldsymbol{Z}_{(K)} : \mathcal{T}_s \to \mathbb{R}^{N \times K}$ is defined as follows:

$$\boldsymbol{Z}_{(K)}(\boldsymbol{z}) \triangleq \left[\boldsymbol{z} \quad \boldsymbol{D}\boldsymbol{z} \quad \ldots \boldsymbol{D}^{K-1} \boldsymbol{z} \right], \boldsymbol{z} \in \mathcal{T}_s. \tag{A.24}$$

Further, we have

$$\left\| \frac{1}{\sqrt{Km}} \widetilde{\boldsymbol{A}}_{(K)} \boldsymbol{z} \right\|^2 = \left\| \frac{1}{\sqrt{Km}} \boldsymbol{Z}_{(K)}^{\mathsf{T}}(\boldsymbol{z}) \boldsymbol{A}^{\mathsf{T}} \right\|_F^2 \tag{A.25}$$

$$= \left\| \widetilde{\boldsymbol{Z}}_{(K)}(\boldsymbol{z}) \mathsf{vec}\left\{ \boldsymbol{A}^{\mathsf{T}} \right\} \right\|^2, \tag{A.26}$$

where $\mathsf{vec}\left\{ \boldsymbol{A}^{\mathsf{T}} \right\} \in \mathbb{R}^{mN}$ is the vectorized version of the matrix $\boldsymbol{A}^{\mathsf{T}}$, which has subgaussian entries with common parameter c. The matrix function $\widetilde{\boldsymbol{Z}}_{(K)} : \mathcal{T}_s \to \mathbb{R}^{Km \times mN}$ is a block diagonal matrix with $\frac{1}{\sqrt{Km}} \boldsymbol{Z}_{(K)}^{\mathsf{T}}(\boldsymbol{z})$ as the m^{th} block diagonal entries, for all $\boldsymbol{z} \in \mathcal{T}_s$. Thus,

the first step is complete.

The next step is bound the three terms d_2, d_F and ζ using the following lemmas.

Lemma A.4. *For the set* $\widetilde{\mathcal{Z}}_{(K)}$,

$$d_F\left(\widetilde{\mathcal{Z}}_{(K)}\right) \leq 1 \tag{A.27}$$

$$d_2\left(\widetilde{\mathcal{Z}}_{(K)}\right) \leq \sqrt{\frac{s}{Km}} L\left(\boldsymbol{D}, K\right), \tag{A.28}$$

where $L\left(\boldsymbol{D}, K\right)$ *is as defined in* (2.32).

Proof. See Appendix A.5. □

Lemma A.5. *The functional* $\zeta\left(\widetilde{\mathcal{Z}}_{(K)}\right)$ *can be bounded as follows:*

$$\zeta\left(\widetilde{\mathcal{Z}}_{(K)}\right) \leq C\sqrt{\frac{s}{Km}} L\left(\boldsymbol{D}, K\right) \ln N \ln s, \tag{A.29}$$

for some C *which is universal positive constant that depends only on the subgaussian parameter* c.

Proof. See Appendix A.6. □

Now, we combine the results in the second step to obtain bounds on F_1, F_2, and F_3. Further, we note that we need to bound $\mathbb{P}\left\{\left|\left\|\frac{1}{\sqrt{Km}}\widetilde{\boldsymbol{A}}_{(K)}\boldsymbol{z}\right\|^2 - \frac{1}{K}\sum_{k=0}^{K-1}\left\|\boldsymbol{D}^k\boldsymbol{z}\right\|^2\right| > \widetilde{\delta}\right\}$, for some $0 \leq \tilde{\delta} < 1$ using Theorem A.1. To this end, we use the assumptions of Theorem 2.3 to further upper bound F_1 to make it a multiple of $\tilde{\delta}$. We summarize the third step in the following lemma.

Lemma A.6. *Using the definitions in Theorem A.1 and under the assumptions of Theorem 2.3, there exists a constant $C_1 > 0$ such that*

$$F_1 \leq \tilde{\delta}/2c_1 \tag{A.30}$$

$$F_2 \leq C_1 \frac{s}{Km} L^2(\boldsymbol{D}, K) \tag{A.31}$$

$$F_3 \leq \frac{s}{Km} L^2(\boldsymbol{D}, K), \tag{A.32}$$

where c_1 is the same constant as in Theorem A.1 and

$$\widetilde{\delta} \triangleq \delta - 1 + \lambda^{2(K-1)}. \tag{A.33}$$

Proof. See Appendix A.7. □

Now, we are ready to go the final step of the proof. We apply Theorem A.1 to (A.26) with $t = \widetilde{\delta}/2$ to get

$$\mathbb{P}\left\{\left|\left\|\frac{1}{\sqrt{Km}}\widetilde{\boldsymbol{A}}_{(K)}\boldsymbol{z}\right\|^2 - \frac{1}{K}\sum_{k=0}^{K-1}\left\|\boldsymbol{D}^k\boldsymbol{z}\right\|^2\right| > \widetilde{\delta}\right\} \leq 2\exp\left(-C\min\left\{\widetilde{\delta}^2, \widetilde{\delta}\right\}\frac{Km}{sL^2(\boldsymbol{D}, K)}\right) \tag{A.34}$$

$$\leq 2\exp\left(-C\frac{Km\tilde{\delta}^2}{sL^2(\boldsymbol{D}, K)}\right) \leq \epsilon, \tag{A.35}$$

where the universal positive constant C depends on the subgaussian parameter c, and we use (2.33) of Theorem 2.3 to bound using ϵ in the last step.

Thus, for all $\boldsymbol{z} \in \mathbb{R}^N$ such that $\|\boldsymbol{z}\| = 1$ and $\|\boldsymbol{z}\|_0 \leq s$, with probability at least $1 - \epsilon$,

$$\left|\left\|\frac{1}{\sqrt{Km}}\widetilde{\boldsymbol{A}}_{(K)}\boldsymbol{z}\right\|^2 - \frac{1}{K}\sum_{k=0}^{K-1}\left\|\boldsymbol{D}^k\boldsymbol{z}\right\|^2\right| \leq \widetilde{\delta}. \tag{A.36}$$

Therefore, for $0 \leq \tilde{\delta} < \lambda^{2(K-1)}$,

$$\lambda^{2(K-1)} - \widetilde{\delta} < \frac{1}{Km} \left\| \widetilde{\boldsymbol{A}}_{(K)} \boldsymbol{z} \right\|^2 < 1 + \widetilde{\delta}, \tag{A.37}$$

since $\lambda^{K-1} \leq \lambda^k \leq \left\| \boldsymbol{D}^k \boldsymbol{z} \right\| \leq 1$. We also use (A.33) to relate δ and $\tilde{\delta}$ as follows:

$$1 - \delta < \frac{1}{Km} \left\| \widetilde{\boldsymbol{A}}_{(K)} \boldsymbol{z} \right\|^2 < 1 + \delta, \tag{A.38}$$

for $\delta > 1 - \lambda^{2(K-1)}$, with probability at least $1 - \epsilon$. Hence, $\frac{1}{Km} \widetilde{\boldsymbol{A}}_{(K)}$ satisfies RIP of order s with RIC as δ, with probability at least $1 - \epsilon$. Thus, the proof is complete.

A.4 Proof of Lemma A.3

We have

$$\int_0^a \sqrt{\ln\left(1 + \frac{b}{u}\right)} du = b \int_{\sqrt{\ln(1+b/a)}}^{\infty} t d\left(\frac{1}{\exp(t^2) - 1}\right) \tag{A.39}$$

$$= a\sqrt{\ln\left(1 + \frac{b}{a}\right)} + b \int_{\sqrt{\ln(1+b/a)}}^{\infty} \frac{1}{\exp(t^2) - 1} dt, \tag{A.40}$$

where we use the substitution $t = \sqrt{\ln\left(1 + \frac{b}{u}\right)}$ in (A.39) and integration by parts to get (A.40). Now, the second term simplifies as follows:

$$\int_{\sqrt{\ln(1+b/a)}}^{\infty} \frac{1}{\exp(t^2) - 1} dt \leq \int_{\sqrt{\ln(1+b/a)}}^{\infty} \frac{t}{\sqrt{\ln(1 + b/a)}} \frac{e^{-t^2}}{1 - e^{-t^2}} dt \tag{A.41}$$

$$= \frac{1}{2\sqrt{\ln(1 + b/a)}} \ln\left(1 + \frac{a}{b}\right). \tag{A.42}$$

Therefore, we get

$$\int_0^a \sqrt{\ln\left(1+\frac{b}{u}\right)}du \leq a\sqrt{\ln\left(1+\frac{b}{a}\right)} + \frac{b}{2\sqrt{\ln(1+b/a)}}\ln\left(1+\frac{a}{b}\right) \tag{A.43}$$

$$= a\sqrt{\ln\left(1+\frac{b}{a}\right)}\left[1+\frac{b}{2a}\left(1-\frac{\ln\left(\frac{b}{a}\right)}{\ln(1+b/a)}\right)\right]. \tag{A.44}$$

Now, we need to show that $\frac{b}{a}\left(1-\frac{\ln\left(\frac{b}{a}\right)}{\ln(1+b/a)}\right) \leq 1$ to complete the proof. So, we consider the function $h(u) \triangleq u\left(1-\frac{\ln u}{\ln(u+1)}\right)$, by replacing $b/a = u \geq 1$. Further, we note that $h(1) = 1$, and therefore it suffices to show that $\frac{d}{du}h(u) \leq 0$, which then implies that $h(u) \leq h(1) = 1$, for all $u \geq 1$. We have

$$\frac{d}{du}h(u) = 1 - \frac{\ln u}{\ln(u+1)} - \frac{(u+1)\ln(u+1) - u\ln u}{(u+1)\ln^2(u+1)} \tag{A.45}$$

$$= \frac{\widetilde{h}(u)}{(u+1)\ln^2(u+1)}, \tag{A.46}$$

where we define

$$\widetilde{h}(u) \triangleq (u+1)\ln^2(u+1) - (u+1)\ln(u+1)\ln u - (u+1)\ln(u+1) + u\ln u. \tag{A.47}$$

Now, $\frac{d}{du}h(u) \leq 0$ if $\tilde{h}(u) \leq 0$. Therefore, we show that $\frac{d}{du}\widetilde{h}(u) \leq 0$, which implies that $\tilde{h}(u) \leq \tilde{h}(1) = 2\ln^2 2 - 2\ln 2 < 0$. Then, we get

$$\frac{d}{du}\widetilde{h}(u) = \ln(u+1)\left(\ln(u+1) - \ln u - \frac{u+1}{u}\right) \tag{A.48}$$

$$= -\ln(u+1)\left(\ln u + 1 - \ln(u+1) + \frac{1}{u}\right). \tag{A.49}$$

Using the same technique again, we now consider the function $\ln u + 1 - \ln(u+1)$. Since derivative of $\ln u + 1 - \ln(u+1)$ is $\frac{1}{u(u+1)} > 0$, for $u \geq 1$, $\ln u + 1 - \ln(u+1) \geq 1 - \ln 2 > 0$.

Therefore, $\frac{d}{du}\widetilde{h}(u) \leq 0$ because $\ln u + 1 - \ln(u+1) \geq 0$ and $\frac{1}{u} \geq 0$, for $u \geq 1$. Hence, we get $\widetilde{h}(u) < 0$. This implies that $\frac{d}{du}h(u) < 0$, and thus, $h(u) \leq h(1) = 1$, for $u \geq 1$. Substituting this in (A.44) completes the proof.

A.5 Proof of Lemma A.4

To show the first part of the lemma, we have

$$d_F\left(\widetilde{\mathcal{Z}}_{(K)}\right) = \sup_{\boldsymbol{z}\in\mathcal{T}_s} \left\|\widetilde{\boldsymbol{Z}}_{(K)}(\boldsymbol{z})\right\|_F \tag{A.50}$$

$$= \frac{1}{\sqrt{K}} \sup_{\boldsymbol{z}\in\mathcal{T}_s} \left\|\boldsymbol{Z}_{(K)}(\boldsymbol{z})\right\|_F \tag{A.51}$$

$$= \sup_{\boldsymbol{z}\in\mathcal{T}_s} \frac{1}{\sqrt{K}} \sqrt{\sum_{k=0}^{K-1} \left\|\boldsymbol{D}^k\boldsymbol{z}\right\|^2} \leq 1, \tag{A.52}$$

where the last step follows from the definition of $\mathcal{T}_s$, and the fact that the largest singular value of $\boldsymbol{D}$ is unity.

To show the second part of the lemma, we have

$$d_2\left(\widetilde{\mathcal{Z}}_{(K)}\right) = \sup_{\boldsymbol{z}\in\mathcal{T}_s} \left\|\widetilde{\boldsymbol{Z}}_{(K)}(\boldsymbol{z})\right\|_2 = \frac{1}{\sqrt{Km}} \sup_{\boldsymbol{z}\in\mathcal{T}_s} \left\|\boldsymbol{Z}_{(K)}(\boldsymbol{z})\right\|_2 = \frac{1}{\sqrt{Km}} \sup_{\boldsymbol{z}\in\mathcal{T}_s} \left\|\sum_{i=1}^{N} \widetilde{\boldsymbol{D}}_{(K,i)}\boldsymbol{z}_i\right\|_2 \tag{A.53}$$

$$\leq \frac{1}{\sqrt{Km}} \sup_{\boldsymbol{z}\in\mathcal{T}_s} \sum_{i=1}^{N} |\boldsymbol{z}_i| \left\|\widetilde{\boldsymbol{D}}_{(K,i)}\right\|_2 \leq \frac{L(\boldsymbol{D},K)}{\sqrt{Km}} \sup_{\boldsymbol{z}\in\mathcal{T}_s} \|\boldsymbol{z}\|_1 \tag{A.54}$$

$$\leq L(\boldsymbol{D},K)\sqrt{\frac{s}{Km}}\,\|\boldsymbol{z}\| = \sqrt{\frac{s}{Km}}L(\boldsymbol{D},K)\,. \tag{A.55}$$

where (A.53) and (A.54) follow from the definitions of $\widetilde{\boldsymbol{D}}_{(K,i)}$ and $L(\boldsymbol{D},K)$ in (2.31) and (2.32), respectively. Also, (A.55) is because $\boldsymbol{z}$ is at most $s-$sparse. Hence, the proof is complete.

A.6 Proof of Lemma A.5

From Lemma A.4, for all $\boldsymbol{Z} \in \widetilde{\mathcal{Z}}_{(K)}$ and any $\boldsymbol{z} \in \mathbb{R}^{mN}$,

$$\left\| \frac{1}{\sqrt{s}} \boldsymbol{Z}\boldsymbol{z} \right\| \leq \sqrt{\frac{1}{Km}} L\left(\boldsymbol{D}, K\right) \|\boldsymbol{z}\| \leq \sqrt{\frac{1}{Km}} L\left(\boldsymbol{D}, K\right) \|\boldsymbol{z}\|_1 . \tag{A.56}$$

Then, from Lemma A.1, we have

$$\begin{aligned}
\frac{1}{\sqrt{s}} \zeta\left(\widetilde{\mathcal{Z}}_{(K)}\right) &\leq \frac{1}{\sqrt{s}} \int_0^{\infty} \ln^{1/2}\left[\mathsf{CN}\left\{\widetilde{\mathcal{Z}}_{(K)}, u\right\}\right] du & \text{(A.57)} \\
&= \frac{1}{\sqrt{s}} \int_0^{\frac{L(\boldsymbol{D},K)}{\sqrt{sKm}}} \ln^{1/2}\left[\mathsf{CN}\left\{\widetilde{\mathcal{Z}}_{(K)}, u\right\}\right] du \\
&\qquad + \frac{1}{\sqrt{s}} \int_{\frac{L(\boldsymbol{D},K)}{\sqrt{sKm}}}^{\frac{L(\boldsymbol{D},K)}{\sqrt{Km}}} \ln^{1/2}\left[\mathsf{CN}\left\{\widetilde{\mathcal{Z}}_{(K),u}\right\}\right] du, & \text{(A.58)}
\end{aligned}$$

Further, using Lemma A.2 with $\kappa = \sqrt{\frac{1}{Km}} L\left(\boldsymbol{D}, K\right)$, for some positive constant C', we have

$$\begin{aligned}
\frac{1}{\sqrt{s}} \zeta\left(\widetilde{\mathcal{Z}}_{(K)}\right) &\leq C' \int_0^{\frac{L(\boldsymbol{D},K)}{\sqrt{sKm}}} \sqrt{s \ln N + s \ln\left(1 + \frac{2L\left(\boldsymbol{D}, K\right)}{u\sqrt{Km}}\right)} du \\
&\qquad + C' \int_{\frac{L(\boldsymbol{D},K)}{\sqrt{sKm}}}^{\frac{L(\boldsymbol{D},K)}{\sqrt{Km}}} \frac{L\left(\boldsymbol{D}, K\right)}{u\sqrt{Km}} \ln N du & \text{(A.59)} \\
&\leq C' \int_0^{\frac{L(\boldsymbol{D},K)}{\sqrt{sKm}}} \sqrt{s \ln N} + \sqrt{s \ln\left(1 + \frac{2L\left(\boldsymbol{D}, K\right)}{u\sqrt{Km}}\right)} du \\
&\qquad + C' \frac{L\left(\boldsymbol{D}, K\right)}{\sqrt{Km}} \ln N \ln \sqrt{s} & \text{(A.60)} \\
&\leq C' \frac{L\left(\boldsymbol{D}, K\right)}{\sqrt{Km}} \left(\sqrt{\ln N} + 3/2 \sqrt{\ln\left(1 + 2\sqrt{s}\right)} + \ln N \ln \sqrt{s}\right) & \text{(A.61)} \\
&\leq C \frac{L\left(\boldsymbol{D}, K\right)}{\sqrt{Km}} \ln N \ln s, & \text{(A.62)}
\end{aligned}$$

where $C = 3C'$. Also, (A.60) uses the fact that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, for any $a, b > 0$, and (A.61) uses Lemma A.3. Thus, the proof is complete.

A.7 Proof of Lemma A.6

From Lemma A.4 and Lemma A.5, we get

$$F_1 = \zeta\left(\widetilde{\mathcal{Z}}_{(K)}\right)\left[\zeta\left(\widetilde{\mathcal{Z}}_{(K)}\right) + d_F\left(\widetilde{\mathcal{Z}}_{(K)}\right)\right] + d_F\left(\widetilde{\mathcal{Z}}_{(K)}\right) d_2\left(\widetilde{\mathcal{Z}}_{(K)}\right) \tag{A.63}$$

$$\leq CL\left(\boldsymbol{D}, K\right)\sqrt{\frac{s}{Km}}\ln N \ln s\left(CL\left(\boldsymbol{D}, K\right)\sqrt{\frac{s}{Km}}\ln N \ln s + 1\right) + L\left(\boldsymbol{D}, K\right)\sqrt{\frac{s}{Km}}\frac{\ln N \ln s}{\ln^2 2}, \tag{A.64}$$

where we use the bound $\frac{\ln N \ln s}{\ln^2 2} > 1$ when $N \geq s > 1$ to get (A.64). Next, we use assumption (2.33) in Theorem 2.3, i.e.,

$$L\left(\boldsymbol{D}, K\right)\sqrt{\frac{s}{Km}}\ln N \ln s \leq \sqrt{\widetilde{c}\widetilde{\delta}}, \tag{A.65}$$

to get

$$F_1 \leq C\sqrt{\widetilde{c}\widetilde{\delta}}\left(C\sqrt{\widetilde{c}\widetilde{\delta}} + 1\right) + \frac{\sqrt{\widetilde{c}\widetilde{\delta}}}{\ln^2 2} \tag{A.66}$$

$$\leq \sqrt{\widetilde{c}\widetilde{\delta}}\left(C^2\sqrt{\widetilde{c}} + C + \frac{1}{\ln^2 2}\right). \tag{A.67}$$

The last step above follows because of the bound $\widetilde{\delta} \leq 1$. Finally, we choose $\widetilde{c}$ in (2.33) of Theorem 2.3 such that

$$\sqrt{\widetilde{c}}\left(C^2\sqrt{\widetilde{c}} + C + \frac{1}{\ln^2 2}\right) \leq \frac{1}{2c_1}, \tag{A.68}$$

where c_1 and C are the same constants as in Theorem A.1 and Lemma A.5, respectively. We note that $\sqrt{\widetilde{c}}\left(C^2\sqrt{\widetilde{c}} + C + \frac{1}{\ln^2 2}\right)$ is strictly increasing with $\widetilde{c}$, for $\widetilde{c} \geq 0$, and the left hand side equals zero when $\widetilde{c} = 0$. Thus, there exists a $\widetilde{c} > 0$ that satisfies (A.68), for any c_1 and C. Thus, from (A.67) and (A.68) we get $F_1 \leq \frac{\tilde{\delta}}{2c_1}$, and thus, we complete of the

first part of the proof.

Similarly, we simplify F_2 using Lemma A.4, Lemma A.5 and (A.65) as follows:

$$F_2 = d_2^2\left(\widetilde{\mathcal{Z}}_{(K)}\right)\left[\zeta\left(\widetilde{\mathcal{Z}}_{(K)}\right) + d_F\left(\widetilde{\mathcal{Z}}_{(K)}\right)\right]^2 \tag{A.69}$$

$$\leq \frac{s}{Km}L^2\left(\boldsymbol{D},K\right)\left(CL\left(\boldsymbol{D},K\right)\sqrt{\frac{s}{Km}}\ln N\ln s+1\right)^2 \tag{A.70}$$

$$\leq \frac{s}{Km}L^2\left(\boldsymbol{D},K\right)\left(C\sqrt{\widetilde{c}}\widetilde{\delta}+1\right)^2 \leq C_1\frac{s}{Km}L^2\left(\boldsymbol{D},K\right), \tag{A.71}$$

where we use the fact that $\widetilde{\delta}<1$ and define $C_1 \triangleq (C\sqrt{\widetilde{c}}+1)^2$. Finally, we have,

$$F_3 = d_2^2\left(\widetilde{\mathcal{Z}}_{(K)}\right) \leq \frac{s}{Km}L^2\left(\boldsymbol{D},K\right), \tag{A.72}$$

which completes the proof.

A.8 Proof of Proposition 2.2

To prove the result, we first upper and lower bound the term $L(\boldsymbol{D},K)$. We have,

$$L(\boldsymbol{D},K) \leq \max_i \left\|\widetilde{\boldsymbol{D}}_{(K,i)}\right\|_F \tag{A.73}$$

$$= \max_i \sqrt{\sum_{k=0}^{K-1}\left\|\boldsymbol{D}_i^k\right\|^2} \leq \sqrt{K}, \tag{A.74}$$

where we obtain the last step from the fact that the largest singular value of $\boldsymbol{D}^k$ is at most unity, and Rayleigh-Ritz theorem [187, Theorem 4.2.2] which gives

$$1 = \sup_{\boldsymbol{z}\in\mathbb{R}^N,\boldsymbol{z}\neq 0}\frac{\left\|\boldsymbol{z}^{\mathsf{T}}\boldsymbol{D}^{k\mathsf{T}}\boldsymbol{D}^k\boldsymbol{z}\right\|}{\left\|\boldsymbol{z}\right\|^2} \geq \max_i\left(\boldsymbol{D}^{k\mathsf{T}}\boldsymbol{D}^k\right)_{i,i} = \max_i\left\|\boldsymbol{D}_i^k\right\|^2. \tag{A.75}$$

Similarly, we also have,

$$L(\boldsymbol{D},K)^2 = \max_{i=1,2,\ldots,N}\left[\sup_{\boldsymbol{z}\in\mathbb{R}^K,\boldsymbol{z}\neq 0}\frac{\left\|\boldsymbol{z}^\mathsf{T}\widetilde{\boldsymbol{D}}_{(K,i)}^\mathsf{T}\widetilde{\boldsymbol{D}}_{(K,i)}\boldsymbol{z}\right\|}{\|\boldsymbol{z}\|^2}\right] \tag{A.76}$$

$$\geq \max_{\substack{i=1,2,\ldots,N\\ k=0,1,\ldots,K-1}}\left(\widetilde{\boldsymbol{D}}_{(K,i)}^\mathsf{T}\widetilde{\boldsymbol{D}}_{(K,i)}\right)_{k,k} \tag{A.77}$$

$$= \max_i\left[\max_{k=0,1,\ldots,K-1}\left\|\boldsymbol{D}_i^k\right\|^2\right] = 1, \tag{A.78}$$

where (A.78) uses the fact that $\left\|\boldsymbol{D}_i^0\right\|^2 = \|\boldsymbol{I}_i\|^2 = 1$ and $\left\|\boldsymbol{D}_i^k\right\|^2 \leq 1$, for $k = 1, 2, \ldots, K-1$ from (A.75). Combining (A.74) and (A.78), we obtain the desired result.

