Fundamental Limits and Constructive Methods for Estimation and Sensing of Sparse Signals

A dissertation presented

by

Behtash Babadi

to

The School of Engineering and Applied Sciences

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Engineering Sciences

Harvard University

Cambridge, Massachusetts

May 2011

Thesis advisor

Vahid Tarokh

Author

Behtash Babadi

Fundamental Limits and Constructive Methods for Estimation and Sensing of Sparse Signals

Abstract

The ever-growing size of everyday data in the digital age has urged researchers to devise new techniques for measurement, storage and restoration. A promising mathematical theory for this purpose is Compressed Sensing (CS), which goes beyond the traditional sampling limits by simultaneously measuring and compressing data which is known to have an underlying sparse structure. In this thesis, we analyze fundamental information-theoretic limits of CS, develop algorithms for adaptive estimation and sensing of sparse signals and systems, and study the group randomness properties of certain combinatorial structures which compose the measurement mechanisms of some emerging CS frameworks.

We first study information-theoretic limits on the estimation error of CS reconstruction algorithms, as well as universal sufficiency conditions on the number of measurements required to reliably reconstruct a sparse signal from noisy measurements.

Next, we develop several adaptive algorithms for estimation of sparse signals and systems such as wireless communication channels. We show, both analytically and through simulations, that these algorithms outperform the widely-used traditional

algorithms for adaptive estimation.

Finally, we study the spectral behavior of random matrices from binary combinatorial structures, which have attracted recent attention due to their utility in CS measurement mechanisms. Not only this problem is interesting from the random matrix theory viewpoint, but also sheds light onto the group randomness properties and design of pseudo-random sequences.

Contents

List of Figures

Citations to Previously Published Work

Most of the chapters of this thesis have appeared in print or are currently in press elsewhere. Below is a list, by chapter number, of previously published or in press articles.

- Chapter 2: 'Asymptotic Achievability of the Cramér-Rao Bound for Noisy Compressive Sampling," B. Babadi, N. Kalouptsidis, and V. Tarokh, *IEEE Trans. Sig. Proc.*, Vol. 57, No. 3, pp. 1233 - 1236, March 2009.

- Chapter 4: "SPARLS: The Sparse RLS Algorithm," B. Babadi, N. Kalouptsidis, and V. Tarokh, *IEEE Trans. on Sig. Proc.*, Vol. 58, No. 8, pp. 4013 - 4025, August 2010 *and* "Comparison of SPARLS and RLS Algorithms for Adaptive Filtering," B. Babadi, N. Kalouptsidis, and V. Tarokh, *Proceedings of the 2009 IEEE Sarnoff Symposium*, Princeton, NJ, March 30 - April 1, 2009.

- Chapter 5: "Adaptive Algorithms for Sparse System Identification", N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh, *Signal Processing*, Vol. 91, No. 8, pp. 1910-1919, August 2011 *and* "Adaptive Algorithms for Sparse Nonlinear Channel Estimation", N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh, *Proceedings of the 2009 IEEE Workshop on Statistical Signal Processing (SSP'09)*, Cardiff, Wales, UK, Aug. 31 - Sep. 1, 2009.

- Chapter 6: "An Adaptive Greedy Algorithm with Application to Nonlinear Communications," G. Mileounis, B. Babadi, N. Kalouptsidis, and V. Tarokh, *IEEE Trans. Sig. Proc.*, Vol. 58, No. 6, pp. 2998 - 3007, June 2010 *and* "An Adaptive Greedy Algorithm with Application to Sparse NARMA Identification", G. Mileounis, B. Babadi, N. Kalouptsidis, and V. Tarokh, *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2010)*, March 14 - 19, Dallas, TX.

- Chapter 7: "Spectral Distribution of Random Matrices from Binary Linear Block Codes", B. Babadi and V. Tarokh, *IEEE Trans. Inf. Theor.*, Vol. 57, No. 6, June 2011, "Random Frames from Binary Linear Block Codes", B. Babadi and V. Tarokh, *Proceedings of the 44th Annual Conference on Information Sciences and Systems (CISS 2010)*, Princeton, NJ, March 17 - 19, 2010 (invited), *and* "Group Randomness Properties of Pseudo-Noise and Gold Sequences", B. Babadi, S. Ghassemzadeh, and V. Tarokh, *Proceedings of the 2011 Canadian Workshop on Information Theory*, Kelowna, British Columbia, May 17–20, 2011.

Chapter 1

Introduction and Outline

One of the main challenges of living in the digital age, is the ever-growing need for data storage. Fortunately, a large majority of these data, including images, videos, sounds and texts, are compressible. This allows us to use devices which compress and store the data, using only a fraction of the raw data in certain representational domains. A careful choice of the compression method makes it possible to recover the data accurately from its compressively stored version. This property is referred to as *sparsity* or *sparseness*. In recent years, several researchers have been investigating the idea of employing the underlying sparsity of everyday signals in order to acquire and store them more efficiently. This has led to the emergence of a new branch of applied mathematics with many applications, including medical imaging, computational biology, geophysical data analysis and communication systems.

In this thesis, we employ various techniques from information theory, adaptive filtering and random matrix theory in order to address several fundamental questions in sparsity-aware signal processing. We first study and determine fundamental infor-

mation theoretic limits on the utility of sparsity-aware signal processing. Next, we develop fast, adaptive and online algorithms for estimation and sensing of sparse signals, as well the identification of linear and non-linear systems with underlying sparse structures. Finally, we study the group randomness properties of certain combinatorial structures, which have recently received much attention as efficient building blocks for the sensing of sparse signals.

In the remainder of this chapter, we will first give an overview of compressed sensing, which is the leading mathematical theory of simultaneous data acquisition and compression. We will review some of the main reconstruction techniques in compressed sensing, and highlight some of the fundamental questions therein. Next, we motivate the need to incorporate sparsity-aware mechanisms in developing online algorithms for compressed sensing and estimation of sparse signals and systems. Then, we will pose the problem of group randomness of combinatorial structures in the context of random matrix theory, and discuss its connection to compressed sensing. In each of the above sections, we briefly give an overview of the existing results and techniques and highlight our attempts to improve them. We will conclude this chapter by giving an outline of the remainder of the thesis.

1.1 Compressed Sensing

1.1.1 Introduction

One of the most fundamental paradigms of signal processing is the Nyquist-Shannon Sampling Theorem: it simply states that in order to perfectly reconstruct a

bandlimited signal from its samples, sampling at a rate at least twice the bandwidth of the signal is required.