A.9 Proof of Proposition 2.3

To prove the monotonicity of the function, we need to show that

$$L^2(\boldsymbol{D},K+1) \leq L^2(\boldsymbol{D},K)\left(1+\frac{1}{K}\right). \tag{A.79}$$

Therefore, we consider the following relation connecting $L^2(\boldsymbol{D},K+1)$ and $L^2(\boldsymbol{D},K)$:

$$L^2(\boldsymbol{D},K+1) = \max_i\left\|\widetilde{\boldsymbol{D}}_{(K+1,i)}\right\|_2 = \max_i\sup_{\substack{\boldsymbol{z}\in\mathbb{R}^N\\ \|\boldsymbol{z}\|=1}}\boldsymbol{z}^\mathsf{T}\widetilde{\boldsymbol{D}}_{(K+1,i)}\widetilde{\boldsymbol{D}}_{(K+1,i)}^\mathsf{T}\boldsymbol{z} \tag{A.80}$$

$$= \max_i\sup_{\substack{\boldsymbol{z}\in\mathbb{R}^N\\ \|\boldsymbol{z}\|=1}}\sum_{j=0}^{K}\left|\boldsymbol{z}^\mathsf{T}\boldsymbol{D}^j\boldsymbol{I}_i\right|^2 \tag{A.81}$$

$$\leq \max_i\sup_{\substack{\boldsymbol{z}\in\mathbb{R}^N\\ \|\boldsymbol{z}\|=1}}\sum_{j=0}^{K-1}\left|\boldsymbol{z}^\mathsf{T}\boldsymbol{D}^j\boldsymbol{I}_i\right|^2 + \max_i\sup_{\substack{\boldsymbol{z}\in\mathbb{R}^N\\ \|\boldsymbol{z}\|=1}}\left|\boldsymbol{z}^\mathsf{T}\boldsymbol{D}^K\boldsymbol{I}_i\right|^2 \tag{A.82}$$

$$\leq L^2(\boldsymbol{D},K) + \max_i\left\|\boldsymbol{D}^K\boldsymbol{I}_i\right\|^2, \tag{A.83}$$

where we use the definition of $L^2(\boldsymbol{D}, K)$ and Cauchy-Schwarz inequality to get the last step. Now, to complete the proof, it suffices to show that

$$\max_i \left\|\boldsymbol{D}^K \boldsymbol{I}_i\right\|^2 \leq L^2(\boldsymbol{D}, K)/K. \tag{A.84}$$

Since $\boldsymbol{D}$ is a psd matrix with largest singular value as unity, $\boldsymbol{D}^j - \boldsymbol{D}^k$ is a psd matrix, for any pair of integers $j \leq k$. Then, we have

$$K^2 \left\|\boldsymbol{D}^K \boldsymbol{I}_i\right\|^2 = K^2 \boldsymbol{I}_i^\mathsf{T} \boldsymbol{D}^{2K} \boldsymbol{I}_i \leq \sum_{j=0}^{K-1} \sum_{k=0}^{K-1} \boldsymbol{I}_i^\mathsf{T} \boldsymbol{D}^{j+k} \boldsymbol{I}_i \tag{A.85}$$

$$= \sum_{j=0}^{K-1} \sum_{k=0}^{K-1} \boldsymbol{I}_i^\mathsf{T} \boldsymbol{D}^{j\mathsf{T}} \boldsymbol{D}^k \boldsymbol{I}_i \tag{A.86}$$

$$= K \left(\mathbf{1}/\sqrt{K}\right)^\mathsf{T} \widetilde{\boldsymbol{D}}_{(K+1,i)}^\mathsf{T} \widetilde{\boldsymbol{D}}_{(K+1,i)} \left(\mathbf{1}/\sqrt{K}\right) \leq K L^2(\boldsymbol{D}, K). \tag{A.87}$$

Hence, (A.84) holds, which in turn shows (A.79). Thus, the proof is complete.

Appendix B

Appendix to Chapter 3

B.1 Proof of Theorem 3.1

Proof. The proof is by showing that the conditions of the theorem are equivalent to the Kalman-type rank test. The proof relies on the fact that the Kalman rank test for the unconstrained system is equivalent to the PBH test. We note that the PBH test is same as condition 1 of Theorem 3.1 [90].

We first prove that conditions of Theorem 3.1 imply the Kalman-type rank test. Suppose that the Kalman-type rank test fails. Then, we consider the following matrix of size $N \times N\tilde{K}s$:

$$\begin{aligned}\tilde{\boldsymbol{H}}^* = [\boldsymbol{D}^{\tilde{K}N-1}\boldsymbol{H}_{\mathcal{S}_1} \quad \boldsymbol{D}^{\tilde{K}N-2}\boldsymbol{H}_{\mathcal{S}_1} \quad \dots \quad \boldsymbol{D}^{(\tilde{K}-1)N}\boldsymbol{H}_{\mathcal{S}_1} \\ \dots \quad \boldsymbol{D}^{(\tilde{K}-1)N-1}\boldsymbol{H}_{\mathcal{S}_2} \quad \dots \boldsymbol{D}^{(\tilde{K}-2)N}\boldsymbol{H}_{\mathcal{S}_2} \quad \dots \\ \dots \quad \boldsymbol{D}^{N-1}\boldsymbol{H}_{\mathcal{S}_{\tilde{K}}} \quad \dots \quad \boldsymbol{H}_{\mathcal{S}_{\tilde{K}}}], \quad \text{(B.1)}\end{aligned}$$

where we define $\tilde{K} \triangleq \lceil L/s \rceil$ index sets as follows:

$$|\mathcal{S}_i| = s, \quad \cup_{i=1}^{\tilde{K}} \mathcal{S}_i = \{1, 2, \dots, L\}\,. \quad \text{(B.2)}$$

Since the Kalman-type rank test fails, $\tilde{\boldsymbol{H}}^*$ does not have full row rank for any finite K. Further, we can rearrange the columns of $\tilde{\boldsymbol{H}}^*$ to get the following matrix which has the same rank as that of $\tilde{\boldsymbol{H}}^*$: $\begin{bmatrix}\boldsymbol{D}^{N-1}\boldsymbol{H}^* & \boldsymbol{D}^{N-2}\boldsymbol{H}^* & \dots & \boldsymbol{H}^*\end{bmatrix}$, where $\boldsymbol{H}^* \in \mathbb{R}^{N\times \tilde{K}s}$ is defined as follows:

$$\boldsymbol{H}^* \triangleq \begin{bmatrix}\boldsymbol{D}^{(\tilde{K}-1)N}\boldsymbol{H}_{\mathcal{S}_1} & \boldsymbol{D}^{(\tilde{K}-2)N}\boldsymbol{H}_{\mathcal{S}_2}\dots & \boldsymbol{H}_{\mathcal{S}_{\tilde{K}}}\end{bmatrix}. \tag{B.3}$$

Thus, using the Kalman rank test for unconstrained inputs, the system with transfer matrix $\boldsymbol{D}$ and input matrix $\boldsymbol{H}^*$ is not controllable. Then, the PBH test for unconstrained inputs implies that the matrix $\begin{bmatrix}\boldsymbol{D}-\lambda\boldsymbol{I} & \boldsymbol{H}^*\end{bmatrix} \in \mathbb{R}^{N\times N+\tilde{K}s}$ has rank less than N, for some $\lambda \in \mathbb{C}$. Therefore, there exists a nonzero vector $\boldsymbol{z} \in \mathbb{R}^N$ such that $\boldsymbol{z}^\mathsf{T}\boldsymbol{D} = \lambda\boldsymbol{z}^\mathsf{T}$ and $\boldsymbol{z}^\mathsf{T}\boldsymbol{H}^* = \boldsymbol{0}$. However, we have

$$\boldsymbol{0} = \boldsymbol{z}^\mathsf{T}\boldsymbol{H}^* = \boldsymbol{z}^\mathsf{T}\begin{bmatrix}\lambda^{(\tilde{K}-1)N}\boldsymbol{H}_{\mathcal{S}_1} & \lambda^{(\tilde{K}-2)N}\boldsymbol{H}_{\mathcal{S}_2}\dots & \boldsymbol{H}_{\mathcal{S}_{\tilde{K}}}\end{bmatrix}. \tag{B.4}$$

So either $\lambda = 0$ and $\boldsymbol{z}^\mathsf{T}\boldsymbol{H}_{\mathcal{S}_{\tilde{K}}} = \boldsymbol{0}$, or, if λ is nonzero, then $\boldsymbol{z}^\mathsf{T}\boldsymbol{H} = \boldsymbol{0}$ because $\boldsymbol{z}$ is orthogonal to all columns of $\boldsymbol{H}$ due to (B.2). Repeating the same arguments for all possible index sets $\{\mathcal{S}_i\}$ satisfying (B.2), we get that for every index set $\mathcal{S}_i$ with s entries, there exists a vector $\boldsymbol{z} \in \mathbb{R}^N$ such that $\boldsymbol{z}^\mathsf{T}\boldsymbol{D} = \lambda\boldsymbol{z}^\mathsf{T}$, and one of the following conditions hold: $\lambda = 0$ and $\boldsymbol{z}^\mathsf{T}\boldsymbol{H}_{\mathcal{S}_i} = \boldsymbol{0}$, or $\boldsymbol{z}^\mathsf{T}\boldsymbol{H} = \boldsymbol{0}$. Therefore, we get that one of the following cases hold:

1. There exists a left eigenvector $\boldsymbol{z}$ of $\boldsymbol{D}$, such that $\boldsymbol{z}^\mathsf{T}\boldsymbol{H} = \boldsymbol{0}$. In this case, condition 1 of Theorem 3.1 does not hold.

2. For every index set $\mathcal{S}$ with s entries, there exists a nonzero vector $\boldsymbol{z} \in \mathbb{R}^N$ such that $\boldsymbol{z}^\mathsf{T}\boldsymbol{D} = \boldsymbol{0}$, and $\boldsymbol{z}^\mathsf{T}\boldsymbol{H}_{\mathcal{S}} = \boldsymbol{0}$. This implies that $\boldsymbol{z}^\mathsf{T}\begin{bmatrix}\boldsymbol{D} & \boldsymbol{H}_{\mathcal{S}}\end{bmatrix} = \boldsymbol{0}$. Therefore, rank of $\begin{bmatrix}\boldsymbol{D} & \boldsymbol{H}_{\mathcal{S}}\end{bmatrix}$ is less than N, for every index set $\mathcal{S}$. Thus, condition 2 of Theorem 3.1

does not hold.

Thus, when the Kalman-type rank test is unsuccessful, the conditions of the theorem are also violated.

Next, we prove that the Kalman-type rank test implies the conditions of the theorem. Suppose that the two conditions do not hold simultaneously. This could happen under the following two exhaustive cases:

1. Suppose that condition 1 does not hold. Then, the PBH test is violated, and the system is not controllable. When a system is not controllable, it cannot be sparse-controllable.

2. Suppose condition 2 does not hold. Then, for every index set $\mathcal{S}$ with s entries, there exists a nonzero vector $\boldsymbol{z}$ such that $\boldsymbol{z}^{\mathsf{T}}\boldsymbol{H}_{\mathcal{S}} = \mathbf{0}$ and $\boldsymbol{z}^{\mathsf{T}}\boldsymbol{D} = \mathbf{0}$. This implies that for any set of K index sets $\{\mathcal{S}_i : |\mathcal{S}_i| = s\}_{i=1}^{K}$ there exists a nonzero vector $\boldsymbol{z} \in \mathbb{R}^N$ such that

$$\boldsymbol{z}^{\mathsf{T}} \begin{bmatrix} \boldsymbol{D}^{K-1}\boldsymbol{H}_{\mathcal{S}_1} & \boldsymbol{D}^{K-2}\boldsymbol{H}_{\mathcal{S}_2} & \dots & \boldsymbol{H}_{\mathcal{S}_K} \end{bmatrix} = \mathbf{0}. \tag{B.5}$$

   Hence, the Kalman-type rank test fails.

Thus, the proof is complete. □

B.2 Proof of Corollary 3.2

Proof. We first note that if a system is $s-$sparse-controllable, it is controllable. Hence, we need to prove that if a system with an invertible state transition matrix $\boldsymbol{D}$ is controllable, it is $s-$sparse-controllable, for any positive integer $s \leq L$.

If the system is controllable, the PBH test for unconstrained input succeeds, which implies that the condition 1 of Theorem 3.1 holds. Further, if $\boldsymbol{D}$ is invertible, the matrix $\begin{bmatrix}\boldsymbol{D} & \boldsymbol{H}_{\mathcal{S}}\end{bmatrix}$ has rank N for any $\boldsymbol{H}$ and index set $\mathcal{S}$. Therefore, condition 2 of Theorem 3.1 always holds, for any $s \leq L$. Hence, from Theorem 3.1, the system is $s-$sparse-controllable. Thus, the desired result is proved. □

B.3 Proof of Theorem 3.3

Using the Kalman-type rank test, the minimum number of input vectors required to ensure controllability is the smallest integer K that satisfies rank condition of the test. So, for any finite K, we define $\mathcal{H}_{(K)} \subseteq \mathbb{R}^{N \times Ks}$ as the set of submatrices of $\tilde{\boldsymbol{H}}_{(K)}$ of the following form (as defined for the Kalman-type rank test):

$$\begin{bmatrix}\boldsymbol{D}^{K-1}\boldsymbol{H}_{\mathcal{S}_1} & \boldsymbol{D}^{K-2}\boldsymbol{H}_{\mathcal{S}_2} & \dots & \boldsymbol{H}_{\mathcal{S}_K}\end{bmatrix}. \tag{B.6}$$

Here, the index set $\mathcal{S}_i$ has s entries, for $i = 1, 2, \dots, K$. Also, we define the following:

$$R^*_{(K)} = \max_{\boldsymbol{H}_{(K)} \in \mathcal{H}_{(K)}} \mathsf{Rank}\left\{\boldsymbol{H}_{(K)}\right\}. \tag{B.7}$$

$$\mathcal{H}^*_{(K)} = \left\{\boldsymbol{H}_{(K)} \in \mathcal{H}_{(K)} : \mathsf{Rank}\left\{\boldsymbol{H}_{(K)}\right\} = R^*_{(K)}\right\} \tag{B.8}$$

With these definitions, K^* is the smallest integer such that $R^*_{(K^*)} = N$.

Before starting the proof, we outline the main steps involved. At a high level, there are five steps to the proof:

1. We begin by showing that for any matrix $\boldsymbol{H}_{(K)} \in \mathcal{H}_{(K)}$, we can find a matrix

$\boldsymbol{H}^*_{(K)} \in \mathcal{H}^*_{(K)}$ such that

$$\mathcal{CS}\left\{\boldsymbol{H}_{(K)}\right\} \subseteq \mathcal{CS}\left\{\boldsymbol{H}^*_{(K)}\right\}. \tag{B.9}$$

2. Second, using the above claim, we show that if K is any integer such that

$$R^*_{(K)} = R^*_{(K+1)}, \tag{B.10}$$

then $R^*_{(K+Q)} = R^*_{(K)}$, for any positive integer Q.

3. Third, we prove that K^* is the smallest integer K such that (B.10) holds, which in turn leads to the upper bound: $K^* \leq N + 1 - R^*_{\boldsymbol{H},s}$, where $R^*_{\boldsymbol{H},s}$ is as defined in the statement of the theorem.

4. Fourth, we show that in order to satisfy the rank criterion in (B.10), $\boldsymbol{H}^*_{(K^*)}$ needs to contain at most $qR_{\boldsymbol{H}}$ number of columns with a particular structure. Then, we provide a choice of index sets $\{\mathcal{S}_i\}_{i=1}^{K=q\lceil R_{\boldsymbol{H}}/s\rceil}$ which can lead to that particular structure. Since the smallest integer K that can achieve rank criterion in (B.10) is K^*, we assert that $K^* \leq q\lceil R_{\boldsymbol{H}}/s\rceil$. Thus, together with the above step, we establish the upper bound in the theorem.

5. Finally, we lower bound K^* to complete the proof.

B.3.1 Characterizing $\mathcal{H}^*_{(K)}$

If $\boldsymbol{H}_{(K)} \in \mathcal{H}^*_{(K)}$, the result is trivial: $\boldsymbol{H}^*_{(K)} = \boldsymbol{H}_{(K)}$. Suppose that $\boldsymbol{H}_{(K)} \notin \mathcal{H}^*_{(K)}$, then $\mathsf{Rank}\left\{\boldsymbol{H}_{(K)}\right\} < R^*_{(K)}$. Therefore, to find $\boldsymbol{H}^*_{(K)}$, we have to replace some linearly dependent columns of $\boldsymbol{H}_{(K)}$ with columns which are linearly independent of the rest of the columns of $\boldsymbol{H}_{(K)}$, as follows:

1. Find a set $\{\boldsymbol{u}_i\}_{i=1}^{\mathsf{Rank}\{\boldsymbol{H}_{(K)}\}}$ of columns of $\boldsymbol{H}_{(K)}$ that are linearly independent and span $\mathcal{CS}\left\{\boldsymbol{H}_{(K)}\right\}$.

2. Since $\boldsymbol{H}_{(K)}$ is a submatrix of $\tilde{\boldsymbol{H}}_{(K)}$, we can extend the set $\{\boldsymbol{u}_i\}_{i=1}^{\mathsf{Rank}\{\boldsymbol{H}_{(K)}\}}$ to form a basis $\{\boldsymbol{u}_i\}_{i=1}^{\mathsf{Rank}\{\tilde{\boldsymbol{H}}_{(K)}\}}$ of $\mathcal{CS}\left\{\tilde{\boldsymbol{H}}_{(K)}\right\}$ by adding columns from $\tilde{\boldsymbol{H}}_{(K)}$. We note that $\boldsymbol{u}_i = \boldsymbol{D}^p\boldsymbol{H}_j$ for some integers p and j because of the structure of $\tilde{\boldsymbol{H}}_{(K)}$.

3. Replace the linearly dependent columns of $\boldsymbol{H}_{(K)}$ with the columns from the set $\{\boldsymbol{u}_i\}_{i=\mathsf{Rank}\{\boldsymbol{H}_{(K)}\}+1}^{\mathsf{Rank}\{\tilde{\boldsymbol{H}}_{(K)}\}}$ to get a new matrix $\bar{\boldsymbol{H}}_{(K)} \in \mathbb{R}^{N\times Ks}$.We only replace a column of form $\boldsymbol{D}^p\boldsymbol{H}_j$ in $\boldsymbol{H}_{(K)}$ with another column of the form $\boldsymbol{D}^p\boldsymbol{H}_{j'}$, for all p and j and some integer j'. This ensures that $\bar{\boldsymbol{H}}_{(K)} \in \mathcal{H}_{(K)}$. In this fashion, we replace as many columns of $\boldsymbol{H}_{(K)}$ as necessary to ensure that $\bar{\boldsymbol{H}}_{(K)}$ has the maximum rank, $R^*_{(K)}$. However, since we are only replacing linearly dependent columns, we have

$$\mathcal{CS}\left\{\boldsymbol{H}_{(K)}\right\} \subseteq \mathcal{CS}\left\{\bar{\boldsymbol{H}}_{(K)}\right\}. \tag{B.11}$$

Since $\mathsf{Rank}\left\{\bar{\boldsymbol{H}}_{(K)}\right\} = R^*_{(K)}$ and $\bar{\boldsymbol{H}}_{(K)} \in \mathcal{H}_{(K)}$, we get that $\bar{\boldsymbol{H}}_{(K)} \in \mathcal{H}^*_{(K)}$, satisfying (B.11). Hence, the first step of the proof is complete.

B.3.2 Characterizing $R^*_{(K)}$

We use the proof by induction technique to show that $R^*_{(K+Q)} = R^*_{(K)}$, for any integer $Q > 0$. Hence, it suffices to show the following:

$$R^*_{(K+2)} = R^*_{(K+1)}. \tag{B.12}$$

From (B.7), we know that $R^*_{(K+2)} \geq R^*_{(K+1)}$. Also,

$$R^*_{(K)} = \max_{\boldsymbol{H}_{(K)} \in \mathcal{H}_{(K)}} \dim \left\{ \mathcal{CS}\left\{ \boldsymbol{H}_{(K)} \right\} \right\}, \tag{B.13}$$

where $\dim\{\cdot\}$ denotes the dimension of a subspace. Thus, we establish (B.12) by showing that for any matrix $\boldsymbol{H}_{(K+2)} \in \mathcal{H}_{(K+2)}$, there exists a matrix $\boldsymbol{H}^*_{(K+1)} \in \mathcal{H}^*_{(K+1)}$ such that

$$\mathcal{CS}\left\{ \boldsymbol{H}_{(K+2)} \right\} \subseteq \mathcal{CS}\left\{ \boldsymbol{H}^*_{(K+1)} \right\}. \tag{B.14}$$

We prove this relation by separately looking at the column spaces spanned by the first s columns and the last $(K+1)s$ columns of $\boldsymbol{H}_{(K+2)}$. We know that the submatrix formed by the last $(K+1)s$ columns of any matrix in $\mathcal{H}_{(K+2)}$ belongs to $\mathcal{H}_{(K+1)}$. Thus, using the claim in the first step, we can find a matrix $\boldsymbol{H}^*_{(K+1)}$ such that the column space spanned by the last $(K+1)s$ columns of $\boldsymbol{H}_{(K)}$ is contained in $\mathcal{CS}\left\{ \boldsymbol{H}^*_{(K+1)} \right\}$. Therefore, it suffices to show that the column space spanned by the first s columns of $\boldsymbol{H}_{(K+2)}$ is contained in the column space of the same matrix $\boldsymbol{H}^*_{(K+1)}$.

To prove the above statement, we note that the column space of the first s columns of $\boldsymbol{H}_{(K+2)}$ is contained in $\mathcal{CS}\left\{ \boldsymbol{D}^{K+1}\boldsymbol{H} \right\}$. Also, $\mathcal{CS}\left\{ \boldsymbol{H}^*_{(K+1)} \right\}$ contains the intersection $\bigcap_{\boldsymbol{H}^*_{(K+1)} \in \mathcal{H}^*_{(K+1)}} \mathcal{CS}\left\{ \boldsymbol{H}^*_{(K+1)} \right\}$. Hence, it suffices to show that

$$\mathcal{CS}\left\{ \boldsymbol{D}^{K+1}\boldsymbol{H} \right\} \subseteq \bigcap_{\boldsymbol{H}^*_{(K+1)} \in \mathcal{H}^*_{(K+1)}} \mathcal{CS}\left\{ \boldsymbol{H}^*_{(K+1)} \right\}, \tag{B.15}$$

which we prove using the relation (B.10).

To show that (B.15) holds, we consider an index set $\mathcal{S} \subseteq \{1, 2, \ldots, L\}$ with s entries and a matrix $\boldsymbol{H}^*_{(K)} \in \mathcal{H}^*_{(K)}$. Now, the matrix $\begin{bmatrix} \boldsymbol{D}^K \boldsymbol{H}_{\mathcal{S}} & \boldsymbol{H}^*_{(K)} \end{bmatrix} \in \mathbb{R}^{N \times (K+1)s}$ belongs to

$\mathcal{H}_{(K+1)}$. Thus, from (B.7) and (B.10) we have

$$\mathsf{Rank}\left\{\begin{bmatrix}\boldsymbol{D}^K\boldsymbol{H}_{\mathcal{S}} & \boldsymbol{H}^*_{(K)}\end{bmatrix}\right\} \leq R^*_{(K+1)} = R^*_{(K)}. \tag{B.16}$$

However, we also have

$$\mathsf{Rank}\left\{\begin{bmatrix}\boldsymbol{D}^K\boldsymbol{H}_{\mathcal{S}} & \boldsymbol{H}^*_{(K)}\end{bmatrix}\right\} \geq \mathsf{Rank}\left\{\boldsymbol{H}^*_{(K)}\right\} = R^*_{(K)}. \tag{B.17}$$

Thus, for all index sets $\mathcal{S}$ with s entries and any matrix $\boldsymbol{H}^*_{(K)} \in \mathcal{H}^*_{(K)}$,

$$\mathsf{Rank}\left\{\begin{bmatrix}\boldsymbol{D}^K\boldsymbol{H}_{\mathcal{S}} & \boldsymbol{H}^*_{(K)}\end{bmatrix}\right\} = \mathsf{Rank}\left\{\boldsymbol{H}^*_{(K)}\right\} \tag{B.18}$$

This relation immediately implies the following:

$$\mathsf{Rank}\left\{\begin{bmatrix}\boldsymbol{D}^K\boldsymbol{H} & \boldsymbol{H}^*_{(K)}\end{bmatrix}\right\} = \mathsf{Rank}\left\{\boldsymbol{H}^*_{(K)}\right\}, \tag{B.19}$$

for any matrix $\boldsymbol{H}^*_{(K)} \in \mathcal{H}^*_{(K)}$. Thus, we get that the columns of $\boldsymbol{D}^K\boldsymbol{H}$ belong to $\mathcal{CS}\left\{\boldsymbol{H}^*_{(K)}\right\}$, for any matrix $\boldsymbol{H}^*_{(K)} \in \mathcal{H}^*_{(K)}$. Hence,

$$\mathcal{CS}\left\{\boldsymbol{D}^K\boldsymbol{H}\right\} \subseteq \bigcap_{\boldsymbol{H}^*_{(K)}\in\mathcal{H}^*_{(K)}} \mathcal{CS}\left\{\boldsymbol{H}^*_{(K)}\right\}. \tag{B.20}$$

Therefore, we get

$$\mathcal{CS}\left\{\boldsymbol{D}^{K+1}\boldsymbol{H}\right\} \subseteq \bigcap_{\boldsymbol{H}^*_{(K)}\in\mathcal{H}^*_{(K)}} \mathcal{CS}\left\{\boldsymbol{D}\boldsymbol{H}^*_{(K)}\right\}. \tag{B.21}$$

Hence, to prove (B.15), we need to show that

$$\bigcap_{\boldsymbol{H}^*_{(K)}\in\mathcal{H}^*_{(K)}} \mathcal{CS}\left\{\boldsymbol{D}\boldsymbol{H}^*_{(K)}\right\} \subseteq \bigcap_{\boldsymbol{H}^*_{(K+1)}\in\mathcal{H}^*_{(K+1)}} \mathcal{CS}\left\{\boldsymbol{H}^*_{(K+1)}\right\}. \tag{B.22}$$

We prove the above relation by showing that there exists a matrix $\boldsymbol{H}^*_{(K+1)} \in \mathcal{H}^*_{(K+1)}$

such that

$$\mathcal{CS}\left\{\boldsymbol{D}\boldsymbol{H}^*_{(K)}\right\} \subseteq \mathcal{CS}\left\{\boldsymbol{H}^*_{(K+1)}\right\}, \tag{B.23}$$

for every matrix $\boldsymbol{H}^*_{(K)} \in \mathcal{H}^*_{(K)}$. So we consider a new matrix $\bar{\boldsymbol{H}}_{(K+1)} \in \mathbb{R}^{N\times(K+1)s}$ as follows:

$$\bar{\boldsymbol{H}}_{(K+1)} \triangleq \left[\boldsymbol{D}\boldsymbol{H}^*_{(K)} \quad \boldsymbol{H}_{\mathcal{S}}\right], \tag{B.24}$$

for some index set $\mathcal{S} \subseteq \{1, 2, \ldots, L\}$ and $|\mathcal{S}| = s$. Since $\bar{\boldsymbol{H}}_{(K+1)} \in \mathcal{H}_{(K+1)}$, using the arguments in the first step, we can find a matrix $\boldsymbol{H}^*_{(K+1)} \in \mathcal{H}^*_{(K+1)}$ such that

$$\mathcal{CS}\left\{\bar{\boldsymbol{H}}_{(K+1)}\right\} \subseteq \mathcal{CS}\left\{\boldsymbol{H}^*_{(K+1)}\right\}. \tag{B.25}$$

However, (B.24) implies that

$$\mathcal{CS}\left\{\boldsymbol{D}\boldsymbol{H}^*_{(K)}\right\} \subseteq \mathcal{CS}\left\{\bar{\boldsymbol{H}}_{(K+1)}\right\}. \tag{B.26}$$

Therefore, (B.23) holds, and hence (B.22) is proved.

Recall that (B.22) implies (B.15), which in turn establishes the relation (B.12). By mathematical induction, we conclude that $\mathsf{Rank}\left\{\boldsymbol{H}^*_{(K+Q)}\right\} = \mathsf{Rank}\left\{\boldsymbol{H}^*_{(K)}\right\}$, for any positive integer Q, and the proof of the second step in the outline is complete.

B.3.3 First part of the upper bound

Suppose that K_* is the smallest integer such that $R^*_{(K_*)} = R^*_{(K_*+1)}$. From (B.7), it is clear that

$$R^*_{(K)} \leq R^*_{(K+1)} \leq N, \tag{B.27}$$

for any positive integer K. Since $R^*_{(K^*)} = N$, we have $R^*_{(K^*)} = R^*_{(K^*+1)} = N$. Therefore, $K_* \leq K^*$, and $R^*_{(K_*)} = N$ from the claim in the second step.

Further, we know that K^* is the smallest integer such that $R^*_{(K^*)} = N$. Therefore, $K_* = K^*$. Hence, we conclude that $R^*_{(K)}$ strictly increases with K, for $1 \leq K \leq K^*$. Therefore, we have

$$N = R^*_{(K^*)} \geq R^*_{(K^*-1)} + 1 \geq R^*_{(K^*-2)} + 2 \tag{B.28}$$

$$\geq R^*_{(1)} + K^* - 1 \tag{B.29}$$

$$= R^*_{\boldsymbol{H},s} + K^* - 1. \tag{B.30}$$

Hence, the third step in the outline is complete.

B.3.4 Upper bounding K^*

To prove that $K^* \leq q\lceil R_{\boldsymbol{H}}/s \rceil$, we first look at the linearly independent columns in $\boldsymbol{H}^*_{(K^*)}$. We note that for any K, each column of $\boldsymbol{H}^*_{(K)}$ is of the form $\boldsymbol{D}^p\boldsymbol{H}_j$, for some integer p, and $j \in \{1, 2, \ldots, L\}$. However, since q is the degree of the minimal polynomial of $\boldsymbol{D}$, then for any integer Q, $\boldsymbol{D}^p$ can be expressed as a linear combination of $\left\{\boldsymbol{D}^i\right\}_{i=Q}^{Q+q-1}$, for all $p \geq Q$. Therefore, for any j, if $\left\{\boldsymbol{D}^i\boldsymbol{H}_j \in \mathbb{R}^N\right\}_{i=Q}^{Q+q-1}$ are any q columns of $\boldsymbol{H}^*_{(K)}$, further adding columns of the form $\boldsymbol{D}^p\boldsymbol{H}_j$, for $p \geq Q$ does not improve the rank of the matrix. Therefore, for a given j, at most q columns of the form $\boldsymbol{D}^p\boldsymbol{H}_j$ need to be present in $\boldsymbol{H}^*_{(K)}$ to ensure the rank criterion in (B.10).

Further, let $\boldsymbol{H}_{\mathcal{S}'}$ represents the set of $R_{\boldsymbol{H}}$ linearly independent columns of $\boldsymbol{H}$, i.e., $\mathcal{S}' \subseteq \{1, 2, \ldots, L\}$ and $|\mathcal{S}'| = R_{\boldsymbol{H}}$. Then, for any integer p, if $\left\{\boldsymbol{D}^p\boldsymbol{H}_j \in \mathbb{R}^N\right\}_{j\in\mathcal{S}'}$ are any $R_{\boldsymbol{H}}$ columns of $\boldsymbol{H}^*_{(K)}$, further adding columns of the form $\boldsymbol{D}^p\boldsymbol{H}_j$, for $j \notin \mathcal{S}'$ does not improve

the rank of the matrix. Therefore, for any given p, at most $R_{\boldsymbol{H}}$ columns of the form $\boldsymbol{D}^p\boldsymbol{H}_j$ need to be present in $\boldsymbol{H}^*_{(K)}$ to ensure the rank criterion.

In short, we have proved that, in order to ensure the rank criterion in (B.10), $\boldsymbol{H}^*_{(K)}$ needs to have at most q columns of the form $\boldsymbol{D}^p\boldsymbol{H}_j$, for any given j, and at most $R_{\boldsymbol{H}}$ columns of the form $\boldsymbol{D}^p\boldsymbol{H}_j$, for any given p. Hence, $\boldsymbol{H}^*_{(K)}$ needs to have at most $qR_{\boldsymbol{H}}$ columns to satisfy the rank criterion in (B.10).

We complete the proof for the upper bound by providing a choice of index sets for each input vector, that satisfies the above conditions. We form index sets $\{\mathcal{S}'_i\}_{i=1}^{K=\lceil R_{\boldsymbol{H}}/s\rceil}$ that partition the set of $R_{\boldsymbol{H}}$ linearly independent columns into groups of size at most s. The index sets are selected such that $\cup_{j=1}^{K}\mathcal{S}'_i = \mathcal{S}'$, $|\mathcal{S}_i| = s$, and $\mathcal{S}_K$ is such that $\begin{bmatrix}\boldsymbol{D} & \boldsymbol{H}_{\mathcal{S}_K}\end{bmatrix}$ has rank N. The existence of such an index set $\mathcal{S}_K$ is ensured by condition 2 of Theorem 3.1. Also, we note that due to the condition $|\mathcal{S}_i| = s$, the index sets need not be disjoint. Next, we choose $\mathcal{S}_i = \mathcal{S}'_j$, for $i = (j-1)q+1, (j-1)q+2, \ldots, jq$. Hence, we get the following $N \times qKs$ submatrix of $\tilde{\boldsymbol{H}}_{(K)} \in \mathbb{R}^{N\times qKL}$:

$$\begin{aligned}\boldsymbol{H}^*_{(K)} = [&\boldsymbol{D}^{Kq-1}\boldsymbol{H}_{\mathcal{S}_1} \quad \boldsymbol{D}^{Kq-2}\boldsymbol{H}_{\mathcal{S}_1} \quad \ldots \quad \boldsymbol{D}^{(K-1)q}\boldsymbol{H}_{\mathcal{S}_1} \\ &\ldots \quad \boldsymbol{D}^{(K-1)q-1}\boldsymbol{H}_{\mathcal{S}_2} \quad \ldots \boldsymbol{D}^{(K-2)q}\boldsymbol{H}_{\mathcal{S}_2} \quad \ldots \\ &\ldots \quad \boldsymbol{D}^{q-1}\boldsymbol{H}_{\mathcal{S}_K} \quad \ldots \quad \boldsymbol{H}_{\mathcal{S}_K}]. \quad \text{(B.31)}\end{aligned}$$

It is easy to see that this choice of index sets ensures that for any given p, $R_{\boldsymbol{H}}$ columns of the form $\boldsymbol{D}^p\boldsymbol{H}_j$ are present in $\boldsymbol{H}^*_{(K)}$. Also, for any given $j \in \mathcal{S}'$, q columns of $\left\{\boldsymbol{D}^i\boldsymbol{H}_j \in \mathbb{R}^N\right\}_{i=Q}^{Q+q-1}$ are present in $\boldsymbol{H}^*_{(K)}$. Hence, $K^* \leq q\lceil R_{\boldsymbol{H}}/s\rceil$, which establishes the upper bound in (3.15).

B.3.5 Lower bounding K^*

The lower bound is achieved when all columns of $\boldsymbol{H}^*_{(K)}$ are linearly independent. Thus, to ensure that rank $\boldsymbol{H}^*_{(K)}$ is N, $Ks \geq N$. However, if $s \geq R_{\boldsymbol{H}}$, the maximum number of independent columns become $KR_{\boldsymbol{H}}$, and thus we get that $KR_{\boldsymbol{H}} \geq N$. Hence, the lower bound in (3.15) is proved.

As noted in the proof outline, this suffices to establish Theorem 3.3.

Appendix C

Appendix to Chapter 5

C.1 Proof of Proposition 5.1

We first prove a lemma to show that the noise term $\boldsymbol{e}_k$ is bounded, which then enables us to establish the required result.

Lemma C.1. *In our online algorithm given by* (5.2), $\lim_{k\to\infty}\sum_{t=1}^{k}\frac{1}{t}\boldsymbol{e}_t$ *exists and is finite.*

Proof. We define $\boldsymbol{l}_k = \sum_{t=1}^{k}\frac{1}{t}\boldsymbol{e}_t$, and $\mathcal{F}_k$ as the $\sigma-$algebra generated by $\boldsymbol{y}^k$. Then, $\mathbb{E}\left\{\boldsymbol{l}_k|\mathcal{F}_{k-1}\right\} = \mathbb{E}\left\{\boldsymbol{l}_{k-1}|\mathcal{F}_{k-1}\right\} + \frac{1}{k}\mathbb{E}\left\{\boldsymbol{e}_k|\mathcal{F}_{k-1}\right\} = \boldsymbol{l}_{k-1}$. Thus, $\boldsymbol{l}_{k-1}$ is a martingale. Further, using the orthogonality property of martingales [188],

$$\mathbb{E}\left\{\|\boldsymbol{l}_k\|^2\right\} = \sum_{t=1}^{k}\mathbb{E}\left\{\|\boldsymbol{l}_t - \boldsymbol{l}_{t-1}\|^2\right\} = \sum_{t=1}^{k}\frac{1}{t^2}\mathbb{E}\left\{\|\boldsymbol{e}_t\|^2\right\}. \tag{C.1}$$

We note that $\|\boldsymbol{y}\|_\infty < \infty$ a.s., thus (5.8) shows that $\|\boldsymbol{e}_t\| < \infty$ a.s., if $\|\boldsymbol{\gamma}_{k-1}\|_\infty < \infty$. When $\|\boldsymbol{\gamma}_{k-1}\|_\infty \to \infty$, from (5.8), it can be shown that

$$\lim_{\|\boldsymbol{\gamma}\|_\infty\to\infty}\boldsymbol{\Gamma}\boldsymbol{A}^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\mathsf{T}+\boldsymbol{R}\right)^{-1} = \lim_{\|\boldsymbol{\gamma}\|_\infty\to\infty}\|\boldsymbol{\gamma}\|_\infty^{-\frac{1}{2}}\boldsymbol{\Gamma}^{\frac{1}{2}}\left[\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\left(\|\boldsymbol{\gamma}\|_\infty^{-1}\boldsymbol{\Gamma}\right)\boldsymbol{A}^\mathsf{T}\boldsymbol{R}^{-\frac{1}{2}}\right]^\dagger\boldsymbol{R}^{-\frac{1}{2}}. \tag{C.2}$$

Hence, all entries of $\lim_{\gamma\to\infty} \boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)^{-1}$ are finite, and $\|\boldsymbol{e}_t\| < \infty$ with probability one. Thus, $\mathbb{E}\left\{\|\boldsymbol{e}_t\|^2\right\}$ is bounded, and hence by Jensen's inequality and (C.1), the martingale is bounded in $\mathcal{L}^1$. Applying Doob's forward convergence theorem [188] to each coordinate of the martingale $\boldsymbol{l}_k[i], i = 1, 2, \ldots, N$, the limit $\lim_{k\to\infty} \boldsymbol{l}_k = \lim_{k\to\infty} \sum_{t=1}^{k} \frac{1}{t}\boldsymbol{e}_t$ exists, and is finite. □

We now formally prove Proposition 5.1.

Proof. Using (5.2), we have,

$$\boldsymbol{\gamma}_k = \frac{k-1}{k}\boldsymbol{\gamma}_{k-1} + \frac{1}{k}\mathrm{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma}_{k-1}) + \widehat{\boldsymbol{x}}(\boldsymbol{y}_k, \boldsymbol{\gamma}_{k-1})\widehat{\boldsymbol{x}}(\boldsymbol{y}_k, \boldsymbol{\gamma}_{k-1})^{\mathsf{T}}\right\}. \tag{C.3}$$

All entries of $\mathsf{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma}_{k-1}) + \widehat{\boldsymbol{x}}(\boldsymbol{y}_k, \boldsymbol{\gamma}_{k-1})\widehat{\boldsymbol{x}}(\boldsymbol{y}_k, \boldsymbol{\gamma}_{k-1})^{\mathsf{T}}\right\}$ are nonnegative. This ensures that $\boldsymbol{\gamma}_k[i] \geq 0$ for $i = 1, 2, \ldots, N$ and $\forall k$, if $\boldsymbol{\gamma}_0$ is a nonnegative vector. Thus, the sequence $\boldsymbol{\gamma}_k$ is bounded from below.

Next, we use [189, Theorem 7] to show that the sequence is bounded from above, and hence it remains in a compact set. For that, we check if the conditions below hold in our case:

(i) The function $\boldsymbol{f}$ is Lipschitz

(ii) $\lim_{k\to\infty} \sum_{t=1}^{k} \frac{1}{t}\boldsymbol{e}_t$ exists

(iii) The function $\boldsymbol{f}_\infty(\boldsymbol{\gamma}) = \lim_{c\to\infty} \boldsymbol{f}(c\boldsymbol{\gamma})/c$ is continuous, and the ordinary differential equation (ODE)

$$\frac{d}{dt}\boldsymbol{\gamma}(t) = \boldsymbol{f}_\infty(\boldsymbol{\gamma}(t)), \tag{C.4}$$

has the origin as its unique globally asymptotic stable equilibrium.

Since $\boldsymbol{P}(\boldsymbol{\gamma})$ and $\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}$ are positive semidefinite, all of their diagonal entries are nonnegative. Hence, using (5.3),

$$\boldsymbol{f}(\boldsymbol{\gamma}) \geq -\boldsymbol{\gamma} + \mathsf{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathsf{T}}\boldsymbol{R}^{-1}\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\right\}\boldsymbol{R}^{-1}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\} \geq -\boldsymbol{\gamma}, \tag{C.5}$$

where $\boldsymbol{a} \geq \boldsymbol{b}$ denotes that every entry of $\boldsymbol{a}$ is greater than or equal to the corresponding entry of $\boldsymbol{b}$. Further, since the matrix $\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}$ is positive semidefinite, every diagonal entry of $\boldsymbol{P}(\boldsymbol{\gamma}) = \boldsymbol{\Gamma} - \boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}$ is less than the corresponding diagonal entry of $\boldsymbol{\Gamma}$. Thus, we get

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{\gamma}) &\leq \mathsf{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathsf{T}}\boldsymbol{R}^{-1}\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\right\}\boldsymbol{R}^{-1}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\} \\ &\leq \lambda\mathsf{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathsf{T}}\boldsymbol{R}^{-2}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\},\end{aligned} \tag{C.6}$$

where λ is the largest eigenvalue of the positive semidefinite matrix $\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\right\}$, and $\boldsymbol{a} \leq \boldsymbol{b}$ denotes an entry-wise inequality. Thus,

$$-\boldsymbol{\gamma}[i] \leq \boldsymbol{f}(\boldsymbol{\gamma})[i] \leq \lambda\mathsf{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathsf{T}}\boldsymbol{R}^{-2}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\}[i], \tag{C.7}$$

for $i = 1, 2, \ldots, N$. To further bound the last term of the inequality, we use (5.6) to get

$$\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathsf{T}}\boldsymbol{R}^{-2}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma}) = \boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{B}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{B}^{\mathsf{T}}\boldsymbol{\Gamma}^{\frac{1}{2}}. \tag{C.8}$$

where $\boldsymbol{B} \triangleq \boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)^{-\frac{1}{2}}$. This implies

$$\mathsf{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathsf{T}}\boldsymbol{R}^{-2}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\}[i] = \boldsymbol{\gamma}[i]\boldsymbol{B}[i]^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{B}[i] \tag{C.9}$$

$$\leq \boldsymbol{\gamma}[i]\boldsymbol{B}[i]^{\mathsf{T}}\boldsymbol{R}^{-1}\boldsymbol{B}[i], \tag{C.10}$$

where $\boldsymbol{B}[i] \in \mathbb{R}^N$ is the i^{th} column of $\boldsymbol{B}^{\mathsf{T}}$. Then, we have

$$\begin{aligned}\boldsymbol{B}\boldsymbol{B}^{\mathsf{T}} &= \boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}} \\ &= \boldsymbol{I}-\left(\boldsymbol{I}+\boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{R}^{-1}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right)^{-1}. \end{aligned} \tag{C.11}$$

This shows that $\boldsymbol{I}-\boldsymbol{B}\boldsymbol{B}^{\mathsf{T}}$ is a positive semidefinite matrix, and its diagonal entries are nonnegative. Thus, $\boldsymbol{B}[i]^{\mathsf{T}}\boldsymbol{B}[i] \leq 1$, for $i = 1, 2, \ldots, N$. Hence, we get

$$\mathsf{Diag}\left\{\boldsymbol{P}(\boldsymbol{\gamma})\boldsymbol{A}^{\mathsf{T}}\boldsymbol{R}^{-2}\boldsymbol{A}\boldsymbol{P}(\boldsymbol{\gamma})\right\}[i] \leq \bar{\lambda}\boldsymbol{\gamma}[i], \tag{C.12}$$

where $\bar{\lambda}$ is the largest eigenvalue of $\boldsymbol{R}^{-1}$. Substituting this relation in (C.7), we get

$$-\boldsymbol{\gamma}[i] \leq \boldsymbol{f}(\boldsymbol{\gamma})[i] \leq \bar{\lambda}\lambda\boldsymbol{\gamma}[i]. \tag{C.13}$$

Thus, (i) is satisfied. The assumption (ii) is true by Lemma C.1. To check (iii), we start with (5.7) to get

$$\begin{aligned}\boldsymbol{f}_\infty(\boldsymbol{\gamma}) = \lim_{c\to\infty}\frac{1}{c}\mathrm{Diag}\Big\{&c^2\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}\big(c\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\big)^{-1}\big(\mathbb{E}\left\{\boldsymbol{y}_k\boldsymbol{y}_k^{\mathsf{T}}\right\} \\ &-c\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}-\boldsymbol{R}\big)\left(c\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}\Big\} \end{aligned} \tag{C.14}$$

$$= -\lim_{c\to\infty}\mathrm{Diag}\left\{\boldsymbol{\Gamma}\left(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right)^{\mathsf{T}}\left[\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\left(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right)^{\mathsf{T}}+\boldsymbol{I}/c\right]^{-1}\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right\} \tag{C.15}$$

$$= -\mathsf{Diag}\left\{\boldsymbol{\Gamma}\left(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right)^{\dagger}\left(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right)\right\}. \tag{C.16}$$

Note that $\mathrm{Rank}\{(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}})\} = \min\{\mathsf{Rank}\{\boldsymbol{\Gamma}\}, m\}$. For the case when $\mathsf{Rank}\{\boldsymbol{\Gamma}\} < m$, we have $\mathrm{Rank}\{(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}})\} = \mathsf{Rank}\{\boldsymbol{\Gamma}\}$, and thus, $\boldsymbol{f}_\infty(\boldsymbol{\gamma}) = -\boldsymbol{\gamma}$. Since $\mathbf{0}$ is the only globally asymptotically stable equilibrium of the ODE $\frac{d}{dt}\boldsymbol{\gamma}(t) = -\boldsymbol{\gamma}(t)$, (iii) holds. When

$\text{Rank}\{\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\} = m$, we have

$$\left(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\right)^{\dagger} = \boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{R}^{-\frac{1}{2}}\left(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{R}^{-\frac{1}{2}}\right)^{-1}, \tag{C.17}$$

which implies the following:

$$(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}})^{\dagger}(\boldsymbol{R}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}) = \boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}. \tag{C.18}$$

Since the diagonal entries of $\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{A}$ are positive, the only possible equilibrium for the ODE is $\mathbf{0}$. However, when $\boldsymbol{\gamma} = \mathbf{0}$, $\text{Rank}\{\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\} \neq m$ which is a contradiction. Hence, there is no equilibrium point with $\text{Rank}\{\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{A}\boldsymbol{\Gamma}^{\frac{1}{2}}\} = m$. Thus, (iii) holds, and the proof is complete. □

C.2 Proof of Theorem 5.1

Before we prove the main theorem, we need two lemmas.

Lemma C.2. *The solution set of* $\boldsymbol{f}(\boldsymbol{\gamma}) = \mathbf{0}$ *is* $\{\mathbf{0}\} \cup \{\boldsymbol{\gamma} \in \mathbb{R}^N : \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}}\}$, *when* $\mathbb{E}\left\{\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\right\} = \boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}$.

Proof. From (5.7), we get

$$\boldsymbol{f}(\boldsymbol{\gamma}) = \text{Diag}\left\{\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{A}\left(\boldsymbol{\Gamma}_{\mathsf{opt}} - \boldsymbol{\Gamma}\right)\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}\right\}. \tag{C.19}$$

Clearly, $\boldsymbol{\gamma} = \mathbf{0}$ is a zero of $\boldsymbol{f}(\boldsymbol{\gamma})$. Let us consider the solutions whose support is the vector $\boldsymbol{s} \in \{0,1\}^N$ and $\boldsymbol{s} \neq \mathbf{0}$, and let the number of nonzero entries in $\boldsymbol{s}$ be denoted by s. The union of the solutions over all possible supports gives the solution set. Let $\boldsymbol{\gamma}_{\boldsymbol{s}} \in \mathbb{R}^{s\times 1}$ be the vector of nonzero entries of $\boldsymbol{\gamma}$ and $\boldsymbol{A}_{\boldsymbol{s}} \in \mathbb{R}^{m\times s}$ be the matrix formed by restricting $\boldsymbol{A}$

to the s columns corresponding to the support $\boldsymbol{s}$. Let $\boldsymbol{B_s} = \left(\boldsymbol{A\Gamma A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-\frac{1}{2}} \boldsymbol{A_s} \in \mathbb{R}^{m\times s}$, and $\boldsymbol{B} = \left(\boldsymbol{A\Gamma A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-\frac{1}{2}} \boldsymbol{A} \in \mathbb{R}^{m\times N}$. Then, the reduced set of equations corresponding to $\boldsymbol{f}(\boldsymbol{\gamma}) = \boldsymbol{0}$ is given by

$$\mathsf{Diag}\left\{\boldsymbol{B_s}^{\mathsf{T}}\boldsymbol{B_s}\boldsymbol{\Gamma}_{\boldsymbol{s}}\boldsymbol{B_s}^{\mathsf{T}}\boldsymbol{B_s}\right\} = \mathrm{Diag}\left\{\boldsymbol{B_s}^{\mathsf{T}}\boldsymbol{B}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{B}^{\mathsf{T}}\boldsymbol{B_s}\right\}, \tag{C.20}$$

where $\boldsymbol{\Gamma_s} = \mathsf{Diag}\left\{\boldsymbol{\gamma_s}\right\}$ is an invertible matrix. We note that the above system of equations is linear in the vector $\boldsymbol{\gamma_s}$, for any given fixed matrices $\boldsymbol{B_s}$ and $\boldsymbol{B}$. However, $\mathsf{Diag}\left\{\boldsymbol{B_s}^{\mathsf{T}}\boldsymbol{B_s}\boldsymbol{\Gamma}_{\boldsymbol{s}}\boldsymbol{B_s}^{\mathsf{T}}\boldsymbol{B_s}\right\} = \left(\boldsymbol{B_s}^{\mathsf{T}}\boldsymbol{B_s}\right) \circ \left(\boldsymbol{B_s}^{\mathsf{T}}\boldsymbol{B_s}\right)\boldsymbol{\gamma_s}$, where $\circ$ represents the Hadamard product of matrices. Thus, the solution set of the system of equations is an affine space $\mathcal{U}_{\boldsymbol{s}}$ of dimension given by

$$\begin{aligned}
\dim(\mathcal{U}_{\boldsymbol{s}}) &= s - \mathsf{Rank}\left\{\left(\boldsymbol{B_s}^{\mathsf{T}}\boldsymbol{B_s}\right) \circ \left(\boldsymbol{B_s}^{\mathsf{T}}\boldsymbol{B_s}\right)\right\} && \text{(C.21)}\\
&= s - \mathsf{Rank}\left\{\left(\boldsymbol{B_s} \odot \boldsymbol{B_s}\right)^{\mathsf{T}}\left(\boldsymbol{B_s} \odot \boldsymbol{B_s}\right)\right\} && \text{(C.22)}\\
&= s - \mathsf{Rank}\left\{\boldsymbol{B_s} \odot \boldsymbol{B_s}\right\}. && \text{(C.23)}
\end{aligned}$$

We now consider another affine space $\mathcal{W}_{\boldsymbol{s}}$ of dimension $s - \mathsf{Rank}\left\{\boldsymbol{B_s} \odot \boldsymbol{B_s}\right\}$ given by the set of $\boldsymbol{\gamma_s}$ satisfying

$$\mathrm{vec}\left\{\boldsymbol{B_s}\boldsymbol{\Gamma}_{\boldsymbol{s}}\boldsymbol{B_s}^{\mathsf{T}}\right\} = \left(\boldsymbol{B_s} \odot \boldsymbol{B_s}\right)\boldsymbol{\gamma_s} = \mathrm{vec}\left\{\boldsymbol{B}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{B}^{\mathsf{T}}\right\}. \tag{C.24}$$

It is easy to see that $\mathcal{W}_{\boldsymbol{s}} \subseteq \mathcal{U}_{\boldsymbol{s}}$ and $\dim(\mathcal{U}_{\boldsymbol{s}}) = \dim(\mathcal{W}_{\boldsymbol{s}})$, which implies $\mathcal{W}_{\boldsymbol{s}} = \mathcal{U}_{\boldsymbol{s}}$. Rearranging, we get, for $\boldsymbol{\gamma}_{\boldsymbol{s}} \in \mathcal{U}_{\boldsymbol{s}}$,

$$\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-\frac{1}{2}} \boldsymbol{A}_{\boldsymbol{s}}\boldsymbol{\Gamma}_{\boldsymbol{s}}\boldsymbol{A}_{\boldsymbol{s}}^{\mathsf{T}} \left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-\frac{1}{2}} = \left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-\frac{1}{2}} \boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}} \left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-\frac{1}{2}}. \quad \text{(C.25)}$$

Thus,

$$\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A}_{\boldsymbol{s}}\boldsymbol{\Gamma}_{\boldsymbol{s}}\boldsymbol{A}_{\boldsymbol{s}}^{\mathsf{T}} = \boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}}, \quad \text{(C.26)}$$

and $\mathcal{U}_{\boldsymbol{s}} \subseteq \{\boldsymbol{\gamma} : \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}}\}$, for all support sets $\boldsymbol{s} \neq \boldsymbol{0}$. From (C.19), it is easy to see that $\{\boldsymbol{\gamma} \in \mathbb{R}^N : \boldsymbol{A}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathsf{opt}}\right)\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{0}\}$ satisfies $\boldsymbol{f}(\boldsymbol{\gamma}) = \boldsymbol{0}$. Therefore, $\bigcup_{\boldsymbol{s} \in \{0,1\}^N \setminus \boldsymbol{0}} \mathcal{U}_{\boldsymbol{s}} = \{\boldsymbol{\gamma} : \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}}\}$. Thus, we get that the solution set of $\boldsymbol{f}(\boldsymbol{\gamma}) = \boldsymbol{0}$ is $\{\boldsymbol{0}\} \cup \{\boldsymbol{\gamma} \in \mathbb{R}^N : \boldsymbol{A}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathsf{opt}}\right)\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{0}\}$. □

We define some notation to state the next lemma. The notation $\boldsymbol{X} \succ \boldsymbol{0}$ denotes that $\boldsymbol{X}$ is a positive definite matrix and $\boldsymbol{X} \succcurlyeq \boldsymbol{0}$ denotes that $\boldsymbol{X}$ is a positive semidefinite matrix.