Several researchers have tried to improve over the Shannon-Nyquist rate, when there is more information about the spectrum of the underlying signal. For instance, when the signal is multi-band, it has been shown that a sampling rate arbitrarily close to twice the ratio of the Lebesgue measure of the spectral support to the total bandwidth is enough to guarantee perfect reconstruction [82]. Several sampling schemes for these signals have been proposed in the literature (See, for example, [37], [75], [101], and [119]). However, Landau [83] proves that stable reconstruction is not possible for any rate below the Shannon-Nyquist rate.

There are many practical signals of interest that are sparse, i.e., the signal has only a few non-zero elements in a certain transform domain (not necessarily the time or frequency domains). For concreteness, consider a class of one-dimensional continuous signals $\mathscr{X} \subseteq \mathcal{L}^2([0,T])$, where $\mathcal{L}^2([0,T])$ denotes the completion of the set of continuous square Lebesgue integrable functions supported on the interval $[0,T]$. Suppose that there exists a set of functions $\{\psi_i(t)\}_{i=1}^M$, such that any $x(t) \in \mathscr{X}$ can be represented as:

$$x(t) = \sum_{i=1}^{M} x_i \psi_i(t), \qquad 0 \leq t \leq T$$

We refer to the set $\mathbf{\Psi} := \{\psi_i(t)\}_{i=1}^M$ and the set of coefficients $\{x_i\}_{i=1}^M$ as *dictionary* and *transform coefficients*, respectively. For a given $x(t) \in \mathscr{X}$, let $L := |\{x_i | x_i \neq 0\}|$. If $L \ll M$, we say that the signal $x(t)$ has a sparse representation in the dictionary $\mathbf{\Psi}$. Also, if the signal $x(t)$ can be well approximated (with a certain fidelity criterion) with only taking L of the transform coefficients $\{x_i\}_{i=1}^M$, we say that $x(t)$ has a sparse

approximation in the dictionary $\mathbf{\Psi}$. For instance, in the context of communication systems, the error sequence occurred in the transmission of a block of data stream is sparse and we are interested in detecting the error sequence [29]. As another example, consider a two-dimensional image. Most images are not sparse in the spatial domain. However, if we transform the image onto the wavelet domain, many of the elements (denoted by the wavelet coefficients) are close to zero and hence the image can be well approximated by a sparse signal in the wavelet domain.

Motivated by the sparseness characteristics of the practical signals, researchers attempted to go beyond the Nyquist-Shannon Sampling Theorem: do we really need to sample at a rate linear in the length of a signal, while we know that there are only a few non-zero elements? Intuitively, the answer is *no*, since the sparseness assumption is *a priori* information which we might be able to exploit to sample the signal economically.

Yet, there are three fundamental questions beyond the Nyquist-Shannon realm:

1) How much can we compress the number of measurements, given the sparseness assumption?

2) How can we sample efficiently, given the sparseness assumption?

3) How can we efficiently reconstruct the original signal given the compressed measurements?

The field of *compressed sensing* aims at simultaneously addressing all the above questions in a unified and tractable framework [27, 29, 46]. That is, if a signal is known to be sparse in a certain representational domain, the premises of compressed sensing are to acquire the signal with very few measurements, and to reconstruct

it faithfully from these measurements, using sophisticated mathematical techniques that aim to overcome under-sampling with the hypothesis of sparsity. Compressed sensing has applications in medical imaging, high-speed digital-to-analog conversion, computational biology, communication systems and statistical signal processing.

There has been a lot of research going on regarding the first question both from the algorithmic and information theoretic points of view. The second question addresses the issue of designing the sampling scheme. In the Nyquist-Shannon sampling regime, the sampling is carried out in both uniform and non-uniform fashions at a rate higher than that specified by the sampling theorem. The sparsity assumption on the other hand, allows one to carefully under-sample the underlying signal, and hence the sampling strategy becomes of utmost importance. For example, if a signal has only one non-zero element, we must be able to reconstruct it by not much more than one measurement, if we design an appropriate sampling scheme to capture the trace of the non-zero element. The third question has also attracted much attention, due to the growing utility of compressed sensing in analyzing very high dimensional data sets. In this thesis, we contribute to better understanding and providing answers to each of these three questions.

1.1.2 A Canonical Model for Compressed Sensing

Suppose that we are interested in measuring, estimating or sensing a signal $\mathbf{x} \in \mathbb{C}^M$. Note that the discrete-time signal $\mathbf{x}$ may represent the transform coefficients of a possibly continuous-time signal which is known to have a sparse representation. In this case, the vector of transform coefficients $\mathbf{x}$ is related to the original signal

by the standard analysis operator defined over the dictionary in which the sparse representation exists. As mentioned earlier, the traditional sensing technique would measure each of the M coordinates of the signal individually. Compressed sensing provides a framework to simultaneously sense and compress the data, whereas data acquisition and compression are typically done in a two-stage fashion in traditional sensing systems.

Suppose that $\mathbf{x} \in \mathbb{C}^M$ is sparse and let the number of non-zero coefficients of $\mathbf{x}$ be L such that $L \ll M$. Similar to several other sensing frameworks, compressed sensing is composed of the the analysis and synthesis stages. In the analysis stage, which is commonly referred to as measurement, a linear combination of the components of the signal are acquired via an $N \times M$ measurement matrix $\mathbf{A}$, such that $L < N \ll M$. Thus the measurement vector, $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$ is a lower dimensional vector than the original signal, where $\mathbf{n}$ stands for the additive noise of the measurement process. The synthesis stage, which is commonly referred to as reconstruction, provides an estimate $\hat{\mathbf{x}}$ of the sparse representation $\mathbf{x}$, satisfying certain error criteria. These two stages compose the canonical framework of compressed sensing.

1.1.3 Noiseless and Noisy Compressed Sensing

In *noiseless compressed sensing*, the vector to be measured $\mathbf{x}$ has exactly $L \ll M$ non-zero coefficients and the measurement process is of the form

$$\mathbf{y} = \mathbf{A}\mathbf{x}.$$

The question of more practical interest is when $\mathbf{x}$ is not necessarily sparse, but can be well-approximated by its largest L components (*i.e.*, it is compressible), and

when the measurement process is noisy,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}.$$

This problem is referred to as *noisy compressed sensing*. In this case, a theoretical question of immense interest in the trade-off between the number of measurements required N, the dimension M, sparsity L and the noise level $\|\mathbf{n}\|_2$.