Lemma C.3. *The set* $\mathbb{O} = \{\boldsymbol{\gamma} \in \mathbb{R}^N : \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R} \succ \boldsymbol{0}\}$ *is an open set and its closure is* $\{\boldsymbol{\gamma} \in \mathbb{R}^N : \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R} \succcurlyeq \boldsymbol{0}\}$.

Proof. Let $\boldsymbol{\gamma} \in \mathbb{O}$. Then, $\boldsymbol{u}^{\mathsf{T}}(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R})\boldsymbol{u} > 0 \ \forall \boldsymbol{u} \in \mathbb{R}^m \setminus \{\boldsymbol{0}\}$, and the minimum eigenvalue of $\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}$ is strictly greater than some $\beta > 0$. We need to show that there exists an $\epsilon > 0$ such that $\boldsymbol{A}\widetilde{\boldsymbol{\Gamma}}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}$ is positive definite for all $\widetilde{\boldsymbol{\gamma}}$ in the ϵ-neighborhood of $\boldsymbol{\gamma}$, i.e., $\|\boldsymbol{\gamma} - \widetilde{\boldsymbol{\gamma}}\| < \epsilon$.

For a given $\boldsymbol{u} \in \mathbb{R}^m \setminus \{\boldsymbol{0}\}$, if $\boldsymbol{u}^{\mathsf{T}}(\boldsymbol{A}\widetilde{\boldsymbol{\Gamma}}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R})\boldsymbol{u} \geq \boldsymbol{u}^{\mathsf{T}}(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R})\boldsymbol{u}$, then $\boldsymbol{u}^{\mathsf{T}}(\boldsymbol{A}\widetilde{\boldsymbol{\Gamma}}\boldsymbol{A}^{\mathsf{T}} +$

$\boldsymbol{R})\boldsymbol{u} > 0$. Otherwise,

$$\boldsymbol{u}^\mathsf{T}\left(\boldsymbol{A}\widetilde{\boldsymbol{\Gamma}}\boldsymbol{A}^\mathsf{T}+\boldsymbol{R}\right)\boldsymbol{u} = \boldsymbol{u}^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\mathsf{T}+\boldsymbol{R}\right)\boldsymbol{u} - \left|\boldsymbol{u}^\mathsf{T}\boldsymbol{A}\left(\boldsymbol{\Gamma}-\widetilde{\boldsymbol{\Gamma}}\right)\boldsymbol{A}^\mathsf{T}\boldsymbol{u}\right| \tag{C.27}$$

$$\geq \left(\beta - \|\boldsymbol{\Gamma}-\widetilde{\boldsymbol{\Gamma}}\|_2\|\boldsymbol{A}\|_2^2\right)\|\boldsymbol{u}\|^2 \tag{C.28}$$

$$\geq \left(\beta - \epsilon\|\boldsymbol{A}\|_2^2\right)\|\boldsymbol{u}\|^2, \tag{C.29}$$

where $\|\cdot\|_2$ denotes the induced l_2 norm. We can always find an $\epsilon > 0$ such that $(\beta - \epsilon\|\boldsymbol{A}\|_2^2) > 0$. Therefore, $\boldsymbol{u}^\mathsf{T}(\boldsymbol{A}\widetilde{\boldsymbol{\Gamma}}\boldsymbol{A}^\mathsf{T}+\boldsymbol{R})\boldsymbol{u} > 0\,\forall\,\boldsymbol{u} \in \mathbb{R}^m \setminus \{\boldsymbol{0}\}$, and thus $\mathbb{O}$ is an open set.

To prove the second part of the lemma, suppose the sequence $\boldsymbol{\gamma}_k \in \mathbb{O}$ converges to $\boldsymbol{\gamma}$. Then, for any vector $\boldsymbol{u} \in \mathbb{R}^m \setminus \{\boldsymbol{0}\}$, $\boldsymbol{u}^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^\mathsf{T}+\boldsymbol{R}\right)\boldsymbol{u}$ converges to $\boldsymbol{u}^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\mathsf{T}+\boldsymbol{R}\right)\boldsymbol{u}$ by the continuity of the function. Therefore,

$$\boldsymbol{u}^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^\mathsf{T}+\boldsymbol{R}\right)\boldsymbol{u} > 0 \implies \boldsymbol{u}^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\mathsf{T}+\boldsymbol{R}\right)\boldsymbol{u} \geq 0. \tag{C.30}$$

Thus $\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\mathsf{T}+\boldsymbol{R} \succcurlyeq \boldsymbol{0}$. Conversely, if there is exists a $\boldsymbol{\gamma} \in \mathbb{R}^m$ such that $\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\mathsf{T}+\boldsymbol{R} \succcurlyeq \boldsymbol{0}$, the sequence $\boldsymbol{\gamma}_k = \boldsymbol{\gamma} + (1/k)\boldsymbol{1}$ converges to $\boldsymbol{\gamma}$. We also note that $\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^\mathsf{T}+\boldsymbol{R} = \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^\mathsf{T}+\boldsymbol{R}+(1/k)\boldsymbol{A}\boldsymbol{A}^\mathsf{T} \succ \boldsymbol{0}$ since $\boldsymbol{A}$ has full row rank. Thus, there exists a sequence $\{\boldsymbol{\gamma}_k\} \in \mathbb{O}$ that converges to $\boldsymbol{\gamma}$. Hence, the proof is complete. □

Proof of Theorem 5.1

We prove the convergence using [190, Theorem 2] which states that: Suppose $\boldsymbol{f}(\cdot)$ is a continuous vector field defined on an open set $\mathbb{O} \subset \mathbb{R}^N$ such that $\mathbb{G} = \{\boldsymbol{\gamma} \in \mathbb{O} : \boldsymbol{f}(\boldsymbol{\gamma}) = \boldsymbol{0}\}$ is a compact subset of $\mathbb{O}$. Then the distance of the sequence $\boldsymbol{\gamma}_k$ given by (5.2) to the set $\mathbb{G}$ converges to 0 *a.s.* provided:

(i) There exists a $\mathcal{C}^1$ function $V : \mathbb{O} \to \mathbb{R}_+$ such that

(a) $V(\boldsymbol{\gamma}) \to \infty$ if $\boldsymbol{\gamma} \to$ the boundary of $\mathbb{O}$ or $\|\boldsymbol{\gamma}\| \to \infty$

(b) $\langle \nabla_{\boldsymbol{\gamma}} V(\boldsymbol{\gamma}), \boldsymbol{f}(\boldsymbol{\gamma}) \rangle < 0$, $\forall \boldsymbol{\gamma} \notin \mathbb{G}$.

(ii) $\boldsymbol{\gamma}_k$ belongs to a compact set of $\mathbb{O}$.

(iii) $\lim_{k\to\infty} \sum_{t=1}^{k} \frac{1}{t} \boldsymbol{e}_t$ exists and is finite.

To check whether assumptions (i)-(iii) hold in our case, we define the set $\mathbb{O} = \{\boldsymbol{\gamma} : \mathsf{Rank}\left\{\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right\} = m\}$ which is an open set by Lemma C.3. Note that $\boldsymbol{f}$ is a continuous function of $\boldsymbol{\gamma}$. Also, the inverse image of the compact set $\{\boldsymbol{0}\}$ by $\boldsymbol{f}(\boldsymbol{\gamma})$ is compact, and hence, $\mathbb{G}$ is a compact subset of $\mathbb{O}$.

We define the $\mathcal{C}^1$ function in (i) as follows:

$$V(\boldsymbol{\gamma}) = \mathsf{Tr}\left\{\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\left(\boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)\right\} - \log\left|\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\left(\boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)\right|. \quad \text{(C.31)}$$

Note that $V(\boldsymbol{\gamma}) - m$ gives the KL divergence between the distributions $\mathcal{N}(\boldsymbol{0}, \boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R})$ and $\mathcal{N}(\boldsymbol{0}, \boldsymbol{A}\boldsymbol{\Gamma}_{\mathsf{opt}}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R})$. Therefore, $V(\boldsymbol{\gamma}) \geq m > 0$. By Lemma C.3, if $\boldsymbol{\gamma}$ is on the boundary of $\mathbb{O}$, at least one eigenvalue of $\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}$ is zero. Hence, (ia) is satisfied. The gradient of $V(\boldsymbol{\gamma})$ is given by

$$\begin{aligned}\nabla_{\boldsymbol{\gamma}} V(\boldsymbol{\gamma}) &= \mathsf{Diag}\left\{\boldsymbol{A}^{\mathsf{T}} \nabla_{\{\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}}+\boldsymbol{R}\}} V\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)\boldsymbol{A}\right\} \\ &= \mathrm{Diag}\left\{\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{A}\left(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathsf{opt}}\right)\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{\Gamma}\boldsymbol{A}^{\mathsf{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{A}\right\}. \quad \text{(C.32)}\end{aligned}$$

Substituting this relation in (5.7) gives $\boldsymbol{f}(\boldsymbol{\gamma}) = -\boldsymbol{\Gamma}^2 \nabla_{\boldsymbol{\gamma}} V(\boldsymbol{\gamma})$. Therefore, for $\boldsymbol{\gamma} \in \mathbb{O} \setminus \mathbb{G}$,

we have $\langle \nabla_{\boldsymbol{\gamma}} V(\boldsymbol{\gamma}), \boldsymbol{f}(\boldsymbol{\gamma}) \rangle < 0$. Thus, (ib) is satisfied.

Assumptions (ii) and (iii) holds because of Proposition 5.1 and Lemma C.1, respectively. Hence, $\boldsymbol{\gamma}_k$ converges to the set $\mathbb{G}$. Further, Proposition 5.1 shows that $\boldsymbol{\gamma}_k \geq 0$, and hence, we get that $\boldsymbol{\gamma}_k$ converges to the set $\{\mathbf{0}\} \cup \{\boldsymbol{\gamma} \in \mathbb{R}_+^N : \boldsymbol{A}(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathsf{opt}})\boldsymbol{A}^{\mathsf{T}} = \mathbf{0}\}$. Finally, if $\mathsf{Rank}\{\boldsymbol{A} \odot \boldsymbol{A}\} = N$, then $\left\{\boldsymbol{\gamma} \in \mathbb{R}_+^N : \boldsymbol{A}(\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathsf{opt}})\boldsymbol{A}^{\mathsf{T}} = \mathbf{0}\right\} = \left\{\boldsymbol{\gamma}_{\mathsf{opt}}\right\}$. Thus, the proof is complete. ■

C.3 Proof of Proposition 5.2

Proof. From (5.14), we get,

$$\begin{aligned}
\boldsymbol{P}_k &= \boldsymbol{P}_{k-1} - \boldsymbol{P}_{k-1}\boldsymbol{A}_k^{\mathsf{T}}\left(\boldsymbol{A}_k\boldsymbol{P}_{k-1}\boldsymbol{A}_k^{\mathsf{T}} + \boldsymbol{R}_k\right)^{-1}\boldsymbol{A}_k\boldsymbol{P}_{k-1} \\
&= \left(\boldsymbol{P}_{k-1}^{-1} + \boldsymbol{A}_k^{\mathsf{T}}\boldsymbol{R}_k^{-1}\boldsymbol{A}_k\right)^{-1} \qquad \text{(C.33)} \\
&= \left(\boldsymbol{P}_0^{-1} + \sum_{t=1}^{k}\boldsymbol{A}_t^{\mathsf{T}}\boldsymbol{R}_t^{-1}\boldsymbol{A}_t\right)^{-1}. \qquad \text{(C.34)}
\end{aligned}$$

Let $\boldsymbol{Q} \triangleq \lim_{k\to\infty}\frac{1}{k}\sum_{t=1}^{k}\boldsymbol{A}_t^{\mathsf{T}}\boldsymbol{R}^{-1}\boldsymbol{A}_t$. From assumptions A1 and A2, we have

$$\boldsymbol{Q} \triangleq \lim_{k\to\infty}\frac{1}{k}\sum_{t=1}^{k}\boldsymbol{A}_t^{\mathsf{T}}\boldsymbol{R}^{-1}\boldsymbol{A}_t, \qquad \text{(C.35)}$$

Thus, we get

$$\boldsymbol{Q} = \boldsymbol{S} + \mathbb{E}\left\{\boldsymbol{A}_t^{\mathsf{T}}\right\}\boldsymbol{R}^{-1}\mathbb{E}\left\{\boldsymbol{A}_t\right\}, \qquad \text{(C.36)}$$

where $\boldsymbol{S} \triangleq \mathsf{Diag}\left\{\mathsf{Tr}\left\{\boldsymbol{R}^{-1}\mathrm{cov}(\boldsymbol{A}_t[i])\right\}, i = 1, 2, \ldots, N\right\}$. Further, since $\boldsymbol{A}_t$ is random, $\boldsymbol{S}$ is a positive definite matrix and hence, $\boldsymbol{Q}$ is a positive definite matrix. Let $\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathsf{T}}$ be the eigen decomposition such that $\boldsymbol{\Lambda}$ is a diagonal matrix containing the positive eigen

values of $\boldsymbol{Q}$. Then,

$$\lim_{k\to\infty} \boldsymbol{P}_k = \lim_{k\to\infty} \boldsymbol{U}(\boldsymbol{U}\boldsymbol{P}_0^{-1}\boldsymbol{U}^\mathsf{T} + k\boldsymbol{\Lambda})^{-1}\boldsymbol{U}^\mathsf{T}. \tag{C.37}$$

Let $\lambda_{\min} > 0$ and $\gamma_{\min} > 0$ be the smallest diagonal entries of $\boldsymbol{\Lambda}$ and $\boldsymbol{P}_0^{-1}$, respectively. Then, the largest eigenvalue of $(\boldsymbol{U}\boldsymbol{P}_0^{-1}\boldsymbol{U}^\mathsf{T} + k\boldsymbol{\Lambda})^{-1}$, denoted by $\tilde{\lambda}_k$, can be bounded using Weyl's inequality as follows:

$$0 \leq \tilde{\lambda}_k \leq \frac{1}{\gamma_{\min} + k\lambda_{\min}}. \tag{C.38}$$

Hence, we get

$$\lim_{k\to\infty} \boldsymbol{P}_k = \boldsymbol{U} \lim_{k\to\infty} (\boldsymbol{U}\boldsymbol{P}_0^{-1}\boldsymbol{U}^\mathsf{T} + k\boldsymbol{\Lambda})^{-1}\boldsymbol{U}^\mathsf{T} = \boldsymbol{0}. \tag{C.39}$$

From (5.12), we get $\lim_{k\to\infty} \boldsymbol{J}_k = \boldsymbol{0}$, and from (5.13), we get $\widehat{\boldsymbol{x}}_k = \widehat{\boldsymbol{x}}_{k-1}$ as $k \to \infty$. Hence, the algorithm converges.

□

C.4 Proof of Theorem 5.3

Proof. Let the true solution be $\boldsymbol{x}$, and $\widehat{\boldsymbol{x}}_\infty \triangleq \lim_{k\to\infty} \widehat{\boldsymbol{x}}_k$ as guaranteed by Proposition 5.2. From (5.13),

$$\widehat{\boldsymbol{x}}_k = (\boldsymbol{I} - \boldsymbol{J}_k\boldsymbol{A}_k)\,\widehat{\boldsymbol{x}}_{k-1} + \boldsymbol{J}_k\boldsymbol{y}_k \tag{C.40}$$

$$= \boldsymbol{P}_k\boldsymbol{P}_{k-1}^{-1}\widehat{\boldsymbol{x}}_{k-1} + \boldsymbol{J}_k\boldsymbol{y}_k \tag{C.41}$$

$$= \boldsymbol{P}_k\boldsymbol{P}_0^{-1}\widehat{\boldsymbol{x}}_0 + \boldsymbol{P}_k\sum_{t=1}^{k}\boldsymbol{P}_t^{-1}\boldsymbol{J}_t\boldsymbol{y}_t. \tag{C.42}$$

Using (C.39), we get

$$\boldsymbol{x}_\infty = \lim_{k\to\infty} \boldsymbol{P}_k\sum_{t=1}^{k}\boldsymbol{P}_t^{-1}\boldsymbol{J}_t\boldsymbol{y}_t. \tag{C.43}$$

Since $\boldsymbol{y}_t = \boldsymbol{A}_t\boldsymbol{x} + \boldsymbol{w}_t$, and from (5.14) $\boldsymbol{J}_t\boldsymbol{A}_t = \boldsymbol{I} - \boldsymbol{P}_t\boldsymbol{P}_{t-1}^{-1}$,

$$\begin{aligned}
\widehat{\boldsymbol{x}}_\infty &= \lim_{k\to\infty} \boldsymbol{P}_k \sum_{t=1}^{k} \left[\boldsymbol{P}_t^{-1}(\boldsymbol{I} - \boldsymbol{P}_t\boldsymbol{P}_{t-1}^{-1})\boldsymbol{x} + \boldsymbol{P}_t^{-1}\boldsymbol{J}_t\boldsymbol{w}_t\right] \\
&= \lim_{k\to\infty} \left[\left(\boldsymbol{I} - \boldsymbol{P}_k\boldsymbol{P}_0^{-1}\right)\boldsymbol{x} + \boldsymbol{P}_k \sum_{t=1}^{k} \boldsymbol{P}_t^{-1}\boldsymbol{J}_t\boldsymbol{w}_t\right] \qquad \text{(C.44)} \\
&= \boldsymbol{x} + \lim_{k\to\infty} \boldsymbol{P}_k \sum_{t=1}^{k} \boldsymbol{P}_t^{-1}\boldsymbol{J}_t\boldsymbol{w}_t. \qquad \text{(C.45)}
\end{aligned}$$

We now consider the term $\boldsymbol{P}_t^{-1}\boldsymbol{J}_t$ to simplify the second term in the above expression, and using (C.33) and (5.12) we get

$$\begin{aligned}
\boldsymbol{P}_t^{-1}\boldsymbol{J}_t &= \left(\boldsymbol{P}_{t-1}^{-1} + \boldsymbol{A}_t^{\mathsf{T}}\boldsymbol{R}^{-1}\boldsymbol{A}_t\right)\boldsymbol{P}_{t-1}\boldsymbol{A}_t^{\mathsf{T}}\left(\boldsymbol{A}_t\boldsymbol{P}_{t-1}\boldsymbol{A}_t^{\mathsf{T}} + \boldsymbol{R}\right)^{-1} \qquad \text{(C.46)} \\
&= \boldsymbol{A}_t^{\mathsf{T}}\big(\boldsymbol{I} + \boldsymbol{R}^{-1}\boldsymbol{A}_t\boldsymbol{P}_{t-1}\boldsymbol{A}_t^{\mathsf{T}}\big)\big(\boldsymbol{A}_t\boldsymbol{P}_{t-1}\boldsymbol{A}_t^{\mathsf{T}} + \boldsymbol{R}\big)^{-1} \qquad \text{(C.47)} \\
&= \boldsymbol{A}_t^{\mathsf{T}}\boldsymbol{R}^{-1}. \qquad \text{(C.48)}
\end{aligned}$$

Thus,

$$\widehat{\boldsymbol{x}}_\infty = \boldsymbol{x} + \lim_{k\to\infty} (k\boldsymbol{P}_k)\left(\frac{1}{k}\sum_{t=1}^{k} \boldsymbol{A}_t^{\mathsf{T}}\boldsymbol{R}^{-1}\boldsymbol{w}_t\right). \qquad \text{(C.49)}$$

We note that

$$\lim_{k\to\infty} \frac{1}{k}\sum_{t=1}^{k} \boldsymbol{A}_t^{\mathsf{T}}\boldsymbol{R}^{-1}\boldsymbol{w}_t = \mathbb{E}\left\{\boldsymbol{A}_t^{\mathsf{T}}\boldsymbol{R}^{-1}\boldsymbol{w}_t\right\} = \mathbb{E}\left\{\boldsymbol{A}_t\right\}^{\mathsf{T}}\boldsymbol{R}^{-1}\mathbb{E}\left\{\boldsymbol{w}_t\right\} = \boldsymbol{0}. \qquad \text{(C.50)}$$

Here, we use the fact that $\boldsymbol{A}_t$ and $\boldsymbol{w}_t$ are independent and the mean of $\boldsymbol{w}_t$ is zero. Further, from (C.37), we get

$$\lim_{k\to\infty} k\boldsymbol{P}_k = \lim_{k\to\infty} k\boldsymbol{U}(\boldsymbol{U}\boldsymbol{P}_0^{-1}\boldsymbol{U}^{\mathsf{T}} + k\boldsymbol{\Lambda})^{-1}\boldsymbol{U}^{\mathsf{T}} = \lim_{k\to\infty}(k^{-1}\boldsymbol{P}_0^{-1} + \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathsf{T}})^{-1} = \boldsymbol{U}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^{\mathsf{T}}. \qquad \text{(C.51)}$$

Substituting (C.50) and (C.51) in (C.49), we get $\widehat{\boldsymbol{x}}_\infty = \boldsymbol{x}$, and the proof is complete. □

Appendix D

Appendix to Chapter 6

D.1 Proof of Proposition 6.1

Proof. For the AM procedure, since we optimize one column of $\boldsymbol{A}$ at a time, it is easy to see that

$$g\left(\boldsymbol{A}^{(r,u-1)}\right) \geq g\left(\boldsymbol{A}^{(r,u)}\right). \tag{D.1}$$

The above relation holds even if we skip the update of a column when $\left\|\boldsymbol{v}_i^{(r,u)}\right\| = 0$, in which case the value of the cost function remains unchanged. Similarly, from (6.15), the sequence $\left\{g\left(\boldsymbol{A}^{(r,u)}\right)\right\}_{u\in\mathbb{N}}$ generated by the ALS algorithm is also nonincreasing. Thus, we conclude that in both cases, the sequence $\left\{g\left(\boldsymbol{A}^{(r,u)}\right)\right\}_{u\in\mathbb{N}}$ is a nonincreasing sequence bounded by $g\left(\boldsymbol{A}^{(r,0)}\right)$ from above. From (6.10), we get

$$g\left(\boldsymbol{A}\right) = \frac{1}{2}\mathsf{Tr}\left\{\left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}\right)^{\mathsf{T}}\left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}\right) + \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathsf{T}}\right\} - \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{Y}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{M}^{\mathsf{T}}\boldsymbol{Y} + \boldsymbol{\Sigma}\right\} - N/2 \tag{D.2}$$

$$\geq -\frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{Y}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{M}^{\mathsf{T}}\boldsymbol{Y} + \boldsymbol{\Sigma}\right\} - N/2. \tag{D.3}$$

Therefore, the nonincreasing sequence $\left\{g\left(\boldsymbol{A}^{(r,u)}\right)\right\}_{u\in\mathbb{N}}$ is bounded from below, and hence it converges. □

D.2 Proof of Proposition 6.2

Proof. The first part of the result directly follows from the properties of AM. Further, any stationary point of the cost function takes the following form:

$$\boldsymbol{A}\boldsymbol{L} = \boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right), \tag{D.4}$$

for some diagonal matrix $\boldsymbol{L}$. From (6.13), we get

$$G(\boldsymbol{A})_i \left\|\boldsymbol{v}_i\right\| = \boldsymbol{v}_i, \tag{D.5}$$

where

$$\begin{aligned}\boldsymbol{v}_i &= \sum_{k=1}^{K}\boldsymbol{\mu}_k[i]\boldsymbol{y}_k - \sum_{j=1}^{i-1}\boldsymbol{\Sigma}[i,j]G(\boldsymbol{A})_j - \sum_{j=i+1}^{N}\boldsymbol{\Sigma}[i,j]\boldsymbol{A}_j \\ &= \left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}}\right)_i - \boldsymbol{G}(\boldsymbol{A})\left(\hat{\boldsymbol{\Sigma}}^{\mathsf{T}}\right)_i - \boldsymbol{A}\hat{\boldsymbol{\Sigma}}_i,\end{aligned} \tag{D.6}$$

where $\hat{\boldsymbol{\Sigma}}$ is a lower triangular matrix with zero diagonal entries and $\hat{\boldsymbol{\Sigma}} + \hat{\boldsymbol{\Sigma}}^{\mathsf{T}} = \boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}$. When $\boldsymbol{A}$ is a fixed point of G, we get

$$\boldsymbol{v}_i = \left(\boldsymbol{Y}\boldsymbol{M}^{\mathsf{T}}\right)_i - \boldsymbol{A}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)_i. \tag{D.7}$$

Now, from (D.5) and (D.7), it can be seen that $\boldsymbol{A}$ satisfies (D.4) with $\boldsymbol{L}_{ii} = \left\|\boldsymbol{v}_i\right\| \geq 0$, which concludes the proof. □

D.3 Proof of Theorem 6.1

The proof of the theorem rests on the following lemmas.

Lemma D.1. *Let* $\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$ *be a sequence generated by the ALS procedure. Then, there exists* $C_1 > 0$ *such that*

$$\tilde{g}\left(\boldsymbol{A}^{(r,u-1)}\right) - \tilde{g}\left(\boldsymbol{A}^{(r,u)}\right) \geq C_1 \left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)}\right\|^2. \tag{D.8}$$

Proof. We note from (6.19) that

$$\boldsymbol{A}_i^{(r,u)} = \frac{\boldsymbol{A}_i^{(r,u-1)} + \beta^p \alpha \boldsymbol{Z}_i^{(r,u-1)}}{\left\|\boldsymbol{A}_i^{(r,u-1)} + \beta^p \alpha \boldsymbol{Z}_i^{(r,u-1)}\right\|}. \tag{D.9}$$

Also, from (6.18), we know that

$$\boldsymbol{A}_i^{(r,u-1)\mathsf{T}} \boldsymbol{Z}_i^{(r,u-1)} = 0. \tag{D.10}$$

Therefore, we get

$$\begin{aligned}
\frac{1}{2}\left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)}\right\|^2 &= \sum_{i=1}^{N} \frac{1}{2}\left\|\boldsymbol{A}_i^{(r,u-1)} - \boldsymbol{A}_i^{(r,u)}\right\|^2 && \text{(D.11)}\\
&= \sum_{i=1}^{N}\left(1 - \boldsymbol{A}_i^{(r,u-1)\mathsf{T}} \boldsymbol{A}_i^{(r,u)}\right) && \text{(D.12)}\\
&= \sum_{i=1}^{N}\left(1 - \frac{1}{\sqrt{1 + \left\|\beta^p \alpha \boldsymbol{Z}_i^{(r,u-1)}\right\|^2}}\right) && \text{(D.13)}\\
&\leq \sum_{i=1}^{N}\left\|\beta^p \alpha \boldsymbol{Z}_i^{(r,u-1)}\right\|^2 && \text{(D.14)}\\
&\leq \frac{1}{c}\left[g\left(\boldsymbol{A}^{(r,u-1)}\right) - g\left(\boldsymbol{A}^{(r,u)}\right)\right], && \text{(D.15)}
\end{aligned}$$

where (D.12) is because $\boldsymbol{A}_i^{(r,u-1)}$ and $\boldsymbol{A}_i^{(r,u)}$ are unit norm vectors, and (D.13) is a direct consequence of (D.9) and (D.10). (D.14) is due to the fact that $x^2+1/\sqrt{1+x^2}-1 \geq 0$ for all $x \in \mathbb{R}$, and (D.15) follows immediately from (6.15). Thus, the proof is complete. □

Lemma D.2 (Subgradient of δ_{norm})**.** *For any matrix* $\boldsymbol{A} \in \mathbb{O} \subset \mathbb{R}^{m\times N}$*.*

$$\partial\delta_{norm}\left(\boldsymbol{A}\right) = \left\{ \boldsymbol{A}\tilde{\boldsymbol{L}}, \tilde{\boldsymbol{L}} \in \mathbb{R}^{N\times N} : \substack{\boldsymbol{L}_{ii}\geq 0, \forall\, i \\ \tilde{\boldsymbol{L}}_{ij}=0, i\neq j} \right\}. \tag{D.16}$$

Proof. Let $\boldsymbol{Z} \in \partial\delta_{\text{norm}}\left(\boldsymbol{A}\right)$. From the definition of the subgradient, we get $\delta_{\text{norm}}\left(\boldsymbol{A}\right) + \mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\left(\boldsymbol{B}-\boldsymbol{A}\right)\right\} \leq \delta_{\text{norm}}\left(\boldsymbol{B}\right)$, $\forall \boldsymbol{B} \in \mathbb{R}^{m\times N}$. This relation is trivially satisfied for all $\boldsymbol{Z}$ and for any $\boldsymbol{B} \notin \mathbb{O}$. However, when $\boldsymbol{B} \in \mathbb{O}$, $\boldsymbol{Z}$ should satisfy

$$\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{B}\right\} \leq \mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{A}\right\}, \tag{D.17}$$

since $\delta_{\text{norm}}\left(\boldsymbol{A}\right) = \delta_{\text{norm}}\left(\boldsymbol{B}\right)$.

To prove the result, we consider three different cases that cover all possible values for $\boldsymbol{Z}$.

1. We express the columns of the matrix $\boldsymbol{Z}$ as $\boldsymbol{Z}_i = \tilde{\boldsymbol{L}}_{ii}\boldsymbol{A}_i + \boldsymbol{A}_i^{\perp}$, where $\tilde{\boldsymbol{L}}_{ii} \in \mathbb{R}$ and $\boldsymbol{A}_i^{\perp} \in \mathbb{R}^m$ is such that $\boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{A}_i^{\perp} = 0$, $\forall i$. Suppose $\boldsymbol{A}_i^{\perp} \neq \boldsymbol{0}$ for at least one value of i. Also, let $\boldsymbol{B} \in \mathbb{R}^{m\times N} \in \mathbb{O}$ be defined as

$$\boldsymbol{B}_i = \begin{cases} \boldsymbol{e}, & \text{for } \left\|\boldsymbol{Z}_i\right\| = 0 \\ \boldsymbol{Z}_i/\left\|\boldsymbol{Z}_i\right\|, & \text{for } \left\|\boldsymbol{Z}_i\right\| \neq 0, \end{cases} \tag{D.18}$$

where $\boldsymbol{e}$ is any unit norm vector. Then,

$$\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{A}\right\} = \sum_{i=1}^{N}\tilde{\boldsymbol{L}}_{ii} < \sum_{i=1}^{N}\left\|\boldsymbol{Z}_i\right\| = \mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{B}\right\}. \tag{D.19}$$

Therefore, there exists a matrix $\boldsymbol{B} \in \mathbb{O}$ such that (D.17) is not satisfied. Thus, we

get

$$\partial\delta_{\text{norm}}(\boldsymbol{A}) \subseteq \left\{\boldsymbol{A}\tilde{\boldsymbol{L}}, \tilde{\boldsymbol{L}} \in \mathbb{R}^{N\times N} : \tilde{\boldsymbol{L}}_{ij} = 0, \text{ if } i \neq j\right\}. \tag{D.20}$$

2. Let $\boldsymbol{Z} = \boldsymbol{A}\tilde{\boldsymbol{L}}$ for some diagonal matrix such that at least one of the diagonal entries of $\tilde{\boldsymbol{L}}$ is negative. Let $\boldsymbol{B} \in \mathbb{R}^{m\times N} \in \mathbb{O}$ be defined such that $\boldsymbol{B}_i = \text{sign}\left\{\tilde{\boldsymbol{L}}_{ii}\right\}\boldsymbol{A}_i$, where the function sign$\{\cdot\}$ takes values 1 and -1 for nonnegative and negative arguments, respectively. Then,

$$\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{A}\right\} = \sum_{i=1}^{N}\tilde{\boldsymbol{L}}_{ii} < \sum_{i=1}^{N}\left|\tilde{\boldsymbol{L}}_{ii}\right| \leq \mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{B}\right\}, \tag{D.21}$$

Therefore, (D.17) does not hold for $\boldsymbol{B} \in \mathbb{O}$, and from (D.20) we get

$$\partial\delta_{\text{norm}}(\boldsymbol{A}) \subseteq \left\{\boldsymbol{A}\tilde{\boldsymbol{L}}, \tilde{\boldsymbol{L}} \in \mathbb{R}^{N\times N} : \substack{\boldsymbol{L}_{ii}\geq 0 \\ \tilde{\boldsymbol{L}}_{ij}=0, \text{ if } i\neq j}\right\}. \tag{D.22}$$

3. Let $\boldsymbol{Z} = \boldsymbol{A}\tilde{\boldsymbol{L}}$, for some diagonal positive semidefinite (psd) matrix $\tilde{\boldsymbol{L}}$. Here, for any matrix $\boldsymbol{B} \in \mathbb{O}$,

$$\mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{B}\right\} = \mathsf{Tr}\left\{\tilde{\boldsymbol{L}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{B}\right\} = \sum_{i=1}^{N}\tilde{\boldsymbol{L}}_{ii}\boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{B}_i \tag{D.23}$$

$$\leq \sum_{i=1}^{N}\tilde{\boldsymbol{L}}_{ii} = \sum_{i=1}^{N}\tilde{\boldsymbol{L}}_{ii}\boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{A}_i \tag{D.24}$$

$$= \mathsf{Tr}\left\{\tilde{\boldsymbol{L}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right\} = \mathsf{Tr}\left\{\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{A}\right\}. \tag{D.25}$$

Therefore, from (D.22) we get

$$\partial\delta_{\text{norm}}(\boldsymbol{A}) = \left\{\boldsymbol{A}\tilde{\boldsymbol{L}}, \tilde{\boldsymbol{L}} \in \mathbb{R}^{N\times N} : \substack{\boldsymbol{L}_{ii}\geq 0, \forall i \\ \tilde{\boldsymbol{L}}_{ij}=0, \text{ otherwise.}}\right\}. \tag{D.26}$$

Hence, the proof is complete. □

D.3.1 Proof of Theorem 6.1

Proof. In [161, Theorem 2], the authors provide a Kurdyka-Łojasiewicz property based proof of convergence of a proximal algorithm. By careful examination their proof, it can be shown that a bounded sequence of iterates converges to a stationary point of $\tilde{g}$ if the following four conditions hold: [1]

(i) The objective function $\tilde{g}(\boldsymbol{A})$ satisfies

$$\inf_{\boldsymbol{A}\in\mathbb{R}^{m\times N}} \tilde{g}\left(\boldsymbol{A}\right) > -\infty. \tag{D.27}$$

(ii) There exist constants $\theta \in [0,1)$, $C, \epsilon > 0$ such that

$$\left|\tilde{g}\left(\boldsymbol{A}\right) - \tilde{g}\left(\boldsymbol{A}^*\right)\right|^{\theta} \leq C \left\|\boldsymbol{Z}\right\| \tag{D.28}$$

for any stationary point $\boldsymbol{A}^*$ of $\tilde{g}$, any $\boldsymbol{A}$ such that $\|\boldsymbol{A} - \boldsymbol{A}^*\| \leq \epsilon$, and any $\boldsymbol{Z}$ such that $\boldsymbol{Z} \in \partial g\left(\boldsymbol{A}\right)$. The constant θ is called the *Łojasiewicz exponent* of the Łojasiewicz gradient inequality.

(iii) There exists $C_1 > 0$ such that

$$\tilde{g}\left(\boldsymbol{A}^{(r,u-1)}\right) - \tilde{g}\left(\boldsymbol{A}^{(r,u)}\right) \geq C_1 \left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)}\right\|^2 \tag{D.29}$$

(iv) There exist $u_0 > 1$, $C_2 > 0$ and $\boldsymbol{Z} \in \partial g\left(\boldsymbol{A}^{(r,u)}\right)$ such that for all $u > u_0$

$$\left\|\boldsymbol{Z}\right\| \leq C_2 \left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)}\right\|. \tag{D.30}$$

[1] A more detailed version of the proof precisely connecting it to the result in [191] is given in Appendix D.12.

Here, the first two conditions are on the cost function, and the last two are on the iterates. In [161, Theorem 2], these conditions are verified to hold for the proximal algorithm. The rest of the proof below is the verification of the four conditions for the ALS procedure.

As discussed in Appendix C (see (D.3)), the cost function g is bounded from below. Therefore, $\tilde{g}$ is also bounded from below, and hence assumption (i) is satisfied.

Next, we note that $\delta_{\text{norm}}(\cdot)$ is an indicator function of a semi-algebraic set, and g is a real analytic function. Therefore, $\tilde{g}$ is a sum of real analytic and semi-algebraic functions. Thus, from [192, Section 2.2], it can be shown that $\tilde{g}$ satisfies the desired condition (ii).

Assumption (iii) follows from Lemma D.1.

Finally, to verify assumption (iv), we first compute the subgradient of the function $\tilde{g}$ using Lemma D.2. Hence, the desired condition is true if and only if, for all $u > u_0$, it holds that

$$\min_{\tilde{\boldsymbol{Z}} \in \partial \tilde{g}\left(\boldsymbol{A}^{(r,u)}\right)} \left\| \tilde{\boldsymbol{Z}} \right\| \leq C_2 \left\| \boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)} \right\|. \tag{D.31}$$

Now, from Lemma D.2, we have,

$$\min_{\tilde{\boldsymbol{Z}} \in \partial \tilde{g}(\boldsymbol{A})} \left\| \tilde{\boldsymbol{Z}} \right\|^2 = \min_{\tilde{\boldsymbol{L}}_{ii} \geq 0} \left\| \nabla g(\boldsymbol{A}) + \boldsymbol{A}\tilde{\boldsymbol{L}} \right\|^2. \tag{D.32}$$

Since the optimization problem is separable in the diagonal entries of $\tilde{\boldsymbol{L}}$, we get the optimum value $\tilde{\boldsymbol{L}}^*$ as

$$\tilde{\boldsymbol{L}}^*_{ii} = \begin{cases} -\boldsymbol{A}_i^{\mathsf{T}} \nabla g\left(\boldsymbol{A}\right)_i, & \text{if } \boldsymbol{A}_i^{\mathsf{T}} \nabla g\left(\boldsymbol{A}\right)_i \leq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{D.33}$$

for $i = 1, 2, \ldots, N$. This gives

$$\underset{\tilde{\boldsymbol{Z}} \in \partial \tilde{g}(\boldsymbol{A})}{\arg\min} \left\| \tilde{\boldsymbol{Z}} \right\| \leq \sqrt{\sum_{i=1}^{N} \max \left\{ \left\| \left(\boldsymbol{I} - \boldsymbol{A}_i \boldsymbol{A}_i^{\mathsf{T}} \right) \nabla g \left(\boldsymbol{A} \right)_i \right\|, \left\| \nabla g \left(\boldsymbol{A} \right)_i \right\| \right\}} \tag{D.34}$$

$$= \left\| \nabla g \left(\boldsymbol{A} \right) \right\|. \tag{D.35}$$

$$\tag{D.36}$$

Here, (D.35) follows from the fact that $\boldsymbol{I} - \boldsymbol{A}_i \boldsymbol{A}_i^{\mathsf{T}}$ is the projection matrix for the subspace orthogonal to the unit norm column $\boldsymbol{A}_i$. Therefore, $\left\| \left(\boldsymbol{I} - \boldsymbol{A}_i \boldsymbol{A}_i^{\mathsf{T}} \right) \nabla g \left(\boldsymbol{A} \right)_i \right\| \leq \left\| \nabla g \left(\boldsymbol{A} \right)_i \right\|$. Thus, we have

$$\min_{\tilde{\boldsymbol{Z}} \in \partial \tilde{g}\left(\boldsymbol{A}^{(r,u)} \right)} \left\| \tilde{\boldsymbol{Z}} \right\| = \left\| \left(\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)} \right) \left(\boldsymbol{\Sigma} - \mathcal{D} \left\{ \boldsymbol{\Sigma} \right\} \right) \right\| \tag{D.37}$$

$$\leq C_2 \left\| \left(\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)} \right) \right\|, \tag{D.38}$$

where C_2 is the spectral norm of $\boldsymbol{\Sigma} - \mathcal{D} \left\{ \boldsymbol{\Sigma} \right\}$. Also, (D.37) is due to the definition of g in (6.10). Hence, assumption (iv) is satisfied for all u. Therefore, all four conditions are met, and consequently, the convergence is guaranteed. □

D.4 Proof of Proposition 6.3

Proof. From (6.15) and Proposition 5.2,

$$\boldsymbol{0} = \lim_{u \to \infty} \boldsymbol{Z}^{(r,u)} = P_{\boldsymbol{A}^{(r)}} \left(\boldsymbol{Y} \boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}^{(r)} \boldsymbol{\Sigma} \right). \tag{D.39}$$

Thus, (6.18) gives

$$\boldsymbol{Y} \boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}^{(r)} \left(\boldsymbol{\Sigma} - \mathcal{D} \left\{ \boldsymbol{\Sigma} \right\} \right) = \boldsymbol{A}^{(r)} \boldsymbol{L}, \tag{D.40}$$

for some diagonal matrix $\boldsymbol{L}$. Then, the result related to the Nash equilibrium point follows from Corollary 6.1. Further, we have

$$\nabla g\left(\boldsymbol{A}^{(r)}\right) = -\boldsymbol{A}^{(r)}\boldsymbol{L}. \tag{D.41}$$

Let $\boldsymbol{\Delta} = \boldsymbol{A} - \boldsymbol{A}^{(r)}$, where $\boldsymbol{A}$ is any matrix in $\mathbb{O}$. Then, for $i = 1, 2, \ldots, N$ we have

$$1 = \|\boldsymbol{A}_i\|^2 = \left\|\boldsymbol{\Delta}_i + \boldsymbol{A}_i^{(r)}\right\|^2 = \|\boldsymbol{\Delta}_i\|^2 + 1 + 2\boldsymbol{\Delta}_i^{\mathsf{T}}\boldsymbol{A}_i^{(r)}. \tag{D.42}$$

Thus, we get $\frac{1}{2}\|\boldsymbol{\Delta}_i\|^2 = -\boldsymbol{\Delta}_i^{\mathsf{T}}\boldsymbol{A}_i^{(r)}$, and similarly, expanding $\|\boldsymbol{A}_i - \boldsymbol{\Delta}_i\|^2$, we get that $\frac{1}{2}\|\boldsymbol{\Delta}_i\|^2 = \boldsymbol{\Delta}_i^{\mathsf{T}}\boldsymbol{A}_i$. Therefore,

$$\mathcal{D}\left\{\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{A}\right\} = -\mathcal{D}\left\{\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{A}^{(r)}\right\} = \frac{1}{2}\mathcal{D}\left\{\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{\Delta}\right\}. \tag{D.43}$$

Now, using Taylor series expansion around $\boldsymbol{A}^{(r)}$, we have

$$g\left(\boldsymbol{A}\right) - g\left(\boldsymbol{A}^{(r)}\right) = \mathsf{Tr}\left\{\boldsymbol{\Delta}^{\mathsf{T}}\nabla g\left(\boldsymbol{A}^{(r)}\right) + \frac{1}{2}\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{\Delta}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\right\} \tag{D.44}$$

$$= \mathsf{Tr}\left\{-\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{A}^{(r)}\boldsymbol{L} + \frac{1}{2}\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{\Delta}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\right\} \tag{D.45}$$

$$= \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{\Delta}\boldsymbol{L} + \boldsymbol{\Delta}^{\mathsf{T}}\boldsymbol{\Delta}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\right\} \tag{D.46}$$

$$= \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{\Delta}\left(\boldsymbol{L} + \boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\boldsymbol{\Delta}^{\mathsf{T}}\right\}, \tag{D.47}$$

where we use (D.41) and (D.43) to get (D.45) and (D.46) respectively. Note that the Taylor series expansion is not an approximation here, as our cost function is quadratic. The right hand side of (D.47) is non-negative if and only if $\boldsymbol{L} + \boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}$ is positive semi-definite, and strictly positive if and only if $\boldsymbol{L} + \boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}$ is positive definite. Hence, the proof is complete. □

D.5 Proof of Theorem 6.2

Proof. The first part of the result directly follows from Proposition 5.2 and [151, Theorem 4.4.1].