1.1.4 Reconstruction Methods

The problem of reconstructing a sparse signal from noisy measurements may be posed in the following general form:

$$\arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s. t.} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon, \tag{1.1}$$

where $\mathbf{y}$ is the N dimensional vector of measurements, $\mathbf{A}$ is the $N \times M$ measurement matrix, ϵ controls the reconstruction error, and $\|\cdot\|_0$ denotes the ℓ_0 quasi-norm. For the noiseless problems $\epsilon = 0$, and for the noisy problems, $\epsilon > 0$ and is a function of $\|\mathbf{n}\|_2$. We will refer to the problem in (1.1) as the ℓ_0 minimization problem.

However, solving the ℓ_0 minimization problem for general $\mathbf{A}$ is NP-hard [94]. The difficulty of this problem has caused many researchers to look for alternative reconstruction methods. In what follows, we will review some of the signature methods used for reconstruction of sparse signals from compressed measurement, as well as techniques for obtaining sparse approximations.

ℓ_1 Relaxation

One of the major techniques for sparse reconstruction in the literature of compressed sensing is the ℓ_1 relaxation approach [25, 26, 28–31, 35, 46, 47, 55, 56, 106, 116,

121, 124]. For simplicity, suppose that $\mathbf{x}$ is real. Let $\mathbf{A}_{\mathcal{J}}$ denote the sub-matrix of $\mathbf{A}$ with columns corresponding to the index set $\mathcal{J} \subset \{1, 2, \cdots, M\}$. We denote by $|\mathcal{J}|$ the size of the set $\mathcal{J}$.

We first consider the noiseless case, where the ℓ_0 minimization problem is substituted by the following convex program:

$$\arg \min_{\mathbf{x} \in \mathbb{R}^M} ||\mathbf{x}||_1 \quad \text{s. t.} \quad \mathbf{y} = \mathbf{A}\mathbf{x}. \tag{1.2}$$

It has been shown in the literature that under certain conditions, the solution to (1.2) coincides with that of (1.1) [29,46,49]. In particular, the influential paper of Canés and Tao [29] introduces the notion of *restricted isometry property* (RIP), which identifies certain measurement matrices under which the sparse solution to the ℓ_0 minimization problem can be obtained by the ℓ_1 minimization.

Definition 1.1.1. *(Restricted Isometry Constants [29]) For every integer* $1 \leq S \leq N$*, we define the* S*-restricted isometry constants* δ_S *to be the smallest quantity such that*

$$(1 - \delta_S)\|\mathbf{c}\|_2^2 \leq \|\mathbf{A}_{\mathcal{J}}\mathbf{c}\|_2^2 \leq (1 + \delta_S)\|\mathbf{c}\|_2^2,$$

for all index sets $\mathcal{J}$ *such that* $|\mathcal{J}| \leq S$ *and all* $\mathbf{c} \in \mathbb{R}^{|\mathcal{J}|}$.

Using this property, the following result was proved [29]: If an $N \times M$ matrix $\mathbf{A}$ satisfies RIP with $\delta_S + \delta_{2S} + \delta_{3S} < 1$ and $\|\mathbf{x}\|_0 = L \leq S$, then the solutions to the ℓ_0 and ℓ_1 minimization problems are unique and coinciding.

Whether a measurement matrix $\mathbf{A}$ satisfies the RIP is NP-hard to check [29]. However, with high probability certain $N \times M$ random matrices satisfy RIP where N is a function of L, M and δ_S [29, 95]. For instance, consider $N \times M$ matrices whose

elements are drawn i.i.d. from a Gaussian distribution with mean zero and variance $1/N$. It has been shown in [29] that if $N > CL\log(M/L)$, then $\mathbf{A}$ satisfies RIP with constant δ_S with overwhelming probability, where C is a constant depending on δ_S.

It is also possible to substitute the ℓ_0 minimization problem with the ℓ_1 minimization for the noisy compressed sensing setting, where $\mathbf{n} \neq \mathbf{0}$, and signals which are possibly not sparse but can be well-approximated by a sparse signal:

$$\min \|\mathbf{x}\|_1 \quad \text{s. t.} \quad \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon, \tag{1.3}$$

for a parameter $\epsilon > \|\mathbf{n}\|_2$. In this case, the solution $\hat{\mathbf{x}}$ satisfies

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2 \leq c_1\|\mathbf{n}\|_2 + c_2\frac{\|\mathbf{x}^{(L)} - \mathbf{x}\|_1}{\sqrt{L}},$$

where c_1 and c_2 are constants, and $\mathbf{x}^{(L)}$ is the best L-term approximation to $\mathbf{x}$ [27,28]. For the additive white Gaussian noise model, the Dantzig Selector has been proposed by Candés and Tao [30], with similar performance guarantees.

We finally note that obtaining the exact solutions to (1.2) or (1.3) via standard optimization softwares comes with the price of high computational complexity in high dimensional settings, and hence may not be suitable for online implementations on typical computational platforms [41]. An active area of research aims at constructing low-complexity algorithms to approximate the solution to these problems [48, 55, 56, 121].

Greedy Techniques

Another popular approach to reconstruction of sparse signal from compressed measurements is based on greedy algorithms [117]. A canonical problem of this kind is the

orthogonal matching pursuit (OMP) [88], which generalizes the classical orthonormal basis algorithm. The OMP algorithm greedily chooses an index set $\mathcal{I}$ for the support of the estimate $\hat{\mathbf{x}}$. Let $\mathbf{r}$ be a a residual vector, which is initially equal to $\mathbf{y}$. At each step, a column of the measurement matrix with the highest correlation with the residual is chosen and its index is added to the estimate support. Then, the residual is updated by removing the contribution of the updated support of the estimate, *i.e.*, by projecting the residual onto the orthogonal complement of the subspace spanned by the selected columns of the measurement matrix. This procedure will repeat a total of L times, to output an L-sparse estimate for $\mathbf{x}$ [117].

By using ideas from list decoding of error correcting codes, variants of the OMP algorithm such as the Subspace Pursuit [41] and CoSaMP [95] have been introduced. These algorithms keep a set of columns of $\mathbf{A}$ at each step, which have the highest correlation with the residual, as the candidate set. Then, the support of the estimate is updated by merging the current support with the candidate set and choosing the L largest coordinates. The difference between the two methods is the size of the candidate set, and it was shown empirically in [87] that Subspace Pursuit dominates CoSaMP. Both algorithms work for measurement matrices satisfying certain RIP conditions, guarantee perfect reconstruction in the noiseless setting for $N > \check{C}L\log(M/L)$ for a constant $\check{C}$, and provide performance guarantees similar to ℓ_1 relaxation techniques in the noisy setting.