To prove the second part, suppose that $\boldsymbol{A}^{(r)}$ is a strict local minimum. Then, for any neighborhood $\mathcal{U}$ of $\boldsymbol{A}^{(r)}$, there exists $\epsilon > 0$ such that, in the closed ball $\mathcal{H}_\epsilon \subseteq \mathcal{U}$ around $\boldsymbol{A}^{(r)}$, $g(\boldsymbol{A}) > g(\boldsymbol{A}^{(r)})$ for all $\boldsymbol{A} \neq \boldsymbol{A}^{(r)} \in \mathcal{H}_\epsilon$. Here, the closed ball is defined as follows:

$$\mathcal{H}_\epsilon = \left\{ \boldsymbol{A} \in \mathbb{O} : \left\| \boldsymbol{A} - \boldsymbol{A}^{(r)} \right\| \leq \epsilon \right\}. \tag{D.48}$$

Moreover, from Lemma D.1, we get

$$\begin{aligned} \left\| G(\boldsymbol{A}) - \boldsymbol{A}^{(r)} \right\| &\leq \| G(\boldsymbol{A}) - \boldsymbol{A} \| + \left\| \boldsymbol{A} - \boldsymbol{A}^{(r)} \right\| && \text{(D.49)} \\ &\leq C_1 \left[g\left(G(\boldsymbol{A}) \right) - g\left(\boldsymbol{A} \right) \right] + \left\| \boldsymbol{A} - \boldsymbol{A}^{(r)} \right\| && \text{(D.50)} \\ &\leq C_1 \left[g\left(\boldsymbol{A} \right) - g\left(\boldsymbol{A}^{(r)} \right) \right] + \left\| \boldsymbol{A} - \boldsymbol{A}^{(r)} \right\|, && \text{(D.51)} \end{aligned}$$

where the last step is because of Proposition 5.2 which gives $g(\boldsymbol{A}) \geq g\left(G(\boldsymbol{A}) \right) \geq g(\boldsymbol{A}^{(r)})$. From Proposition 6.3, we know that $\boldsymbol{A}^{(r)}$ satisfies the relation:

$$\boldsymbol{A}^{(r)} \boldsymbol{L} = \boldsymbol{Y} \boldsymbol{M}^{\mathsf{T}} - \boldsymbol{A}^{(r)} \left(\boldsymbol{\Sigma} - \mathcal{D}\left\{ \boldsymbol{\Sigma} \right\} \right), \tag{D.52}$$

for some diagonal matrix $\boldsymbol{L}$. Following the same steps as (D.45)-(D.47), we get

$$\begin{aligned} 0 < g\left(\boldsymbol{A} \right) - g\left(\boldsymbol{A}^{(r)} \right) &= \frac{1}{2} \mathsf{Tr} \left\{ \left(\boldsymbol{A} - \boldsymbol{A}^{(r)} \right) \left(\boldsymbol{L} + \boldsymbol{\Sigma} - \mathcal{D}\left\{ \boldsymbol{\Sigma} \right\} \right) \left(\boldsymbol{A} - \boldsymbol{A}^{(r)} \right)^{\mathsf{T}} \right\} && \text{(D.53)} \\ &\leq \frac{\lambda_{\max}}{2} \left\| \boldsymbol{A} - \boldsymbol{A}^{(r)} \right\|^2, && \text{(D.54)} \end{aligned}$$

where $\lambda_{\max} > 0$ is the largest singular value of the matrix $\left(\boldsymbol{L} + \boldsymbol{\Sigma} - \mathcal{D}\left\{ \boldsymbol{\Sigma} \right\} \right)$. Thus, from

(D.51),

$$\left\|G(\boldsymbol{A})-\boldsymbol{A}^{(r)}\right\| \leq \frac{C_1\lambda_{\max}}{2}\left\|\boldsymbol{A}-\boldsymbol{A}^{(r)}\right\|^2+\left\|\boldsymbol{A}-\boldsymbol{A}^{(r)}\right\|. \tag{D.55}$$

Let $\epsilon'>0$ be such that

$$\max_{\boldsymbol{A}\in\mathcal{H}_{\epsilon'}}\left\|G(\boldsymbol{A})-\boldsymbol{A}^{(r)}\right\|=\epsilon\leq\left(\frac{C_1\lambda_{\max}}{2}\epsilon'+1\right)\epsilon'. \tag{D.56}$$

Therefore, for all $\boldsymbol{A}\in\mathcal{H}_{\epsilon'}$, $G(\boldsymbol{A})\in\mathcal{H}_{\epsilon}$. Now, using the proof technique used in [151, Theorem 4.4.2], we define the set

$$\mathcal{V}=\{\boldsymbol{A}\in\mathcal{H}_{\epsilon}: g(\boldsymbol{A})<\alpha\}\subseteq\mathcal{H}_{\epsilon}, \tag{D.57}$$

where α is defined as below:

$$\alpha=\begin{cases}\min\limits_{\boldsymbol{B}\in\mathcal{H}_{\epsilon}\backslash\mathcal{H}_{\epsilon'}} G(\boldsymbol{B}) & \epsilon'\leq\epsilon\\ \infty & \epsilon'>\epsilon.\end{cases} \tag{D.58}$$

Note that, when $\epsilon'\leq\epsilon$, $g(\boldsymbol{A})\geq\alpha$, for all $\boldsymbol{A}\in\mathcal{H}_{\epsilon}\backslash\mathcal{H}_{\epsilon'}$. Thus, $\mathcal{V}\subseteq\mathcal{H}_{\epsilon'}$. Also, when $\epsilon'>\epsilon$, $\mathcal{H}_{\epsilon'}\supset\mathcal{H}_{\epsilon'}\supseteq\mathcal{V}$. Therefore, in both cases, $\mathcal{V}\subseteq\mathcal{H}_{\epsilon'}$. Hence, for every $\boldsymbol{A}\in\mathcal{V}$, $G(\boldsymbol{A})\in\mathcal{H}_{\epsilon}$. Further, by Proposition 6.2, the sequence $g\left(G^{(u)}(\boldsymbol{A})\right)$ generated by ALS is nonincreasing, and thus

$$g\left(G(\boldsymbol{A})\right)\leq g\left(\boldsymbol{A}\right)<\alpha. \tag{D.59}$$

Therefore, $G(\boldsymbol{A})\in\mathcal{V}$ for all $\boldsymbol{A}\in\mathcal{V}$, hence $G^{(u)}(\boldsymbol{A})\in\mathcal{V}\subseteq\mathcal{U}$ for all $u\in\mathbb{N}$. Thus, stability of the point is guaranteed. Moreover, since by assumption $\boldsymbol{A}^{(r)}$ is the only critical point (strict local minimum) of g in $\mathcal{V}$, it follows that $\lim_{u\to\infty}G^{(u)}(\boldsymbol{A})=\boldsymbol{A}^{(r)}$, for all $\boldsymbol{A}\in\mathcal{V}$, which shows the asymptotic stability of $\boldsymbol{A}^{(r)}$. This completes the proof. □

D.6 Proof of Proposition 6.4

Proof. We first prove the convergence of the DL-SBL cost function. We note that the AM and the ALS procedures along with the update equations of $\boldsymbol{\Gamma}$ ensure the the following:

$$Q(\boldsymbol{\Lambda}^{(r)};\boldsymbol{\Lambda}^{(r-1)}) \leq Q(\boldsymbol{\Lambda}^{(r-1)};\boldsymbol{\Lambda}^{(r-1)}), \forall r \geq 1. \tag{D.60}$$

This result immediately follows from Proposition 5.2 and the fact that (6.6) maximizes the part of $Q(\boldsymbol{\Lambda};\boldsymbol{\Lambda}^{(r-1)})$ that depends on $\boldsymbol{\Gamma}_k$. Thus, using the properties of EM [193], we have that

$$T(\boldsymbol{\Lambda}^{(r)}) \leq T(\boldsymbol{\Lambda}^{(r-1)}). \tag{D.61}$$

Further, we know that $\left(\sigma^2\boldsymbol{I} + \boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right)^{-1}$ is a positive-definite matrix, and from (6.3),

$$T(\boldsymbol{\Lambda}) \geq \sum_{k=1}^{K} \log\left|\sigma^2\boldsymbol{I} + \boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right| \geq Km\log\sigma^2. \tag{D.62}$$

Therefore, $\left\{T(\boldsymbol{\Lambda}^{(r)})\right\}_{r\in\mathbb{N}}$ is a monotonically decreasing sequence which is bounded from below. Hence, the sequence of DL-SBL cost function value converges. □

D.7 Proof of Theorem 6.3

Proof. The cost function $T(\boldsymbol{\Lambda})$ is a coercive function of $\boldsymbol{\Lambda}$, i.e.,

$$\lim_{\|\boldsymbol{\Lambda}\|\to\infty} T(\boldsymbol{\Lambda}) = \lim_{\|\boldsymbol{\gamma}_k\|\to\infty} \sum_{k=1}^{K} \left(\log\left|\sigma^2\boldsymbol{I} + \boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right| + \boldsymbol{y}_k^{\mathsf{T}}\left(\sigma^2\boldsymbol{I} + \boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{y}_k\right) = \infty. \tag{D.63}$$

This is because $\|\boldsymbol{\Lambda}\| \to \infty$ only if at least one of the entries of $\{\boldsymbol{\gamma}_k\}_{k=1,2,\ldots,K}$ goes to ∞, since $\boldsymbol{A}$ belongs to a bounded set. Further, [155, Theorem 2] shows that the cost

function $T(\boldsymbol{\Lambda})$ is jointly continuous function of $\boldsymbol{\gamma}_k \in \mathbb{R}_+^N, k = 1, 2, \ldots, K$. Using the coerciveness and continuity properties of the cost function, monotonicity of cost function sequence established by Proposition 6.4, and [155, Corollary 1], it follows that the iterates $\left\{\boldsymbol{\gamma}_k^{(r)}\right\}_{r\in\mathbb{N}}$ admit at least one limit point for $k = 1, 2, \ldots, K$. Further, since $\boldsymbol{A} \in \mathbb{O}$ belongs to a bounded set, the iterates $\left\{\boldsymbol{\Lambda}^{(r)}\right\}_{r\in\mathbb{N}}$ also admit at least one limit point.

Next, we use [193, Theorem 1] to prove that the iterates converge to the set of stationary points of the cost function. Therefore, we need to establish two properties of the algorithm:

(i) $T(\boldsymbol{\Lambda}^{(r)}) > T(\boldsymbol{\Lambda}^{(r-1)})$, for all $\boldsymbol{\Lambda} \notin \text{crit}(T)$, where $\text{crit}(T)$ is the set of stationary points of T.

(ii) The point-to-set mapping G which defines algorithm updates: $\boldsymbol{\Lambda}^{(r-1)} = G(\boldsymbol{\Lambda}^{(r)})$, is such that $G(\boldsymbol{\Lambda}^{(r-1)})$ is a closed set over the complement of $\text{crit}(T)$.

Clearly, Condition (i) is satisfied due to Proposition 6.4 and the properties of E and M steps. To prove Condition (ii), we first note that the AM and the ALS algorithm converge to a closed set, as proved by Proposition 6.3. Further, since $T(\boldsymbol{\Lambda})$ is a continuous function of $\boldsymbol{\gamma}_k$, the closed form M-step update of $\boldsymbol{\gamma}_k$ always satisfies Condition (ii) [193, Theorem 2]. Therefore, the algorithm satisfies both conditions, and hence, converges to the set of stationary points.

The last part of the result about the stability directly follows from Proposition 6.4 and [151, Theorem 4.4.1]. □

D.8 Proof of Proposition 6.5

Proof. Under noiseless condition, the dictionary learning problem reduces to a matrix factorization problem: $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$. Suppose that $\boldsymbol{X}$ is already known to the algorithm.

Then, to uniquely estimate $\boldsymbol{A}$, the condition (6.26) is necessary. Similarly, when $\boldsymbol{A}$ is known to the algorithm, to uniquely estimate the sparse $\boldsymbol{X}$, (6.27) is satisfied. This is because, if the condition is not satisfied, there exists an s sparse vector $\boldsymbol{z}$ in the null space of $\boldsymbol{A}$ such that $\boldsymbol{z} + \boldsymbol{x}_k$ is $s-$sparse for some k, and $\boldsymbol{y}_k = \boldsymbol{A}(\boldsymbol{z} + \boldsymbol{x}_k)$. Thus, the solution is not unique.Also, we observe that for $\boldsymbol{X}$ to have full rank, at least two columns of $\boldsymbol{X}$ must have different supports. Therefore, if $|\mathcal{S}_k| = m$, uniqueness is not guaranteed, and thus we get the condition that $|\mathcal{S}_k| < m$. Thus, the first part of the result is obtained.

Next, consider the special case of $\max_{k=1,2,\ldots,K} \|\boldsymbol{x}_k\|_0 = 1$. Then, every nonzero measurement vector is a scaled version of some column of the measurement matrix. The condition (6.26) guarantees that there is no all-zero row in $\boldsymbol{X}$ and thus, there exists a measurement vector $\boldsymbol{y}_k$ corresponding to every column $\boldsymbol{A}_i$ of the dictionary such that $\boldsymbol{y}_k = \boldsymbol{X}_{ik}\boldsymbol{A}_i$ where $\boldsymbol{X}_{ik}$ is the only nonzero entry of the k^{th} column of $\boldsymbol{X}$. Further, by assumption, the columns of $\boldsymbol{A}$ are unit norm, and hence, given $\boldsymbol{y}_k$, the tuple $(\boldsymbol{X}_{ik}, \boldsymbol{A}_i)$ is unique upto the sign of $\boldsymbol{X}_{ik}$. Thus, the solution is unique under (6.26) and (6.27). □

D.9 Proof of Theorem 6.4

Proof. The proof is adapted from the proof of [23, Theorem 1]. The cost function T in (6.3) consists of two terms: the logarithm of the determinant of the product of matrices of the form $\sigma^2\boldsymbol{I} + \boldsymbol{A}^*\boldsymbol{\Gamma}_k^*\boldsymbol{A}^{*\mathsf{T}}$, and sum of projections of the inverses of the same matrices. Since the second term is positive, the minimum is achieved when the first term goes to minus infinity while maintaining some finite upper bound on the second term. We note

that, from (6.27)

$$\mathsf{Rank}\left\{\boldsymbol{\Gamma}_k^*\right\} = \left\|\mathsf{Diag}\left\{\boldsymbol{\Gamma}_k^*\right\}\right\|_0 < \frac{1}{2}\mathrm{Spark}\left\{\boldsymbol{A}^*\right\} \leq \frac{m+1}{2} \leq m. \tag{D.64}$$

Further, we get that

$$\lim_{\sigma^2\to 0}\left|\sigma^2\boldsymbol{I} + \boldsymbol{A}^*\boldsymbol{\Gamma}_k^*\boldsymbol{A}^{*\mathsf{T}}\right| \leq \lim_{\sigma^2\to 0}(\hat{\lambda}_{max}+\sigma^2)^{\mathsf{Rank}\left\{\boldsymbol{\Gamma}_k^*\right\}}(\sigma^2)^{m-\mathsf{Rank}\left\{\boldsymbol{\Gamma}_k^*\right\}} = 0, \tag{D.65}$$

where $\hat{\lambda}_{max}$ is the largest eigenvalue of $\boldsymbol{A}^*\boldsymbol{\Gamma}_k^*\boldsymbol{A}^{*\mathsf{T}}$. Thus, the first term goes to minus infinity. Using arguments similar to those in [23, Theorem 1], we can show that

$$\lim_{\sigma^2\to 0}\boldsymbol{y}_k^{\mathsf{T}}\left(\sigma^2\boldsymbol{I} + \boldsymbol{A}^*\boldsymbol{\Gamma}_k^*\boldsymbol{A}^{*\mathsf{T}}\right)^{-1}\boldsymbol{y}_k \leq \frac{1}{c}\left\|\boldsymbol{x}_k^*\right\|^2. \tag{D.66}$$

Thus, the second term in the cost function is upper bounded by $\frac{1}{c}\left\|\boldsymbol{X}^*\right\|_F^2 < \infty$. Hence, $\left(\boldsymbol{A}^*, \{\boldsymbol{\Gamma}_k^*\}_{k=1}^K\right)$ achieves global minimum. Further, it is easy to see that the cost function takes the same value over the set $\left(\boldsymbol{A}^*\boldsymbol{P}, \{\boldsymbol{P}\boldsymbol{\Gamma}_k^*\boldsymbol{P}\}_{k=1}^K\right)$, and thus the result is proved. □

D.10 Proof of Theorem 6.5

Proof. It is easy to see that the goal of DL-SBL is to solve the optimization problem:

$$\min_{\boldsymbol{A}\in\mathbb{O}}\left[\sum_{k=1}^K \min_{\boldsymbol{\gamma}_k\in\mathbb{R}_+^N} \log\left|\sigma^2\boldsymbol{I} + \boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right| + \boldsymbol{y}_k^{\mathsf{T}}\left(\sigma^2\boldsymbol{I} + \boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{y}_k\right]. \tag{D.67}$$

For any given $\boldsymbol{A}$, the local minima of the objective function of the sub-optimization problem within the square brackets is at most $m-$sparse [23, Theorem 2]. Hence, the local minima of the DL-SBL cost function are all at most $m-$sparse. □

D.11 Derivation of DL-SBL Algorithm

In this section, we provide the details of the EM-algorithm development, explaining how to obtain (6.3)-(6.10), and the $\boldsymbol{\gamma}_k$ update equations in Algorithm 3 and Algorithm 4. The EM algorithm computes the unknown parameter set $\boldsymbol{\Lambda}$ by minimizing the negative log likelihood $-\log p(\boldsymbol{y}^K;\boldsymbol{\Lambda})$. To compute the likelihood, we first note that the SBL framework imposes a Gaussian prior on the unknown vector $\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{\Gamma}_k)$, where $\boldsymbol{\Gamma}_k$ is an unknown diagonal matrix.. Thus, $\boldsymbol{y}_k$ also follows a Gaussian distribution: $\boldsymbol{y}_k \sim \mathcal{N}(\boldsymbol{0},\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}})$ because the noise term $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0},\sigma^2\boldsymbol{I})$. Therefore, we have

$$p(\boldsymbol{y}^K;\boldsymbol{\Lambda}) = \prod_{k=1}^{K} \frac{1}{\sqrt{(2\pi)^m \left|\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right|}} \exp\left(-\frac{1}{2}\boldsymbol{y}_k^{\mathsf{T}}\left(\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{y}_k\right). \quad \text{(D.68)}$$

Hence, the negative log likelihood is computed as follows:

$$-\log p(\boldsymbol{y}^K;\boldsymbol{\Lambda}) = \frac{1}{2}\sum_{k=1}^{K}\Big[m\log(2\pi)+\log\left|\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right| + \frac{1}{2}\boldsymbol{y}_k^{\mathsf{T}}\left(\sigma^2\boldsymbol{I}+\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{y}_k\Big]. \quad \text{(D.69)}$$

Since the $\log(2\pi)$ term is a constant independent of $\boldsymbol{\Lambda}$, we omit that term and the scaling factor of $\frac{1}{2}$ to obtain the cost function $T(\boldsymbol{\Lambda})$ in (6.3).

The EM algorithm treats the unknowns $\boldsymbol{x}^K$ as the hidden data and the observations $\boldsymbol{y}^K$ as the known data. It is an iterative procedure which updates the estimate of the parameters $\boldsymbol{\Lambda}$ in every iteration using two steps: an expectation step (E-step) and a maximization step (M-step). Let $\boldsymbol{\Lambda}^{(r)}$ be the estimate of $\boldsymbol{\Lambda}$ at the r^{th} iteration. The E-step computes the marginal log-likelihood of the observed data $Q\left(\boldsymbol{\Lambda};\boldsymbol{\Lambda}^{(r-1)}\right)$, and the M-step computes

the parameter tuple $\mathbf{\Lambda}$ that maximizes $Q\left(\mathbf{\Lambda}; \mathbf{\Lambda}^{(r-1)}\right)$.

$$\begin{aligned} \textbf{E-step: } Q\left(\mathbf{\Lambda}; \mathbf{\Lambda}^{(r-1)}\right) &= \mathbb{E}_{\boldsymbol{x}^K|\boldsymbol{y}^K;\mathbf{\Lambda}^{(r-1)}}\left\{\log p\left(\boldsymbol{y}^K, \boldsymbol{x}^K; \mathbf{\Lambda}\right)\right\} \\ \textbf{M-step: } \mathbf{\Lambda}^{(r)} &= \underset{\mathbf{\Lambda}\in\mathbb{O}\times\mathbb{R}_+^{NK}}{\arg\max}\, Q\left(\mathbf{\Lambda}; \mathbf{\Lambda}^{(r-1)}\right). \end{aligned} \tag{D.70}$$

To simplify $Q\left(\mathbf{\Lambda}, \mathbf{\Lambda}^{(r-1)}\right)$, we note that

$$p\left(\boldsymbol{y}^K, \boldsymbol{x}^K; \mathbf{\Lambda}\right) = \prod_{k=1}^{K} p\left(\boldsymbol{y}_k|\boldsymbol{x}_k; \mathbf{\Lambda}\right) p\left(\boldsymbol{x}_k; \mathbf{\Lambda}\right). \tag{D.71}$$

Here, $p\left(\boldsymbol{y}_k|\boldsymbol{x}_k; \mathbf{\Lambda}\right) = \mathcal{N}(\boldsymbol{A}\boldsymbol{x}_k, \sigma^2\boldsymbol{I})$, and $p\left(\boldsymbol{x}_k; \mathbf{\Lambda}\right) = \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}_k)$. Thus, we get,

$$\begin{aligned} \log p\left(\boldsymbol{y}^K, \boldsymbol{x}^K; \mathbf{\Lambda}\right) &= \log\left\{\prod_{k=1}^{K} \frac{1}{\sqrt{(2\pi\sigma)^{2m}}} \exp\left(-\frac{1}{2\sigma^2}\left\|\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{x}_k\right\|^2\right)\right. \\ &\qquad \left.\times\ \frac{1}{\sqrt{(2\pi)^N\left|\mathbf{\Gamma}_k\right|}} \exp\left(-\frac{1}{2}\boldsymbol{x}_k^{\mathsf{T}}\mathbf{\Gamma}_k^{-1}\boldsymbol{x}_k\right)\right\} \qquad \text{(D.72)} \\ &= -\frac{Km}{2}\log((2\pi)^{N+1}\sigma^2) - \frac{1}{2}\sum_{k=1}^{K}\left[\log\left|\mathbf{\Gamma}_k\right| + \mathsf{Tr}\left\{\mathbf{\Gamma}_k^{-1}\boldsymbol{x}_k\boldsymbol{x}_k^{\mathsf{T}}\right\}\right] \\ &\qquad - \frac{1}{2\sigma^2}\sum_{k=1}^{K}\left(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{x}_k\right)^{\mathsf{T}}\left(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{x}_k\right). \qquad \text{(D.73)} \end{aligned}$$

Therefore, eliminating the constant terms, we obtain (6.5) as follows:

$$\begin{aligned} Q\left(\mathbf{\Lambda}; \mathbf{\Lambda}^{(r-1)}\right) = -\frac{K}{2}\log(2\pi\sigma^{2m}) - \frac{1}{2}\sum_{k=1}^{K}\left[\log\left|\mathbf{\Gamma}_k\right| + \mathsf{Tr}\left\{\mathbf{\Gamma}_k^{-1}\mathbb{E}\left\{\boldsymbol{x}_k\boldsymbol{x}_k^{\mathsf{T}}|\boldsymbol{y}^K; \mathbf{\Lambda}^{(r-1)}\right\}\right\}\right] \\ - \frac{1}{2\sigma^2}\sum_{k=1}^{K}\mathbb{E}\left\{\left(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{x}_k\right)^{\mathsf{T}}\left(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{x}_k\right)|\boldsymbol{y}^K; \mathbf{\Lambda}^{(r-1)}\right\}. \end{aligned} \tag{D.74}$$

We notice that the expectation terms in the above expression depend only on $\mathbf{\Lambda}^{(r-1)}$, and are independent of $\mathbf{\Lambda}$. Thus, the dependence of $\mathbf{\Gamma}_k$ in $Q\left(\mathbf{\Lambda}; \mathbf{\Lambda}^{(r-1)}\right)$ is only through the k^{th} term in the first summation, and the dependence on $\boldsymbol{A}$ is only through the last

summation term. Therefore, the optimization in the M-step is separable in its variables $\boldsymbol{\Gamma}_k$ and $\boldsymbol{A}$. Hence, the M-step reduces as follows:

$$\boldsymbol{\gamma}_k^{(r)} = \underset{\boldsymbol{\gamma}\in\mathbb{R}_+^N}{\arg\min} \log|\boldsymbol{\Gamma}_k| + \mathsf{Tr}\left\{\boldsymbol{\Gamma}_k^{-1}\mathbb{E}\left\{\boldsymbol{x}_k\boldsymbol{x}_k^{\mathsf{T}}|\boldsymbol{y}^k;\boldsymbol{\Lambda}^{(r-1)}\right\}\right\} \tag{D.75}$$

$$\boldsymbol{A}^{(r)} = \underset{\boldsymbol{A}\in\mathbb{O}}{\arg\min} \sum_{k=1}^{K}\mathbb{E}\Big\{(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{x}_k)^{\mathsf{T}}(\boldsymbol{y}_k-\boldsymbol{A}\boldsymbol{x}_k)\,|\boldsymbol{y}^k;\boldsymbol{\Lambda}^{(r-1)}\Big\}. \tag{D.76}$$

Here, we note that (D.76) is same as (6.9). Further, differentiating the objective function, we get the update equation (6.6):

$$\boldsymbol{\gamma}_k^{(r)} = \mathsf{Diag}\left\{\mathbb{E}\left\{\boldsymbol{x}_k\boldsymbol{x}_k^{\mathsf{T}}|\boldsymbol{y}^k;\boldsymbol{\Lambda}^{(r-1)}\right\}\right\} \tag{D.77}$$

$$= \mathsf{Diag}\left\{\boldsymbol{\mu}_k\boldsymbol{\mu}_k^{\mathsf{T}} + \boldsymbol{\Sigma}_{(k)}\right\}, \tag{D.78}$$

where we use the following facts:

$$\boldsymbol{\mu}_k \triangleq \mathbb{E}\left\{\boldsymbol{x}_k|\boldsymbol{y}_k;\boldsymbol{\Lambda}^{(r-1)}\right\} \tag{D.79}$$

$$\boldsymbol{\Sigma}_{(k)} \triangleq \mathbb{E}\left\{(\boldsymbol{x}_k-\boldsymbol{\mu}_k)(\boldsymbol{x}_k-\boldsymbol{\mu}_k)^{\mathsf{T}}\,|\boldsymbol{y}_k;\boldsymbol{\Lambda}^{(r-1)}\right\} \tag{D.80}$$

$$= \mathrm{cov}\left\{\boldsymbol{x}_k|\boldsymbol{y}^K;\boldsymbol{\Lambda}^{(r-1)}\right\}. \tag{D.81}$$

Next, we compute the conditional expectations terms needed to find $\boldsymbol{\gamma}_k^{(r)}$. We start with the following cross-covariance matrix:

$$\begin{aligned}\mathbb{E}\left\{\boldsymbol{y}_k\boldsymbol{x}_k^{\mathsf{T}}|\boldsymbol{\gamma}_k,\sigma^2\right\} &= \mathbb{E}\left\{(\boldsymbol{A}\boldsymbol{x}_k+\boldsymbol{w}_k)\boldsymbol{x}_k^{\mathsf{T}}|\boldsymbol{\gamma}_k,\sigma^2\right\}\\ &= \mathbb{E}\left\{\boldsymbol{A}\boldsymbol{x}_k\boldsymbol{x}_k^{\mathsf{T}}|\boldsymbol{\gamma}_k,\sigma^2\right\}\\ &= \boldsymbol{A}\boldsymbol{\Gamma}_k.\end{aligned} \tag{D.82}$$

Thus, the conditional mean and covariance are given as follows:

$$\begin{aligned}
\mathrm{cov}\left\{\boldsymbol{x}_k|\boldsymbol{y}^K;\boldsymbol{\Lambda}\right\} &= \mathbb{E}\left\{\boldsymbol{x}_k\boldsymbol{x}_k^\mathsf{T}|\gamma_k,\sigma^2\right\} - \mathbb{E}\left\{\boldsymbol{x}_k\boldsymbol{y}_k^\mathsf{T}|\gamma_k,\sigma^2\right\} \\
&\qquad \times \mathbb{E}\left\{\boldsymbol{y}_k\boldsymbol{y}_k^\mathsf{T}|\gamma_k,\sigma^2\right\}^{-1}\mathbb{E}\left\{\boldsymbol{y}_k\boldsymbol{x}_k^\mathsf{T}|\gamma_k,\sigma^2\right\} \\
&= \boldsymbol{\Gamma}_k - \boldsymbol{\Gamma}_k\boldsymbol{A}^\mathsf{T}\left(\sigma^2\boldsymbol{I} + \boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^\mathsf{T}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}_k \qquad \text{(D.83)} \\
\mathbb{E}\left\{\boldsymbol{x}_k|\boldsymbol{y}^K;\boldsymbol{\Lambda}\right\} &= \mathbb{E}\left\{\boldsymbol{x}_k|\gamma_k,\sigma^2\right\} + \mathbb{E}\left\{\boldsymbol{x}_k\boldsymbol{y}_k^\mathsf{T}|\gamma_k,\sigma^2\right\} \\
&\qquad \times \mathbb{E}\left\{\boldsymbol{y}_k\boldsymbol{y}_k^\mathsf{T}|\gamma_k,\sigma^2\right\}^{-1}\left(\boldsymbol{y}_k - \mathbb{E}\left\{\boldsymbol{y}_k|\gamma_k,\sigma^2\right\}\right) \\
&= \boldsymbol{\Gamma}_k\boldsymbol{A}^\mathsf{T}\left(\sigma^2\boldsymbol{I} + \boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^\mathsf{T}\right)^{-1}\boldsymbol{y}_k \\
&= \sigma^{-2}\boldsymbol{\Gamma}_k\boldsymbol{A}^\mathsf{T}\left(\boldsymbol{I} - \left(\sigma^2\boldsymbol{I} + \boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^\mathsf{T}\right)^{-1}\boldsymbol{A}\boldsymbol{\Gamma}_k\boldsymbol{A}^\mathsf{T}\right)\boldsymbol{y}_k \\
&= \sigma^{-2}\mathrm{cov}\left\{\boldsymbol{x}_k|\boldsymbol{y}^K;\boldsymbol{\Lambda}\right\}\boldsymbol{A}^\mathsf{T}\boldsymbol{y}_k. \qquad \text{(D.84)}
\end{aligned}$$

Therefore, (D.77), (D.83) and (D.84) together gives the update step for $\boldsymbol{\gamma}_k$ used in Algorithm 3 and Algorithm 4.