Coding Theoretic Methods

Relying on the remarkable success of error correcting codes in the past few decades, some researches have tried to cast the problem of compressed sensing in the coding theory framework [3, 24, 69, 72]. In [3], Akcakaya and Tarokh propose the Vandermonde frame as the measurement matrix, and prove that with the help of algebraic decoding algorithms it is possible to uniquely reconstruct a sparse vector with $N = 2L + 1$ measurements in the noiseless setting. This approach has been generalized to other classes of measurement matrices and reconstruction algorithms. In the noiseless case, methods based on encoding and decoding of codes from expander graphs [72], and second order Reed-Muller codes [69] have been proposed. For the noisy compressed sensing setting, methods based on LDPC codes [1, 2, 18] and cosets of second order Reed-Muller codes such as Delsarte-Goethals and Kerdock codes [24] have been investigated.

1.2 Adaptive Filtering and Identification of Sparse Systems

1.2.1 Adaptive Filtering Setting

Often times in practice, direct modeling of the systems under study is either not possible or is of very high cost. Examples are modeling of biological mechanisms or data communication channels. In many of these cases, one can probe the system with an appropriately chosen input and observe the output in order to infer a model

for the system. This procedure is referred to as *system identification*. In particular, by obtaining such a model, one can control or steer the output of the system by pre-processing of the input. This is commonly referred to as *adaptive filtering*. Adaptive filtering is an important part of statistical signal processing, which is highly appealing in estimation problems based on streaming data in environments with unknown statistics [65]. For instance, it is widely used for echo cancellation in speech processing systems and for equalization or channel estimation in wireless systems.

For concreteness, we give an overview of the conventional adaptive filtering setup, which consists of a transversal filter followed by an adaptation block. The input to the transversal filter at time i, denotey by *tap-input vector*, is defined by

$$\mathbf{x}(i) := [x(i), x(i-1), \cdots, x(i-M+1)]^T \tag{1.4}$$

where $x(k)$ is the input at time k, $k = 1, \cdots, n$. The tap-weight vector at time n is defined by

$$\hat{\mathbf{w}}(n) := [\hat{w}_0(n), \hat{w}_1(n), \cdots, \hat{w}_{M-1}(n)]^T. \tag{1.5}$$

The output of the filter at time i is given by

$$y(i) := \hat{\mathbf{w}}^*(n)\mathbf{x}(i). \tag{1.6}$$

where $(\cdot)^*$ denotes the conjugate transpose operator. Let $d(i)$ be the desired output of the filter at time i. We can define the instantaneous error of the filter by

$$e(i) := d(i) - y(i) = d(i) - \hat{\mathbf{w}}^*(n)\mathbf{x}(i). \tag{1.7}$$

The operation of the adaptation block at time n can therefore be stated as the following optimization problem:

$$\min_{\hat{\mathbf{w}}(n)} \quad f\big(e(1), e(2), \cdots, e(n)\big), \tag{1.8}$$

where $f \geq 0$ is a certain cost function. In particular, if $d(i)$ is generated by an unknown tap-weight $\mathbf{w}(n)$, *i.e.*, $d(i) = \mathbf{w}^*(n)\mathbf{x}(i)$, with an appropriate choice of f, one can possibly obtain a good approximation to $\mathbf{w}(n)$ by solving the optimization problem given in (4.5). This is, in general, a system identification problem and is the topic of interest in Chapters 4 - 6.

1.2.2 Sparse Estimation and System Identification

As mentioned earlier, many real-life systems admit sparse representations, that is they are characterized by small number of non-zero coefficients. Sparse systems can also be found in many signal processing [23] and communications applications [16, 79, 109]. For instance, in High Definition Television the significant echoes form a cluster, yet interarrival times between different clusters can be very long [109]. In wireless multipath channels there is a relatively small number of clusters of significant paths [16]. Finally, underwater acoustic channels exhibit long time delays between the multipath terms due to reflections off the sea surface or sea floor [79].

We are interested in obtaining sparse estimates to these signals, or in other words, identifying these sparse systems. Let $d(n)$ be the desired output at time n, contaminated with noise:

$$d(n) = \mathbf{w}^*(n)\mathbf{x}(n) + \eta(n)$$

where $\mathbf{w}(n)$ is the sparse parameter vector, $\mathbf{x}(n)$ is the tap-input vector at time n and $\eta(n)$ is the additive noise. In order to find a sparse estimate to $\mathbf{w}(n)$, one needs to solve the following problem:

$$\min_{\hat{\mathbf{w}}(n)} \|\hat{\mathbf{w}}(n)\|_0 \quad \text{s. t.} \quad f(e(1), e(2), \cdots, e(n)) \leq \epsilon$$

for a carefully chosen cost function f and threshold $\epsilon > 0$. Let $\mathbf{d}(n) := [d(n-M+1), d(n-M+2), \cdots, d(n)]^T$ and $\mathbf{X}(n) := [\mathbf{x}(n-M+1), \mathbf{x}(n-M+2), \cdots, \mathbf{x}(n)]^T$ be the vector of desired outputs and matrix of tap inputs from time 1 to n. Then, by relaxing the above ℓ_0 minimization to ℓ_1 minimization, and choosing the quadratic cost function $\left\|\mathbf{d}(n) - \mathbf{X}(n)\hat{\mathbf{w}}(n)\right\|_2$, one recovers the ℓ_1 regularization technique of compressed sensing. Note that $\mathbf{d}(n)$ and $\mathbf{X}(n)$ play the roles of the observation vector and measurement matrix, respectively.

With this observation, the problem of adaptive sparse system identification or estimation may be posed in various compressed sensing settings (See, for example, [16], [11], [12], [36] and [78]). However, there are several other trade-offs to be taken into account in this case, such as the trackability of the adaptive algorithm, the possibility of recursive implementation and computational complexity per sample. These trade-offs do not exist in the traditional compressed sensing setup, where the underlying sparse vector is assumed to be constant, and the analysis and synthesis stages are performed in a batch mode. These trade-offs are discussed in detail in Chapters 4 - 6, where several viable solutions to the problem of sparse adaptive estimation and system identification are proposed.

1.3 Group Randomness Properties of Combinatorial Structures

1.3.1 Introduction

The need to process very high dimensional data in most of the compressed sensing applications, such as medical imaging, computational biology and geophysical data analysis, together with the relatively high complexity of the successful recovery algorithms, have urged the researchers to resort to more efficient data acquisition and recovery schemes. In particular, some researches have tried to simplify the recovery process by introducing structure into the measurement matrix. Moreover, storing all the elements of the measurement matrix can be burdensome for high dimensional problems. Hence, the structure of the measurement matrix can potentially reduce the computational and storage costs.