Similarly, the optimization problem corresponding the dictionary update (D.76) reduces as follows:

$$\begin{aligned}
&\underset{\boldsymbol{A}\in\mathbb{O}}{\arg\min}\ \sum_{k=1}^{K}\mathbb{E}\left\{\left(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{x}_k\right)^\mathsf{T}\left(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{x}_k\right)\middle|\,\boldsymbol{y}_k;\boldsymbol{\Lambda}^{(r-1)}\right\} \qquad \text{(D.85)} \\
&\quad = \underset{\boldsymbol{A}\in\mathbb{O}}{\arg\min}\ \sum_{k=1}^{K}\mathbb{E}\left\{-\boldsymbol{y}_k^\mathsf{T}\boldsymbol{A}\boldsymbol{x}_k + \frac{1}{2}\boldsymbol{x}_k^\mathsf{T}\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\boldsymbol{x}_k\middle|\,\boldsymbol{y}_k;\boldsymbol{\Lambda}^{(r-1)}\right\} \\
&\quad = \underset{\boldsymbol{A}\in\mathbb{O}}{\arg\min}\ -\mathsf{Tr}\left\{\left(\sum_{k=1}^{K}\boldsymbol{\mu}_k\boldsymbol{y}_k^\mathsf{T}\right)\boldsymbol{A} + \frac{1}{2}\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\mathsf{T}\right\} \\
&\quad = \underset{\boldsymbol{A}\in\mathbb{O}}{\arg\min}\ \mathsf{Tr}\left\{-\boldsymbol{M}\boldsymbol{Y}^\mathsf{T}\boldsymbol{A} + \frac{1}{2}\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\mathsf{T}\right\}. \qquad \text{(D.86)}
\end{aligned}$$

Since $\boldsymbol{A} \in \mathbb{O}$, we can further simplify the second term here as follows:

$$\mathsf{Tr}\left\{\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathsf{T}}\right\} = \sum_{i,j=1;i\neq j}^{N} \boldsymbol{\Sigma}[i,j]\boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{A}_j + \sum_{i=1}^{N} \boldsymbol{\Sigma}[i,i]\boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{A}_i \tag{D.87}$$

$$= \mathsf{Tr}\left\{\boldsymbol{A}\left(\boldsymbol{\Sigma} - \mathcal{D}\left\{\boldsymbol{\Sigma}\right\}\right)\boldsymbol{A}^{\mathsf{T}}\right\} + \sum_{i=1}^{N} \boldsymbol{\Sigma}[i,i]. \tag{D.88}$$

Here, the second term does not depend on $\boldsymbol{A}$, and hence, we remove the term from the objective function to get an equivalent optimization objective function as in (6.10). Thus, the derivation of algorithm development given by (6.3)-(6.10), and the update equations for $\boldsymbol{\gamma}_k$ in Algorithm 3 and Algorithm 4 are completed.

Learning the noise variance

Following a similar approach as the above, we can learn the noise variance σ^2 along with the dictionary $\boldsymbol{A}$ and covariance matrices $\boldsymbol{\Gamma}_k$. If σ^2 is unknown, we have to incorporate its update to the M-step by maximizing the Q function defined in (D.74). Thus, considering the terms that depend on $\sigma 62$, we get

$$\begin{aligned}\left(\sigma^2\right)^{(r)} &= \underset{\sigma^2\in\mathbb{R}_+}{\arg\min}\; Km\log(\sigma^2) + \frac{1}{\sigma^2}\sum_{k=1}^{K}\mathbb{E}\left\{(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{x}_k)^{\mathsf{T}}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{x}_k)\,|\boldsymbol{y}^K;\boldsymbol{\Lambda}^{(r-1)}\right\} \\ &= \frac{1}{Km}\sum_{k=1}^{K}\mathbb{E}\left\{(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{x}_k)^{\mathsf{T}}(\boldsymbol{y}_k - \boldsymbol{A}\boldsymbol{x}_k)\,|\boldsymbol{y}^K;\boldsymbol{\Lambda}^{(r-1)}\right\} \\ &= \frac{1}{Km}\mathsf{Tr}\left\{\boldsymbol{Y}^{\mathsf{T}}\boldsymbol{Y} - 2\boldsymbol{M}\boldsymbol{Y}^{\mathsf{T}}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\mathsf{T}}\right\},\end{aligned} \tag{D.89}$$

where the last step follows because of the same arguments used to derive (D.86) from (D.85).

D.12 Proof of Kurdyka-Łojasiewicz property based Convergence Result

Theorem D.1. *A bounded sequence of iterates* $\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$ *generated by the ALS algorithm converges to a stationary point of* $\tilde{g}$ *if the following four conditions hold:*

(i) The objective function $\tilde{g}(\boldsymbol{A})$ *satisfies*

$$\inf_{\boldsymbol{A}\in\mathbb{R}^{m\times N}} \tilde{g}\left(\boldsymbol{A}\right) > -\infty. \tag{D.90}$$

(ii) There exist constants $\theta \in [0,1)$, $C, \epsilon > 0$ *such that*

$$\left|\tilde{g}\left(\boldsymbol{A}\right) - \tilde{g}\left(\boldsymbol{A}^{*}\right)\right|^{\theta} \leq C \left\|\boldsymbol{Z}\right\| \tag{D.91}$$

for any stationary point $\boldsymbol{A}^{*}$ *of* $\tilde{g}$*, any* $\boldsymbol{A}$ *such that* $\left\|\boldsymbol{A} - \boldsymbol{A}^{*}\right\| \leq \epsilon$*, and any* $\boldsymbol{Z}$ *such that* $\boldsymbol{Z} \in \partial g\left(\boldsymbol{A}\right)$*. The constant* θ *is called the* Łojasiewicz exponent *of the Łojasiewicz gradient inequality.*

(iii) There exists $C_1 > 0$ *such that*

$$\tilde{g}\left(\boldsymbol{A}^{(r,u-1)}\right) - \tilde{g}\left(\boldsymbol{A}^{(r,u)}\right) \geq C_1 \left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)}\right\|^{2} \tag{D.92}$$

(iv) There exist $u_0 > 1$, $C_2 > 0$ *and* $\boldsymbol{Z} \in \partial g\left(\boldsymbol{A}^{(r,u)}\right)$ *such that for all* $u > u_0$

$$\left\|\boldsymbol{Z}\right\| \leq C_2 \left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)}\right\|. \tag{D.93}$$

The proof is adapted from the proof of [161, Theorem 2]. At a high level, there are four steps to the proof:

(A) We first prove that the sequence $\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$ converges to a bounded connected set $\mathbb{G} \subseteq \text{crit}(\tilde{g}) \subseteq \mathbb{O}$, where $\text{crit}(\tilde{g})$ is the set of stationary points of $\tilde{g}$. Moreover, $\tilde{g}$ is constant over the set $\mathbb{G}$.

(B) Next, we connect the above result to Condition (ii). To establish the connection, we define a new function $\bar{g} : \mathbb{O} \to \mathbb{R}_+$ as $\bar{g}(\boldsymbol{A}) \triangleq \tilde{g}(\boldsymbol{A}) - \tilde{g}(\boldsymbol{A}^{(r)})$, where $\boldsymbol{A}^{(r)}$ is a limit point of the sequence $\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$, and $\boldsymbol{A}$ is any point in the set $\mathbb{O}$. We note that the definition of $\bar{g}$ is unambiguous because Step A shows that $\tilde{g}$ is constant over the set $\mathbb{G}$. We then show that there exists a positive integer $U_0 \in \mathbb{N}$ and $\tilde{C} > 0$ such that for all $u \geq U_0$,

$$\left(\bar{g}\left(\boldsymbol{A}^{(r,u)}\right)\right)^{\theta} \geq \tilde{C}\left\|\boldsymbol{Z}\right\|, \tag{D.94}$$

for any $\boldsymbol{Z}$ such that $\boldsymbol{Z} \in \partial\tilde{g}\left(\boldsymbol{A}^{(r,u)}\right)$.

(C) Finally, using the above relation and other conditions of the theorem, we show that the desired result follows.

Next, we present the details of the above steps:

D.12.1 Characterization of $\mathbb{G}$

From Condition (iii), we get that

$$\sum_{u=1}^{\infty}\left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)}\right\|^2 \leq \frac{1}{C_1}\left[\lim_{u\to\infty}\tilde{g}\left(\boldsymbol{A}^{(r,u-1)}\right) - \tilde{g}\left(\boldsymbol{A}^{(r,0)}\right)\right] < \infty, \tag{D.95}$$

where the last step follows because $\lim_{u\to\infty}\tilde{g}\left(\boldsymbol{A}^{(r,u-1)}\right) < \infty$ due to Proposition 5.2. Further, [191, Theorem 1] states that the set of subsequential limit points of a sequence

$\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$ in a compact metric space is a connected set if it satisfies the following:

$$\sum_{u=1}^{\infty}\left\|\boldsymbol{A}^{(r,u-1)}-\boldsymbol{A}^{(r,u)}\right\|^2<\infty. \tag{D.96}$$

Consequently, the result applies to any bounded sequence satisfying (D.96). Since the sequence $\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$ generated by the AM procedure belongs to the bounded set $\mathbb{O}$, it converges to a bounded connected set $\mathbb{G}\subseteq\mathbb{O}$. Also, since the set of subsequential limits is closed, $\mathbb{G}$ is a connected compact set.

Now, for any limit point $\boldsymbol{A}^{(r)}\in\mathbb{G}$ of the sequence $\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$, there exists a sequence $\{u_j\}_{j\in\mathbb{N}}$ of natural numbers such that $\left\{\left(\boldsymbol{A}^{(r,u_j)},\boldsymbol{Z}^{(r,u_j)},\tilde{g}\left(\boldsymbol{A}^{(r,u_j)}\right)\right)\right\}_{j\in\mathbb{N}}$ converges to the tuple $\left(\boldsymbol{A}^{(r)},\boldsymbol{0},\tilde{g}\left(\boldsymbol{A}^{(r)}\right)\right)$. This is because the subsequence $\left\{\left(\boldsymbol{Z}^{(r,u_j)},\tilde{g}\left(\boldsymbol{A}^{(r,u_j)}\right)\right)\right\}_{j\in\mathbb{N}}$ converges to the same limit point as that of the sequence $\left\{\left(\boldsymbol{Z}^{(r,u)},\tilde{g}\left(\boldsymbol{A}^{(r,u)}\right)\right)\right\}_{u\in\mathbb{N}}$ which is $\left(\boldsymbol{0},\tilde{g}\left(\boldsymbol{A}^{(r)}\right)\right)$ due to (6.15) and Proposition 5.2. Therefore, we conclude that $\mathbb{G}\subset\mathrm{crit}(\tilde{g})$ and $\tilde{g}$ is constant over the set $\mathbb{G}$, completing Step A.

D.12.2 Connection to Kurdyka-Łojasiewicz property

The compact set $\mathbb{G}$ can be covered with finite number of closed balls

$$\mathcal{B}_j=\left\{\boldsymbol{A}\in\mathbb{O}:\left\|\boldsymbol{A}-\boldsymbol{A}^{*(j)}\right\|\leq\epsilon_j\right\}, \tag{D.97}$$

such that Condition (ii) is satisfied by $\boldsymbol{A}^{(r,j)}$ with constants $C^{(j)}$ and $\epsilon_j>0$. Therefore, we have the following relation for $\boldsymbol{A}\in\mathcal{B}_j$:

$$\left|\tilde{g}\left(\boldsymbol{A}\right)-\tilde{g}\left(\boldsymbol{A}^{*(j)}\right)\right|^{\theta_j}\leq C^{(j)}\left\|\boldsymbol{Z}\right\|, \tag{D.98}$$

for some θ_j and any $\boldsymbol{Z}$ such that $\boldsymbol{Z} \in \partial\tilde{g}(\boldsymbol{A})$. Setting $\epsilon = \min_j \epsilon_j$, $\tilde{C} = \max_j C^{(j)}$, and $\theta = \max_j \theta_j$ we get the following:

$$|\tilde{g}(\boldsymbol{A}) - \tilde{g}(\boldsymbol{A}^*)|^{\theta} \leq \tilde{C} \|\boldsymbol{Z}\|, \tag{D.99}$$

for any $\boldsymbol{A}^* \in \mathbb{G}$ of $\tilde{g}$, any $\boldsymbol{A}$ such that $\|\boldsymbol{A} - \mathbb{G}\| \leq \epsilon$, and any $\boldsymbol{Z}$ such that $\boldsymbol{Z} \in \partial\tilde{g}(\boldsymbol{A})$. Further, since $\left\{\boldsymbol{A}^{(r,u)}\right\}_{u\in\mathbb{N}}$ converges to $\mathbb{G}$, for any $\epsilon > 0$, there exists a positive integer U_0 such that for all $u \geq U_0$, we have $\left\|\boldsymbol{A}^{(r,u)} - \mathbb{G}\right\| \leq \epsilon$. Therefore, for all $u \geq U_0$,

$$\left|\bar{g}\left(\boldsymbol{A}^{(r,u)}\right)\right|^{\theta} = \left|\tilde{g}\left(\boldsymbol{A}^{(r,u)}\right) - \tilde{g}\left(\boldsymbol{A}^{(r)}\right)\right|^{\theta} \leq \tilde{C}\|\boldsymbol{Z}\|. \tag{D.100}$$

Thus, Step B is completed.

D.12.3 Convergence to a single point

Since $\left\{\tilde{g}\left(\boldsymbol{A}^{(r,u)}\right)\right\}_{u\in\mathbb{N}}$ is a non-increasing sequence, we have $\bar{g}\left(\boldsymbol{A}^{(r,u)}\right) \geq 0$, and the following relation holds.

$$\lim_{u\to\infty} \bar{g}\left(\boldsymbol{A}^{(r,u)}\right) = 0. \tag{D.101}$$

We first note that the function $h : \mathbb{R}_+ \to \mathbb{R}$ defined as $h(s) = -s^{1-\theta}$ is convex for all $0 \leq \theta \leq 1$. Thus, for all $u \in \mathbb{N}$ and for θ in Condition (ii), it holds that

$$\left[\bar{g}\left(\boldsymbol{A}^{(r,u-1)}\right)\right]^{1-\theta} - \left[\bar{g}\left(\boldsymbol{A}^{(r,u)}\right)\right]^{1-\theta} = h\left(\bar{g}\left(\boldsymbol{A}^{(r,u-1)}\right)\right) - h\left(\bar{g}\left(\boldsymbol{A}^{(r,u)}\right)\right) \tag{D.102}$$

$$\geq \left.\frac{dh(s)}{ds}\right|_{s=\bar{g}\left(\boldsymbol{A}^{(r,u-1)}\right)} \left[\bar{g}\left(\boldsymbol{A}^{(r,u-1)}\right) - \bar{g}\left(\boldsymbol{A}^{(r,u)}\right)\right]. \tag{D.103}$$

Further, we have,

$$\left[\bar{g}\left(\boldsymbol{A}^{(r,u-1)}\right)\right]^{1-\theta} - \left[\bar{g}\left(\boldsymbol{A}^{(r,u)}\right)\right]^{1-\theta} = (1-\theta)\left[\bar{g}(\boldsymbol{A}^{(r,u-1)}\right]^{-\theta}\left[\bar{g}\left(\boldsymbol{A}^{(r,u-1)}\right) - \bar{g}\left(\boldsymbol{A}^{(r,u)}\right)\right] \tag{D.104}$$

$$\geq C_1(1-\theta)\left[\bar{g}\left(\boldsymbol{A}^{(r,u)}\right)\right]^{-\theta}\left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u)}\right\|^2, \tag{D.105}$$

where we use Condition (iii) to obtain the last relation. Further, from Step B, we get that

$$\left[\bar{g}\left(\boldsymbol{A}^{(r,u-1)}\right)\right]^{1-\theta} - \left[\bar{g}\left(\boldsymbol{A}^{(r,u)}\right)\right]^{1-\theta} \geq \frac{C_1(1-\theta)}{C}\frac{\left\|\boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)}\right\|^2}{\|\boldsymbol{Z}\|} \tag{D.106}$$

$$\geq \frac{C_1(1-\theta)}{CC_2}\frac{\left\|\boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)}\right\|^2}{\left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u-2)}\right\|}, \tag{D.107}$$

where we use Condition (iv).

Next, we fix a constant $0 < \tau < 1$. For some $u \geq U_0$, if $\left\|\boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)}\right\| \geq \tau\left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u-2)}\right\|$, from (D.107), we get the following:

$$\frac{CC_2}{rC_1(1-\theta)}\left\{\left[\bar{g}\left(\boldsymbol{A}^{(r,u-1)}\right)\right]^{1-\theta} - \left[\bar{g}\left(\boldsymbol{A}^{(r,u)}\right)\right]^{1-\theta}\right\} \geq \left\|\boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)}\right\|. \tag{D.108}$$

For all other values of $u \geq U_0$, we have the following relation:

$$\left\|\boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)}\right\| \leq \tau\left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u-2)}\right\|. \tag{D.109}$$

Combining (D.108) and (D.109), for all $u \geq U_0$, we get the upper bound as given below:

$$\left\|\boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)}\right\| \leq \tau\left\|\boldsymbol{A}^{(r,u-1)} - \boldsymbol{A}^{(r,u-2)}\right\| + \frac{CC_2}{rC_1(1-\theta)}\left\{\left[\bar{g}\left(\boldsymbol{A}^{(r,u-1)}\right)\right]^{1-\theta} - \left[\bar{g}\left(\boldsymbol{A}^{(r,u)}\right)\right]^{1-\theta}\right\}. \tag{D.110}$$

Summing both sides, and using (D.101), we can simplify the expression as follows:

$$\sum_{u=U_0}^{\infty} \left\| \boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)} \right\| \leq \frac{\tau}{1-\tau} \left\| \boldsymbol{A}^{(r,U_0-1)} - \boldsymbol{A}^{(r,U_0-2)} \right\| + \frac{CC_2}{rC_1(1-\theta)} \left[\bar{g} \left(\boldsymbol{A}^{(r,U_0)} \right) \right]^{1-\theta}. \tag{D.111}$$

Thus, we conclude that the series converges, and there exists a finite constant $\kappa < \infty$ such that the following holds:

$$\sum_{u=1}^{\infty} \left\| \boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)} \right\| = \kappa. \tag{D.112}$$

Hence, for any $\epsilon > 0$, there exists a positive integer U_1 such that for all $U \geq U_1$, we have

$$\kappa - \epsilon/2 \leq \sum_{u=1}^{U} \left\| \boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)} \right\| \leq \kappa + \epsilon/2. \tag{D.113}$$

Thus, for any $U_1 \leq u_1 < u_2$, we have

$$\left| \left\| \boldsymbol{A}^{(r,u_2)} \right\| - \left\| \boldsymbol{A}^{(r,u_1)} \right\| \right| \leq \sum_{u=u_1+1}^{u_2} \left| \left\| \boldsymbol{A}^{(r,u)} \right\| - \left\| \boldsymbol{A}^{(r,u-1)} \right\| \right| \leq \sum_{u=u_1+1}^{u_2} \left\| \boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)} \right\| \tag{D.114}$$

$$= \sum_{u=1}^{u_2} \left\| \boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)} \right\| - \sum_{u=1}^{u_1} \left\| \boldsymbol{A}^{(r,u)} - \boldsymbol{A}^{(r,u-1)} \right\| \leq \epsilon. \tag{D.115}$$

Therefore, the sequence $\left\{ \boldsymbol{A}^{(r,u)} \right\}_{u \in \mathbb{N}}$ is Cauchy, hence it converges.

Bibliography

[1] K. Zhou, J. C. Doyle, K. Glover *et al.*, *Robust and optimal control*. Prentice hall New Jersey, 1996, vol. 40.

[2] B. Anderson and J. Moore, *Optimal filtering*. Courier Dover, 2005.

[3] R. Prasad, C. Murthy, and B. Rao, "Joint approximately sparse channel estimation and data detection in OFDM systems using sparse Bayesian learning," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3591–3603, Jul. 2014.

[4] P. J. Brockwell, R. A. Davis, and M. V. Calder, *Introduction to time series and forecasting*. Springer, 2002, vol. 2.

[5] C. A. Pope III, R. T. Burnett, M. J. Thun, E. E. Calle, D. Krewski, K. Ito, and G. D. Thurston, "Lung cancer, cardiopulmonary mortality, and long-term exposure to fine particulate air pollution," *JAMA*, vol. 287, no. 9, pp. 1132–1141, Mar. 2002.

[6] M. Shao, X. Tang, Y. Zhang, and W. Li, "City clusters in China: air and surface water pollution," *Front. Ecol. Environ.*, vol. 4, no. 7, pp. 353–361, Sep. 2006.

[7] G. Neumann, T. Noda, and Y. Kawaoka, "Emergence and pandemic potential of swine-origin H1N1 influenza virus," *Nature*, vol. 459, no. 7249, pp. 931–939, Jun. 2009.

[8] M. Hvistendahl, D. Normile, and J. Cohen, "Despite large research effort, H7N9 continues to baffle," *Science*, vol. 340, no. 6131, pp. 414–415, Apr. 2013.

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

[10] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[11] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.

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

[13] R. Tibshirani, "Regression shrinkage and selection via the lasso," *. J. Roy. Stat. Soc. B*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[14] E. Candes and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *Ann. Stat.*, vol. 35, no. 6, pp. 2313–2351, 2007.

[15] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *J. Fourier Anal.*, vol. 14, no. 5-6, pp. 629–654, Dec. 2008.

[16] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.

[17] S. Qian and D. Chen, "Signal representation using adaptive normalized Gaussian functions," *Signal Process.*, vol. 36, no. 1, pp. 1–11, Mar. 1994.

[18] G. M. Davis, S. G. Mallat, and Z. Zhang, "Adaptive time-frequency decompositions," *Opt. Eng.*, vol. 33, no. 7, pp. 2183–2192, Jul. 1994.

[19] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. ASILOMAR*, Nov. 1993, pp. 40–44.

[20] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Intl. J. Control*, vol. 50, no. 5, pp. 1873–1896, Nov. 1989.

[21] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, May 2009.

[22] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–214, Sep. 2001.

[23] D. Wipf and B. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[24] J. P. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, Oct 2013.

[25] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a Markov-tree prior," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3439–3448, July 2012.

[26] P. Schniter and S. Rangan, "Compressive phase retrieval via generalized approximate message passing," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 1043–1055, Feb 2015.

[27] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[28] ——, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec 2006.

[29] T. T. Cai and A. Zhang, "Sharp RIP bound for sparse signal and low-rank matrix recovery," *CoRR*, vol. abs/1302.1236, 2013. [Online]. Available: http://arxiv.org/abs/1302.1236

[30] ——, "Sparse representation of a polytope and recovery of sparse signals and low-rank matrices," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, pp. 122–132, Jan. 2014.

[31] R. Zhang and S. Li, "A proof of conjecture on restricted isometry property constants $\delta_{tk}\left(0 < t < \frac{4}{3}\right)$," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1699–1705, Mar. 2018.

[32] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Birkhäuser, 2013.

[33] J. Wen, Z. Zhou, J. Wang, X. Tang, and Q. Mo, "A sharp condition for exact support recovery of sparse signals with orthogonal matching pursuit," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2016.

[34] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty

relations and efficient recovery," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.

[35] M. Stojnic, F. Parvaresh, and B. Hassibi, "On the reconstruction of block-sparse signals with an optimal number of measurements," *IEEE Trans. Signal Process.*, vol. 57, no. 8, pp. 3075–3085, Aug. 2009.

[36] K. Li, C. R. Rojas, T. Yang, H. Hjalmarsson, K. H. Johansson, and S. Cong, "Piecewise sparse signal recovery via piecewise orthogonal matching pursuit," in *Proc. ICASSP*, Mar. 2016, pp. 4608–4612.

[37] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2477–2488, Jul. 2005.

[38] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. part I: Greedy pursuit," *IEEE Trans. Signal Process.*, vol. 86, no. 3, pp. 572–588, Mar. 2006.

[39] J. D. Blanchard, M. Cermak, D. Hanle, and Y. Jing, "Greedy algorithms for joint sparse recovery," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1694–1704, Apr. 2014.