Several structured matrices have been recently proposed for compressed sensing. Akçakaya and Tarokh [3] have introduced Vandermonde frames for noiseless sparse recovery, along with a Berlekamp-Massey-type recovery algorithm. In [1,2], Akçakaya et al. construct low density frames inspired by LDPC codes and use belief propagation techniques for recovery. Applebaum et al. [8] have proposed the chirp sensing matrices which allow very fast recovery for sparse Fourier signals. Calderbank et al. [24] introduce a statistical version of the RIP, namely UStRIP, along with several deterministic measurement matrices which act as near isometry on all but a small fraction of the underlying sparse signals. Howard et al. [68] have constructed the Heisenberg frames based on the Heisenberg representation of the finite Heisenberg group. Frames based

on Reed-Muller codes along with a fast recovery algorithm has been introduced in [69] by Howard et al. Recently, Gurevich et al. [62], [61] have introduced the oscillator and extended oscillator frames based on the Weil and Heisenberg-Weil representations of finite symplectic and finite Jacobi groups, respectively. Finally, Indyk [70] has introduced frames based on hash functions constructed from extractor graphs together with an efficient reconstruction algorithm. Hence, studying the spectral behavior of structured measurement matrices, particularly those from error correcting codes, is of interest.

1.3.2 Group Randomness of Combinatorial Structures

Other than its application in compressed sensing, studying the spectral behavior of random matrices from error correcting codes is also of interest in characterizing the group randomness properties of combinatorial structures.

There are two philosophical approaches to studying natural phenomena: one assumes that every phenomenon comprises numerous random ingredients, which can potentially exhibit deterministic collective behaviors. A canonical example of this approach is the celebrated paradigm of central limit theorem in probability theory. The second approach, on the contrary, interprets every phenomenon as intrinsically deterministic, and associates randomness to a class of deterministic phenomena which are computationally hard to perceive or tackle. A classical example is modeling the Brownian motion in statistical physics.

Whether the first approach gives a better description of the reality or the second, is a deep philosophical question, which has challenged philosophers, religious schol-

ars and scientists for centuries. We do not intend to give a cutting answer to this fundamental dilemma. Though, we must not be discouraged from taking a step forward towards the answer to this question. That is, we intend to study objects which are deterministic, and want to measure how much randomness they are capable of manifesting.

As a concrete example, consider elements from a proper k-dimensional subspace V of the n-dimensional binary vector space $\mathrm{GF}(2)^n$, where $\mathrm{GF}(2)$ is the binary Galois field. Clearly, such elements are structured due to their membership in the proper subspace V. Next, consider a normalized $p \times n$ random matrix $\mathbf{\Phi}$ with ± 1 elements, whose rows are formed by uniformly drawing p elements from V under the mapping $0 \mapsto 1$ and $1 \mapsto -1$. In the jargon of coding theory, V is referred to as a k-dimensional binary linear block code of length n. We investigate the asymptotic spectral behavior of random matrices from binary block codes in Chapter 7. Binary block codes are a class of combinatorial structures, which can be highly structures, but as we will see in Chapter 7, they are capable of manifesting group randomness behavior similar to random structures. We will study the group randomness properties of binary block codes through their spectral behavior.

1.3.3 Group Randomness of Sequences

Pseudo-random sequences are deterministic sequences which exhibit certain randomness properties, or more precisely, satisfy certain randomness tests [89]. A class of such sequences denoted by pseudo-noise or PN sequences, are based on the shortened first-order Reed-Muller codes $\mathscr{S}_m$ [86]. These sequences have several applications

in communication systems, such as range-finding, synchronization, modulation and scrambling (See, for example, [60]). The codewords of the shortened first-order Reed-Muller code are known to have several appealing individual randomness properties. Moreover, they are easy to generate using simple linear feedback shift registers [86]. Hence, they are very desirable in application.

Another class of pseudo-random sequences are the Gold sequences which are widely used—among other applications—in Global Positioning System (GPS), due to their desirable cross-correlation properties [99].

Studying the spectral properties of random matrices from such sequences, results in the characterization of their group randomness properties. In particular, the results of Chapter 7 motivate the need to design group randomness tests for pseudo-random sequences.

1.4 Outline of this thesis

Chapter 2 considers the asymptotic achievability of the Cramér-Rao bound for noisy compressive sampling. We consider the Cramér-Rao bound on the mean square error (MSE) of unbiased estimators, where the estimator is provided with the locations of the non-zero elements of the sparse vector. We refer to this class of estimators as Genie-aided estimators. We constructively prove the existence of an asymptotically unbiased estimator, namely the joint typicality decoder, which surprisingly achieves the Cramér-Rao bound of the Genie-aided estimators *without* the knowledge of the locations of the non-zero elements of the sparse vector.

In Chapter 3, we obain universal sufficiency conditions for the problem of asymp-

totically reliable sparse recovery. The existing results obtain these conditions for a *fixed* sparse vector. That is, for any *fixed* sequence of sparse vectors $\mathbf{x} \in \mathbb{C}^M$ such that $\limsup_L \min_i |x_i| \sqrt[4]{\frac{\log L}{L}} = \infty$, asymptotically reliable sparse recovery is possible with any sequence of $N \times M$ Gaussian measurement matrices, if $N > CL$, for some constant C, with overwhelming probability. We extend this result by proving that for *all* sequences of sparse vectors $\mathbf{x} \in \mathbb{C}^M$ such that $\min_i |x_i| \geq \mu_0$ for some constant μ_0, asymptotically reliable sparse recovery is possible for any sequence of $N \times M$ Gaussian measurement matrices, if $N > \tilde{C}L$, for some constant $\tilde{C}$, with overwhelming probability. This result also provides universality sufficient conditions for the achievability of the Cramér-Rao bound presented in Chapter 2.

We introduce a Recursive ℓ_1-Regularized Least Squares (SPARLS) algorithm for the adaptive estimation of sparse signals in Chapter 4. We present analytical and numerical results for the convergence and the steady state mean square error (MSE), which reveal the significant MSE gain of the SPARLS algorithm over the widely-used RLS algorithm. Moreover, the simulation results suggest that the computational complexity of the SPARLS algorithm (with slight modifications) can be less than that of the RLS algorithm, for sparse vectors with fixed support. In particular, for estimating a time-varying Rayleigh fading wireless channel with 5 nonzero coefficients out of 100, the SPARLS algorithm gains about 7dB over the RLS algorithm in MSE and has about 80% less computational complexity.