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

[41] D. Wipf and B. Rao, "An empirical Bayesian strategy for solving the simultaneous

sparse approximation problem," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3704–3716, Jul. 2007.

[42] R. Zdunek and A. Cichocki, "Improved M-FOCUSS algorithm with overlapping blocks for locally smooth sparse signals," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4752–4761, Oct. 2008.

[43] Z. Zhang and B. D. Rao, "Sparse signal recovery in the presence of correlated multiple measurement vectors," in *Proc. ICASSP*, Mar. 2010.

[44] A. L. Lloyd and R. M. May, "How viruses spread among computers and people?" *Science*, vol. 292, no. 5520, pp. 1316–1317, May 2001.

[45] P. Wang, M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding the spreading patterns of mobile phone viruses," *Science*, vol. 324, no. 5930, pp. 1071–1076, May 2009.

[46] M. Cha, A. Mislove, and K. P. Gummadi, "A measurement-driven analysis of information propagation in the flickr social network," in *Proc. WWW*, Apr. 2009.

[47] Z.-L. Hu, X. Han, Y.-C. Lai, and W.-X. Wang, "Optimal localization of diffusion sources in complex networks," *Roy. Soc. Open Science*, vol. 4, no. 4, p. 170091, Apr. 2017.

[48] M. Nagahara and D. E. Quevedo, "Sparse representations for packetized predictive networked control," *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 84–89, Jan. 2011.

[49] Z. Li, Y. Xu, H. Huang, and S. Misra, "Sparse control and compressed sensing

in networked switched systems," *IET Control Theory & Appl.*, vol. 10, no. 9, pp. 1078–1087, Jun. 2016.

[50] G. Joseph and C. R. Murthy, "On the observability of a linear system with a sparse initial state," *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 994 – 998, Jul. 2018.

[51] ——, "Measurement bounds for observability of linear dynamical systems under sparsity constraints," *IEEE Trans. Signal Process.*, vol. 67, no. 8, pp. 1992–2006, Feb. 2019.

[52] ——, "Controllability of linear dynamical systems under input sparsity constraints," *Submitted, IEEE Trans. Autom. Control*, 2019.

[53] G. Joseph, C. R. Murthy, R. Prasad, and B. D. Rao, "Online recovery of temporally correlated sparse signals using multiple measurement vectors," in *Proc. Globecom*, Dec. 2015.

[54] G. Joseph and C. R. Murthy, "A noniterative online bayesian algorithm for the recovery of temporally correlated sparse vectors," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5510–5525, Oct. 2017.

[55] ——, "A Bayesian algorithm for joint dictionary learning and sparse signal recovery," *Submitted, IEEE Trans. Signal Process.*, 2018.

[56] G. Joseph, A. B. Zoubi, C. R. Murthy, and V. J. Mathews, "Anomaly imaging for structural health monitoring exploiting clustered sparsity," in *Proc. ICASSP*, May 2019.

[57] R. Kalman, "On the general theory of control systems," *IRE Tran. Autom. Control*, vol. 4, no. 3, pp. 110–110, Dec 1959.

[58] C. T. Chen, *Linear System Theory and Design*. Oxford University Press, 1999.

[59] C. Böß, "Using model reduction techniques within the incremental 4D-Var method," Ph.D. dissertation, Universität Bremen, 2008.

[60] M. Verlaan, "Efficient Kalman filtering algorithms for hydrodynamic models," Ph.D. dissertation, Delft University of Technology, 1998.

[61] A. Lawless, N. Nichols, C. Boess, and A. Bunse-Gerstner, "Using model reduction methods within incremental four-dimensional variational data assimilation," *Monthly Weather Review*, vol. 136, no. 4, pp. 1511–1522, Apr. 2008.

[62] M. B. Wakin, B. M. Sanandaji, and T. L. Vincent, "On the observability of linear systems from random, compressive measurements," in *Proc. CDC*, Dec. 2010.

[63] B. M. Sanandaji, M. B. Wakin, and T. L. Vincent, "Technical report: Observability with random observations," *CoRR*, vol. abs/1211.4077, 2013. [Online]. Available: http://arxiv.org/abs/1211.4077

[64] ——, "Observability with random observations," *IEEE Trans. Autom. Control*, vol. 59, no. 11, pp. 3002–3007, Nov. 2014.

[65] K. P. Singh, A. Malik, and S. Sinha, "Water quality assessment and apportionment of pollution sources of Gomti river (India) using multivariate statistical techniques-a case study," *Analytica Chimica Acta*, vol. 538, no. 1, pp. 355 – 374, May 2005.

[66] S. Bhattacharya and T. Basar, "Sparsity based feedback design: A new paradigm in opportunistic sensing," in *Proc. ACC*, Jun. 2011.

[67] W. Dai and S. Yüksel, "Observability of a linear system under sparsity constraints," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2372–2376, Sep. 2013.

[68] S. Sefati, N. J. Cowan, and R. Vidal, "Linear systems with sparse inputs: Observability and input recovery," in *Proc. ACC*, Jul. 2015.

[69] H. Rauhut, "Compressive sensing and structured random matrices," 2011.

[70] M. Rudelson and R. Vershynin, "On sparse reconstruction from Fourier and Gaussian measurements," *Comm. Pure Appl. Math.*, vol. 61, no. 8, pp. 1025–1045, Aug. 2008.

[71] J. Haupt, W. U. Bajwa, G. Raz, and R. Nowak, "Toeplitz compressed sensing matrices with applications to sparse channel estimation," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5862–5875, Nov 2010.

[72] F. Krahmer, S. Mendelson, and H. Rauhut, "Suprema of chaos processes and the restricted isometry property," *Comm. Pure Appl. Math.*, vol. 67, no. 11, pp. 1877–1904, Nov. 2014.

[73] H. Rauhut, J. K. Romberg, and J. A. Tropp, "Restricted isometries for partial random circulant matrices," *Appl. Comput. Harmon. Anal.*, vol. 32, no. 2, pp. 242–254, Mar. 2012.

[74] A. Eftekhari, H. Yap, C. Rozell, and M. Wakin, "The restricted isometry property

for random block diagonal matrices," *Appl. Comput. Harmon. Anal.*, vol. 38, no. 1, pp. 1–31, Jan. 2015.

[75] S. Khanna and C. R. Murthy, "On the restricted isometry of the columnwise Khatri-Rao product," *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1170 – 1183, Dec. 2017.

[76] P. Rigollet, "18.S997: High-Dimensional Statistics, Cambridge, MA, USA:MIT OpenCourseWare," Jul. 2015. [Online]. Available: http://www-math.mit.edu/~rigollet/PDFs/RigNotes15.pdf

[77] Q. Mo and S. Li, "New bounds on the restricted isometry constant δ_{2k}," *Appl. Comput. Harmon. Anal.*, vol. 31, no. 3, pp. 460 – 468, 2011.

[78] T. T. Cai, L. Wang, and G. Xu, "New bounds for restricted isometry constants," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4388–4394, Sep. 2010.

[79] T. Tao and V. Vu, "Random matrices: The distribution of the smallest singular values," *Geom. Funct. Anal.*, vol. 20, no. 1, pp. 260–297, Jun. 2010.

[80] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, and C. Faloutsos, "Epidemic thresholds in real networks," *ACM Trans. Inf. Syst. Security*, vol. 10, no. 4, p. 1, Jan 200-8.

[81] R. Durrett, "Some features of the spread of epidemics and information on a random graph," *Proc. of the National Acad. of Sciences*, Feb. 2010.

[82] ——, *Random graph dynamics*. Cambridge university press Cambridge, 2007.

[83] M. E. Newman, S. Forrest, and J. Balthrop, "Email networks and the spread of computer viruses," *Physical Review E*, vol. 66, no. 3, p. 035101, Sep. 2002.

[84] L. A. Pastur and M. Shcherbina, *Eigenvalue distribution of large random matrices*. American Mathematical Soc., 2011.

[85] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 40–52, Mar. 2014.

[86] S. C. Tatikonda, "Control under communication constraints," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[87] Y.-Y. Liu and A.-L. Barabási, "Control principles of complex systems," *Rev. Mod. Phys.*, vol. 88, no. 3, p. 035006, Sep. 2016.

[88] P. V. Chanekar, N. Chopra, and S. Azarm, "Optimal actuator placement for linear systems with limited number of actuators," in *Proc. ACC*, May 2017, pp. 334–339.

[89] E. Nozari, F. Pasqualetti, and J. Cortés, "Time-invariant versus time-varying actuator scheduling in complex networks," in *Proc. ACC*, May 2017, pp. 4995–5000.

[90] M. Hautus, "Stabilization controllability and observability of linear autonomous systems," in *Indagationes mathematicae (proceedings)*, vol. 73, Jan. 1970, pp. 448–455.

[91] A. Jadbabaie, A. Olshevsky, and M. Siami, "Deterministic and randomized actuator scheduling with guaranteed performance bounds," *arXiv preprint arXiv:1805.00606*, May 2018.

[92] A. Olshevsky, "Minimal controllability problems." *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 3, pp. 249–258, Sep. 2014.

[93] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, "Minimal actuator placement with bounds on control effort," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 1, pp. 67–78, Mar. 2016.

[94] A. Jadbabaie, A. Olshevsky, G. J. Pappas, and V. Tzoumas, "Minimal reachability is hard to approximate," *IEEE Trans. Autom. Control*, vol. 64, no. 2, pp. 783–789, Dec. 2015.

[95] Z. Liu, Y. Long, A. Clark, P. Lee, L. Bushnell, D. Kirschen, and R. Poovendran, "Minimal input selection for robust control," *CoRR*, vol. abs/1712.01232, 2017. [Online]. Available: http://arxiv.org/abs/1712.01232

[96] A. S. Charles, H. L. Yap, and C. J. Rozell, "Short-term memory capacity in networks via the restricted isometry property," *Neural Comput.*, vol. 26, no. 6, pp. 1198–1235, Jun. 2014.

[97] M. Kafashan, A. Nandi, and S. Ching, "Relating observability and compressed sensing of time-varying signals in recurrent linear networks," *Neural Networks*, vol. 83, pp. 11–20, Nov. 2016.

[98] V. Y. Pan and Z. Q. Chen, "The complexity of the matrix eigenproblem," in *Proc. STOC*, May 1999, pp. 507–516.

[99] C.-T. Chen, *Linear system theory and design.* Oxford University Press, Inc., 1998.

[100] R. E. Kalman, "Mathematical description of linear dynamical systems," *SIAM J. Control*, vol. 1, no. 2, pp. 152–192, 1963.

[101] L. C. Westphal, *Handbook of Control Systems Engineering*. Springer Science & Business Media, 2012.

[102] F. E. Hohn, *Elementary matrix algebra*. Courier Corporation, 2013.

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

[104] J. H. G. Ender, "On compressive sensing applied to radar," *Signal Processing*, vol. 90, no. 5, pp. 1402–1414, May 2010.

[105] I. F. Gorodnitsky, J. S. George, and B. D. Rao, "Neuromagnetic source imaging with FOCUSS: a recursive weighted minimum norm algorithm," *Electroencephalogr. Clin. Neurophysiol.*, vol. 95, no. 4, pp. 231–251, Oct. 1995.

[106] D. Wipf, J. Owen, H. Attias, K. Sekihara, and S. Nagarajan, "Robust Bayesian estimation of the location, orientation, and time course of multiple correlated neural sources using MEG," *NeuroImage*, vol. 49, no. 1, pp. 641–655, Jan. 2010.

[107] U. Gamper, P. Boesiger, and S. Kozerke, "Compressed sensing in dynamic MRI," *Magn. Reson. Med.*, vol. 59, no. 2, pp. 365–373, Feb. 2008.

[108] Z. Zhang, T.-P. Jung, S. Makeig, Z. Pi, and B. D. Rao, "Spatiotemporal sparse

Bayesian learning with applications to compressed sensing of multichannel physiological signals," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 6, pp. 1186–1197, Nov. 2014.

[109] N. Vaswani, "LS-CS-residual (LS-CS): Compressive sensing on least squares residual," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4108–4120, Aug 2010.

[110] X. Zhu, L. Dai, W. Dai, Z. Wang, and M. Moonen, "Tracking a dynamic sparse channel via differential orthogonal matching pursuit," in *Proc. MILCOM*, Oct. 2015.

[111] A. S. Charles, A. Balavoine, and C. J. Rozell, "Dynamic filtering of time-varying sparse signals via l_1 minimization," *IEEE Trans. Signal Process.*, vol. 64, no. 21, pp. 5644–5656, Nov 2016.

[112] N. Vaswani, "Kalman filtered compressed sensing," in *ICIP*, Oct. 2008.

[113] E. Karseras, K. K. Leung, and W. Dai, "Tracking dynamic sparse signals using hierarchical Bayesian Kalman filters," in *Proc. ICASSP*, May 2013.

[114] R. Chalasani and J. C. Principe, "Dynamic sparse coding with smoothing proximal gradient method," in *Proc. ICASSP*, May 2014.

[115] J. Mota, N. Deligiannis, A. C. Sankaranarayanan, V. Cevher, and M. Rodrigues, "Dynamic sparse state estimation using $\ell_1 - \ell_1$ minimization: Adaptive-rate measurement bounds, algorithms and applications," in *Proc. ICASSP*, Apr. 2015.

[116] A. Charles, M. S. Asif, J. Romberg, and C. Rozell, "Sparsity penalties in dynamical system estimation," in *Proc. Conf. on Inform. Sci. and Syst. (CISS)*, Mar. 2011.

[117] D. Sejdinović, C. Andrieu, and R. Piechocki, "Bayesian sequential compressed sensing in sparse dynamical systems," in *Proc. Allerton Conf. on Commun., Control and Comput.*, Nov. 2010.

[118] J. B. Moore, P. K. S. Tam, C. John, and M. Richardson, "Fixed-lag smoothing of nonlinear systems with discrete measurements," *Information Sciences*, vol. 6, pp. 151–160, 1973.

[119] V. Krishnamurthy and J. B. Moore, "On-line estimation of hidden markov model parameters based on the Kullback-Leibler information measure," *IEEE Trans. Signal Process.*, vol. 41, no. 8, pp. 2557–2573, Aug. 1993.

[120] R. Hunger, "Floating point operations in matrix-vector calculus," Munich University of Technology, TUM-LNS-TR-05-05, Tech. Rep. TUM-LNS-TR-05-05, Sep. 2007.

[121] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," *IEEE Trans. Signal Process.*, vol. 5, no. 5, pp. 912–926, Sep. 2011.

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

[123] J. Zyren and W. McCoy, "Overview of the 3GPP long term evolution physical layer," Freescale Semiconductor, Inc., Austin, TX, USA, Tech. Rep. 3GPPEVOLUTIONWP, Jul. 2007.

[124] I.-R. Recommendation, "Guidelines for evaluation of radio transmission technologies (RTTs) for IMT-2000," ITU, Tech. Rep. M.1225, Feb. 1997.

[125] Y. Zheng and C. Xiao, "Simulation models with correct statistical prop- erties for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 6, no. 51, pp. 920–928, Jun. 2003.

[126] E. SMG, "Universal mobile telecommunications system (UMTS), selection procedures for the choice of radio transmission technologies of the UMTS," ETSI, Sophia-Antipolis, France, Tech. Rep. UMTS 21.01 version 3.0.1, Nov. 1997.

[127] C. Studer, C. Benkeser, S. Belfanti, and Q. Huang, "Design and implementation of a parallel turbo-decoder ASIC for 3GPP-LTE," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 8–17, Jan. 2011.

[128] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun. 2010.

[129] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 54, no. 12, pp. 3736–3745, Dec. 2006.

[130] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[131] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Nonlocal sparse models for image restoration," in *ICCV*, Sep. 2009, pp. 2272–2279.

[132] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng, "Shift-invariant sparse coding for audio classification," in *Proc. Conf. Uncertainty in Artificial Intelligence*, Jul. 2007.

[133] M. Zibulevsky and B. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Computation*, vol. 13, no. 4, pp. 863–882, Apr. 2001.

[134] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. ICML*, Jun. 2007, pp. 759–766.

[135] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Proc. CVPR*, Jun. 2008, pp. 1–8.

[136] ——, "Supervised dictionary learning," in *Proc. Adv. in Neural Inform. Process. Syst.*, Dec. 2009, pp. 1033–1040.

[137] D. Bradley and J. Bagnell, "Differentiable sparse coding," in *Proc. Adv. in Neural Inform. Process. Syst.*, Dec. 2009, pp. 113–120.

[138] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun, "Learning invariant features through topographic filter maps," in *Proc. CVPR*, Jun. 2009, pp. 1605–1612.

[139] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. CVPR*, Jun. 2009, pp. 1794 – 1801.

[140] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[141] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *Proc. ICASSP*, Mar. 1999.

[142] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[143] M. Yaghoobi, T. Blumensath, and M. E. Davies, "Dictionary learning for sparse approximations with the majorization method," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2178–2191, Jun. 2009.

[144] W. Dai, T. Xu, and W. Wang, "Simultaneous codeword optimization (SimCO) for dictionary update and learning," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6340–6353, Dec. 2012.

[145] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Learning overcomplete dictionaries based on atom-by-atom updating," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 883–891, Feb. 2014.

[146] S. K. Sahoo and A. Makur, "Dictionary training for sparse representation as generalization of K-Means clustering," *IEEE Signal Process. Lett.*, vol. 20, no. 6, pp. 587–590, Jun. 2013.

[147] K. Schnass, "Convergence radius and sample complexity of ITKM algorithms for dictionary learning," *Appl. Comput. Harmo. A.*, vol. 45, no. 1, pp. 22–58, Jul. 2018.

[148] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin, "Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 130–144, Jan. 2012.

[149] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing - part II: Applications," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5854–5867, Nov. 2014.

[150] L. Yang, J. Fang, H. Cheng, and H. Li, "Sparse Bayesian dictionary learning with a Gaussian hierarchical model," *Signal Process.*, vol. 130, pp. 93–104, Jan. 2017.

[151] P.-A. Absil, M. Robert, and R. Sepulchre, *Optimization algorithms on matrix manifolds.* Princeton University Press, 2008.

[152] P.-A. Absil and J. Malick, "Projection-like retractions on matrix manifolds," *SIAM J. Optim.*, vol. 22, no. 1, pp. 135–158, Oct. 2012.

[153] T. Kaneko, S. Fiori, and T. Tanaka, "Empirical arithmetic averaging over the compact Stiefel manifold," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 883–894, Feb. 2013.

[154] D. Wipf and S. Nagarajan, "Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions," *IEEE J. Sel. Topics Sig. Proc.*, vol. 4, no. 2, pp. 317–329, Apr. 2010.

[155] I. Fedorov and B. D. Rao, "Multimodal sparse bayesian dictionary learning," *ArXiv e-prints*, May 2019. [Online]. Available: https://arxiv.org/abs/1804.03740

[156] J. Cruz Neto, L. De Lima, and P. R. Oliveira, "Geodesic algorithms in Riemannian geometry," *Balkan J. Geom. Appl*, vol. 3, no. 2, pp. 89–100, 1998.

[157] J. Cruz Neto, O. Ferreira, and L. R. Lucambio Perez, "A proximal regularization of the steepest descent method in Riemannian manifold," *Balkan J. Geom. Appl*, vol. 4, no. 2, pp. 1–8, 1999.

[158] Y. Yang, "Globally convergent optimization algorithms on Riemannian manifolds: Uniform framework for unconstrained and constrained optimization," *J. Optim. Theory Appl.*, vol. 132, no. 2, pp. 245–265, 2007.

[159] C. Li and J. Wang, "Newton's method for sections on Riemannian manifolds: Generalized covariant α-theory," *J. Complex.*, vol. 24, no. 3, pp. 423–451, Jun. 2008.

[160] X.-b. Li, N.-j. Huang, Q. H. Ansari, and J.-C. Yao, "Convergence rate of descent method with new inexact line-search on Riemannian manifolds," *J. Optim. Theory Appl.*, pp. 1–25, Sep. 2018.

[161] H. Attouch and J. Bolte, "On the convergence of the proximal algorithm for nonsmooth functions involving analytic features," *Mat. Programming*, vol. 116, no. 1-2, pp. 5–16, Jan. 2009.

[162] H. Liu, W. Wu, and A. Man-Cho So, "Quadratic optimization with orthogonality constraints: Explicit Łojasiewicz exponent and linear convergence of line-search methods," *ArXiv e-prints*, Oct. 2015. [Online]. Available: https://arxiv.org/abs/1510.01025

[163] B. Gao, X. Liu, X. Chen, and Y. xiang Yuan, "On the Łojasiewicz exponent of the quadratic sphere constrained optimization problem," *ArXiv e-prints*, Nov. 2016. [Online]. Available: https://arxiv.org/abs/1611.08781

[164] M. Razaviyayn, H.-W. Tseng, and Z.-Q. Luo, "Dictionary learning for sparse representation: Complexity and algorithms," in *Proc. ICASSP*, May 2014, pp. 5247–5251.

[165] M. E. Muller, "A note on a method for generating points uniformly on N-dimensional spheres," *Commun. ACM*, vol. 2, no. 4, pp. 19–20, Apr. 1959.

[166] I. Fedorov, B. D. Rao, and T. Q. Nguyen, "Multimodal sparse bayesian dictionary

learning applied to multimodal data classification," in *Proc. ICASSP*, Mar. 2017, pp. 2237–2241.

[167] J. G. Serra, M. Testa, R. Molina, and A. K. Katsaggelos, "Bayesian k-svd using fast variational inference," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3344–3359, Mar. 2017.

[168] B. W. Drinkwater and P. D. Wilcox, "Ultrasonic arrays for non-destructive evaluation: A review," *NDT & E International*, vol. 39, no. 7, pp. 525–541, Oct. 2006.

[169] R. Geng, "Modern acoustic emission technique and its application in aviation industry," *Ultrasonics*, vol. 44, pp. e1025–e1029, Dec. 2006.

[170] K. Worden, C. R. Farrar, G. Manson, and G. Park, "The fundamental axioms of structural health monitoring," in *Proc. Math. Phys. Eng. Sci.*, Jun. 2007.

[171] E. V. Malyarenko and M. K. Hinders, "Fan beam and double crosshole Lamb wave tomography for mapping flaws in aging aircraft structures," *J. Acoust. Soc. Am*, vol. 108, no. 4, pp. 1631–1639, Oct. 2000.

[172] K. R. Leonard, E. V. Malyarenko, and M. K. Hinders, "Ultrasonic Lamb wave tomography," *Inverse Probl.*, vol. 18, no. 6, pp. 1795–1808, Nov. 2002.

[173] S. M. Prasad, K. Balasubramaniam, and C. Krishnamurthy, "Structural health monitoring of composite structures using Lamb wave tomography," *Smart Mater. Struct.*, vol. 13, no. 5, pp. N73–N79, Jul. 2004.

[174] H. Gao, Y. Shi, J. Rose, D. O. Thompson, and D. E. Chimenti, "Guided wave

tomography on an aircraft wing with leave in place sensors," in *Proc. AIP*, Apr. 2005.

[175] A. B. Zoubi and V. J. Mathews, "Anomaly imaging using decomposed Lamb wave modes," in *Proc. IWSHM*, Sep. 2017.

[176] A. B. Zoubi, S. Kim, D. O. Adams, and V. J. Mathews, "Lamb wave mode decomposition based on cross-Wigner-Ville distribution and its application to anomaly imaging for structural health monitoring," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, 2018, (Accepted).

[177] D. Wang, W. Zhang, X. Wang, and B. Sun, "Lamb-wave-based tomographic imaging techniques for hole-edge corrosion monitoring in plate structures," *Materials*, vol. 9, no. 11, p. 916, Nov. 2016.

[178] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.

[179] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 5302–5316, Nov. 2009.

[180] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Royal Stat. Soc.*, vol. 68, no. 1, pp. 49–67, Feb. 2006.

[181] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 912–926, Sep. 2011.

[182] J. Fang, Y. Shen, H. Li, and P. Wang, "Pattern-coupled sparse Bayesian learning for recovery of block-sparse signals," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 360–372, Jan. 2015.

[183] J. Fang, L. Zhang, and H. Li, "Two-dimensional pattern-coupled sparse Bayesian learning via generalized approximate message passing," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2920–2930, Jun. 2016.

[184] A. B. Zoubi, V. J. Mathews, J. Harley, and D. Adams, "Lamb waves mode decomposition using the cross-Wigner-Ville distribution," in *Proc. IWSHM*, Sep. 2015.

[185] V. J. Mathews, "Damage mapping in structural health monitoring using a multi-grid architecture," in *Proc. AIP*, vol. 1650, no. 1, Mar. 2015, pp. 1247–1255.

[186] R. M. Dudley, "The sizes of compact subsets of Hilbert space and continuity of Gaussian processes," *Journal of Functional Analysis*, vol. 1, no. 3, pp. 290–330, Oct. 1967.

[187] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1990.

[188] D. Williams, *Probability with martingales*. Cambridge University Press, 1991.

[189] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

[190] B. Delyon, "General results on the convergence of stochastic algorithms," *IEEE Trans. Autom. Control*, vol. 41, no. 9, pp. 1245–1255, Sep. 1996.

[191] M. D. Asic and D. D. Adamovic, "Limit points of sequences in metric spaces," *Am. Math. Mon*, vol. 77, no. 6, pp. 613–616, "Jun.-Jul." 1970.

[192] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imaging Sci.*, vol. 6, no. 3, pp. 1758–1789, Sep. 2013.

[193] C. J. Wu, "On the convergence properties of the EM algorithm," *Ann. Stat.*, vol. 11, no. 1, pp. 95–103, Mar. 1983.