In Chapter 5, adaptive identification of sparse linear and nonlinear systems described by finite Volterra models is considered. We extend the framework presented in Chapter 4 and present a class of adaptive algorithms based on Expectration-

Maximization and Kalman filtering. This algorithmic generalization and the particular form of the assumed parameter dynamics, enables us to develop several sparse variants including RLS, LMS and fast RLS schemes. The proposed algorithms are applied to two channel settings, namely the Power Amplifier (PA) dynamic nonlinearities in absence and presence of multi-path fading, which reveal significant MSE gains over the traditional algorithms.

In Chapter 6, we establish a conversion procedure that turns greedy algorithms into adaptive schemes for sparse system identification. In particular, a Sparse Adaptive Orthogonal Matching Pursuit (SpAdOMP) algorithm of linear complexity is developed, based on existing greedy algorithms, that provide optimal performance guarantees. Also, the steady-state Mean Square Error (MSE) of the SpAdOMP algorithm is studied analytically. The SpAdOMP algorithm has a considerably low computational complexity, with significant MSE gains. The algorithm is used to estimate Autoregressive Moving Average (ARMA) and Nonlinear ARMA channels. Computer simulations reveal that the proposed algorithm outperforms most existing adaptive algorithms for sparse channel estimation.

We study the empirical spectral distribution of random matrices from binary block codes in Chapter 7. We prove that if the dual distance of the underlying binary block code is sufficiently large, the asymptotic empirical spectral distribution is very similar to that of random i.i.d. Rademacher matrices, which is given by the Marchenko-Pastur distribution. We uniformly bound the absolute distance of the asymptotic empirical spectral distribution to the Marchenko-Pastur distribution. Numerical experiments on low-rate BCH codes confirm the theoretical results. Not only this result

is interesting from the random matrix theory viewpoint, but also it introduces a new criterion for the group randomness of block codes or sequences. In particular, we study the group randomness of shortened first-order Reed-Muller codes and Gold sequences, and show that although both class of pseudo-random sequences have small Kolmogorov complexity (compared to random i.i.d. sequences), they have very different spectral behaviors.

Finally, in Chapter 8, we summarize our contributions and provide possible future research directions.

Chapter 2

Asymptotic Achievability of the Cramér-Rao Bound for Noisy Compressive Sampling

2.1 Introduction

In this chapter, we consider the problem of estimating a sparse vector based on noisy observations. Suppose that we have a compressive sampling (Please see [29] and [46]) model of the form

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \tag{2.1}$$

where $\mathbf{x} \in \mathbb{C}^M$ is the unknown sparse vector to be estimated, $\mathbf{y} \in \mathbb{C}^N$ is the observation vector, $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I}_N) \in \mathbb{C}^N$ is the Gaussian noise and $\mathbf{A} \in \mathbb{C}^{N\times M}$ is the measurement matrix. Suppose that $\mathbf{x}$ is sparse, *i.e.*, $\|\mathbf{x}\|_0 = L < M$. Let $\mathcal{I} := \mathsf{supp}(\mathbf{x})$

and

$$\begin{aligned} \alpha &:= L/N \\ \beta &:= M/L > 2 \end{aligned}$$

be fixed numbers.

The estimator must both estimate the locations and the values of the non-zero elements of $\mathbf{x}$. If a Genie provides us with $\mathcal{I}$, the problem reduces to estimating the values of the non-zero elements of $\mathbf{x}$. We denote the estimator to this reduced problem by Genie-Aided estimator (GAE).

Clearly, the mean squared estimation error (MSE) of any unbiased estimator is no less than that of the GAE (see [30]), since the GAE does not need to estimate the locations of the nonzero elements of $\mathbf{x}$ $\big($ $\log_2 \binom{M}{L} \doteq MH(1/\beta)$ bits, where $H(\cdot)$ is the binary entropy function$\big)$.

Recently, Haupt and Nowak [64] and Candès and Tao [30] have proposed estimators which achieve the estimation error of the GAE up to a factor of $\log M$. In [64], a measurement matrix based on Rademacher projections is constructed and an iterative bound-optimization recovery procedure is proposed. Each step of the procedure requires $O(MN)$ operations and the iterations are repeated until convergence is achieved. It has been shown that the estimator achieves the estimation error of the GAE up to a factor of $\log M$.

Candès and Tao [30] have proposed an estimator based on linear programming, namely the Dantzig Selector, which achieves the estimation error of the GAE up to a factor of $\log M$, for Gaussian measurement matrices. The Dantzig Selector can be recast as a linear program and can be efficiently solved by the well-known primal-

dual interior point methods, as suggested in [30]. Each iteration requires solving an $M \times M$ system of linear equations and the iterations are repeated until convergence is attained.

We construct an estimator based on Shannon theory and the notion of typicality [40] that asymptotically achieves the Cramér-Rao bound on the estimation error of the GAE without the knowledge of the locations of the nonzero elements of $\mathbf{x}$, for Gaussian measurement matrices. Although the estimator presented in this chapter has higher complexity (exponential) compared to the estimators in [30] and [64], to the best of our knowledge it is the first result establishing the achievability of the Cramér-Rao bound for noisy compressive sampling [11]. The problem of finding efficient and low-complexity estimators that achieve the Cramé-Rao bound for noisy compressive sampling still remains open.

The outline of this chapter follows next. In Section 3.2, we state the main result of this chapter and present its proof in Section 7.3.

2.2 Main Result

The main result of this section is the following:

Theorem 2.2.1 (Main Theorem)**.** *In the compressive sampling model of* $\mathbf{y} = \mathbf{A}\mathbf{x}+\mathbf{n}$*, let* $\boldsymbol{A}$ *be a measurement matrix whose elements are i.i.d. Gaussian* $\mathcal{N}(0,1)$*. Let* $e_G^{(M)}$ *be the Cramér-Rao bound on the mean squared error of the GAE,* $\mu(\mathbf{x}) := \min_{i\in\mathcal{I}} |x_i|$ *and* α *and* β *be fixed numbers. If*

- $L\mu^4(\mathbf{x})/\log L \to \infty$ *as* $M \to \infty$*,*

- $\|\mathbf{x}\|^2$ *grows slower than* L^κ*, for some positive constant* κ*,*
- $N > CL$*, where* $C = \max\{C_1, C_2\}$ *with* $C_1 := (18\kappa\sigma^4 + 1)$ *and* $C_2 := (9 + 4\log(\beta - 1))$*,*

assuming that the locations of the nonzero elements of $\mathbf{x}$ *are not known, there exists an estimator (namely, joint typicality decoder) for the nonzero elements of* $\mathbf{x}$ *with mean squared error* $e_\delta^{(M)}$*, such that with high probability*

$$\limsup_M \left| e_\delta^{(M)} - e_G^{(M)} \right| = 0$$

2.3 Proof of the Main Theorem

In order to establish the Main Result, we need to specify the Cramér-Rao bound of the GAE and define the joint typicality decoder. The following lemma gives the Cramér-Rao bound:

Lemma 2.3.1. *For any unbiased estimator* $\hat{\mathbf{x}}$ *of* $\mathbf{x}$*,*

$$\mathbb{E}\{\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2\} \geq e_G^{(M)} := \sigma^2 \operatorname{Tr}\left((\mathbf{A}_\mathcal{I}^* \mathbf{A}_\mathcal{I})^{-1}\right) \tag{2.2}$$

where $\mathbf{A}_\mathcal{I}$ *is the sub-matrix of* $\mathbf{A}$ *with columns corresponding to the index set* $\mathcal{I}$*.*

Proof. Assuming that a Genie provides us with $\mathcal{I}$, we have

$$p_{\mathbf{y}|\mathbf{x}}(\mathbf{y};\mathbf{x}) = \frac{1}{(2\pi)^{N/2}\sigma^N} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{A}_\mathcal{I}\mathbf{x}_\mathcal{I}\|_2^2 \right). \tag{2.3}$$

where $\mathbf{x}_\mathcal{I}$ is the sub-vector of $\mathbf{x}$ with elements corresponding to the index set $\mathcal{I}$. The Fisher information matrix is then given by

$$\mathbf{J}_{ij} := -\mathbb{E}\left\{ \frac{\partial^2 \ln p_{\mathbf{y}|\mathbf{x}}(\mathbf{y};\mathbf{x})}{\partial x_i \partial x_j} \right\} = \frac{1}{\sigma^2}(\mathbf{A}_\mathcal{I}^* \mathbf{A}_\mathcal{I})_{ij} \tag{2.4}$$

Therefore, for any unbiased estimator $\hat{\mathbf{x}}$ by the Cramér-Rao bound [115],

$$\mathbb{E}\big\{\|\hat{\mathbf{x}}-\mathbf{x}\|_2^2\big\} \geq \mathrm{Tr}(\mathbf{J}^{-1}) = \sigma^2\,\mathrm{Tr}\left((\mathbf{A}_{\mathcal{I}}^*\mathbf{A}_{\mathcal{I}})^{-1}\right) \tag{2.5}$$

□

Next, we state a lemma regarding the rank of sub-matrices of random i.i.d. Gaussian matrices:

Lemma 2.3.2. *Let $\boldsymbol{A}$ be a measurement matrix whose elements are i.i.d. Gaussian $\mathcal{N}(0,1)$, $\mathcal{J} \subset \{1,2,\cdots,M\}$ such that $|\mathcal{J}| = L$ and $\mathbf{A}_{\mathcal{J}}$ be the sub-matrix of $\mathbf{A}$ with columns corresponding to the index set $\mathcal{J}$. Then,* $\mathrm{rank}(\mathbf{A}_{\mathcal{J}}) = L$ *with probability 1.*

We can now define the notion of joint typicality. We adopt the definition from [4]:

Definition 2.3.3. *We say an $N \times 1$ noisy observation vector, $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$ and a set of indices $\mathcal{J} \subset \{1,2,\cdots,M\}$, with $|\mathcal{J}| = L$, are δ-jointly typical if* $\mathrm{rank}(\mathbf{A}_{\mathcal{J}}) = L$ *and*

$$\left|\frac{1}{N}\|\Pi_{\mathbf{A}_{\mathcal{J}}}^{\perp}\mathbf{y}\|^2 - \frac{N-L}{N}\sigma^2\right| < \delta \tag{2.6}$$

where $\mathbf{A}_{\mathcal{J}}$ is the sub-matrix of $\mathbf{A}$ with columns corresponding to the index set $\mathcal{J}$ and $\Pi_{\mathbf{A}_{\mathcal{J}}}^{\perp} := \mathbf{I} - \mathbf{A}_{\mathcal{J}}(\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{J}})^{-1}\mathbf{A}_{\mathcal{J}}^*$. We denote the event of δ-jointly typicality of $\mathcal{J}$ with $\mathbf{y}$ by $E_{\mathcal{J}}$.

Note that we can make the assumption of $\mathrm{rank}(\mathbf{A}_{\mathcal{J}}) = L$ without loss of generality, based on Lemma 2.3.2.

Definition 2.3.4 (Joint Typicality Decoder)**.** *The joint typicality decoder finds a set of indices $\hat{\mathcal{I}}$ which is δ-jointly typical with $\mathbf{y}$, by projecting $\mathbf{y}$ onto all the possible $\binom{M}{L}$*

L-dimensional subspaces spanned by the columns of $\mathbf{A}$ *and choosing the one satisfying Eq. (3.1). It then produces the estimate* $\hat{\mathbf{x}}^{(\hat{\mathcal{I}})}$ *by projecting* $\mathbf{y}$ *onto the subspace spanned by* $\mathbf{A}_{\hat{\mathcal{I}}}$*:*

$$\hat{\mathbf{x}}^{(\hat{\mathcal{I}})} = \begin{cases} \left(\mathbf{A}_{\hat{\mathcal{I}}}^* \mathbf{A}_{\hat{\mathcal{I}}}\right)^{-1} \mathbf{A}_{\hat{\mathcal{I}}}^* \mathbf{y}\Big|_{\hat{\mathcal{I}}} & \text{on } \hat{\mathcal{I}}, \\ \mathbf{0} & \text{elsewhere.} \end{cases} \tag{2.7}$$

If the estimator does not find any set δ*-typical to* $\mathbf{y}$*, it will output the zero vector as the estimate. We denote this event by* E_0*.*

In what follows, we show that the joint typicality decoder has the property stated in the Main Theorem. The next lemma from [4] gives non-asymptotic bounds on the probability of the error events $E_{\mathcal{I}}^c$ and $E_{\mathcal{J}}$, averaged over the ensemble of random i.i.d. Gaussian measurement matrices:

Lemma 2.3.5 (Lemma 3.3 of [4])**.** *For any* $\delta > 0$*,*

$$\mathbb{P}\left(\left|\frac{1}{N}\|\Pi_{\mathbf{A}_{\mathcal{I}}}^{\perp}\mathbf{y}\|^2 - \frac{N-L}{N}\sigma^2\right| > \delta\right) \leq 2\exp\left(-\frac{\delta^2}{4\sigma^4}\frac{N^2}{N-L+\frac{2\delta}{\sigma^2}N}\right) \tag{2.8}$$

and

$$\mathbb{P}\left(\left|\frac{1}{N}\|\Pi_{\mathbf{A}_{\mathcal{J}}}^{\perp}\mathbf{y}\|^2 - \frac{N-L}{N}\sigma^2\right| < \delta\right) \leq \exp\left(-\frac{N-L}{4}\left(\frac{\sum_{k\in\mathcal{I}\setminus\mathcal{J}}|x_k|^2 - \delta'}{\sum_{k\in\mathcal{I}\setminus\mathcal{J}}|x_k|^2 + \sigma^2}\right)^2\right) \tag{2.9}$$

where $\mathcal{J}$ *is an index set such that* $|\mathcal{J}| = L$*,* $|\mathcal{I}\cap\mathcal{J}| = K < L$*,* $rank(\mathbf{A}_{\mathcal{J}}) = L$ *and*

$$\delta' := \delta\frac{N}{N-L} < \min_k |x_k|^2.$$

Proof. The proof is given in [4].

□

Proof of the Main Theorem. Let $\mathcal{A}$ be the set of all $N \times M$ matrices with i.i.d. Gaussian $\mathcal{N}(0,1)$ entries. Consider $\mathbf{A} \in \mathcal{A}$. Then, it is known that [29] for all $\mathcal{K} \in \{1, 2, \cdots, M\}$ such that $|\mathcal{K}| = L$,

$$\mathbb{P}\bigg(\lambda_{\max}\Big(\frac{1}{N}\mathbf{A}_{\mathcal{K}}^{*}\mathbf{A}_{\mathcal{K}}\Big) > (1+\sqrt{\alpha}+\epsilon)^2\bigg) \leq \exp\bigg(-\frac{1}{2}M\frac{\sqrt{H(1/\beta)}}{\sqrt{\alpha\beta}}\epsilon\bigg), \tag{2.10}$$

and

$$\mathbb{P}\bigg(\lambda_{\min}\Big(\frac{1}{N}\mathbf{A}_{\mathcal{K}}^{*}\mathbf{A}_{\mathcal{K}}\Big) < (1-\sqrt{\alpha}-\epsilon)^2\bigg) \leq \exp\bigg(-\frac{1}{2}M\frac{\sqrt{H(1/\beta)}}{\sqrt{\alpha\beta}}\epsilon\bigg). \tag{2.11}$$

Let $\mathcal{A}_{\epsilon}^{(\alpha)} \subset \mathcal{A}$, such that for all $\mathbf{A} \in \mathcal{A}_{\epsilon}^{(\alpha)}$, the eigenvalues of $\frac{1}{N}\mathbf{A}_{\mathcal{K}}^{*}\mathbf{A}_{\mathcal{K}}$ lie in the interval $[(1-\sqrt{\alpha}-\epsilon)^2, (1+\sqrt{\alpha}+\epsilon)^2]$. Clearly, we have

$$P_{\mathcal{A}}\Big(\mathcal{A}_{\epsilon}^{(\alpha)}\Big) \geq 1 - 2\exp\bigg(-\frac{1}{2}M\frac{\sqrt{H(1/\beta)}}{\sqrt{\alpha\beta}}\epsilon\bigg).$$

where $P_{\mathcal{A}}(\cdot)$ denotes the Gaussian measure defined over $\mathcal{A}$. It is easy to show that

$$\mathbb{E}_{\mathbf{A}\in\mathcal{A}_{\epsilon}^{(\alpha)}}\big\{f(\mathbf{A})\big\} \leq \frac{\mathbb{E}_{\mathbf{A}\in\mathcal{A}}\big\{f(\mathbf{A})\big\}}{P_{\mathcal{A}}\Big(\mathcal{A}_{\epsilon}^{(\alpha)}\Big)}, \tag{2.12}$$

where $f(\mathbf{A})$ is any function of $\mathbf{A}$ for which $\mathbb{E}_{\mathbf{A}\in\mathcal{A}}\big\{f(\mathbf{A})\big\}$ exists. In what follows, we will upper-bound the MSE of the joint typicality decoder, averaged over all Gaussian measurement matrices in $\mathcal{A}_{\epsilon}^{(2\alpha)}$. Let $e_{\delta}^{(M)}$ denote the MSE of the joint typicality decoder. We have:

$$\begin{aligned} e_{\delta}^{(M)} &= \mathbb{E}_{\mathbf{n}}\Big\{\|\hat{\mathbf{x}}-\mathbf{x}\|_2^2\Big\} \\ &\leq \mathbb{E}_{\mathbf{n}}\Big\{\big\|\hat{\mathbf{x}}^{(\mathcal{I})}-\mathbf{x}\big\|_2^2\Big\} + \|\mathbf{x}\|_2^2\mathbb{E}_{\mathbf{n}}\big\{\mathbb{I}(E_{\mathcal{I}}^{c})\big\} \\ &\quad + \sum_{\mathcal{J}\neq\mathcal{I}}\mathbb{E}_{\mathbf{n}}\Big\{\big\|\hat{\mathbf{x}}^{(\mathcal{J})}-\mathbf{x}\big\|_2^2\mathbb{I}(E_{\mathcal{J}})\Big\} := \check{e}_{\delta}^{(M)} \end{aligned} \tag{2.13}$$

where $\mathbb{I}(\cdot)$ is the indicator function, $\mathbb{E}_{\mathbf{n}}\{\cdot\}$ denotes the expectation operator defined over the noise density, and the inequality follows from the union bound and the facts

that $\mathbb{E}_{\mathbf{n}}\{\mathbb{I}(E_0)\} =: \mathbb{P}(E_0) \leq \mathbb{P}(E_{\mathcal{I}}^c) := \mathbb{E}_{\mathbf{n}}\{\mathbb{I}(E_{\mathcal{I}}^c)\}$ and $\mathbb{I}(E_{\mathcal{I}}) \leq 1$. Now, averaging the upper bound $\check{e}_{\delta}^{(M)}$ (defined in (2.13)) over $\mathcal{A}_{\epsilon}^{(2\alpha)}$ yields:

$$\begin{aligned}\mathbb{E}_{\mathbf{A}}\Big\{\check{e}_{\delta}^{(M)}\Big\} &= \int_{\mathcal{A}_{\epsilon}^{(2\alpha)}} \mathbb{E}_{\mathbf{n}|\mathbf{A}}\Big\{\big\|\hat{\mathbf{x}}^{(\mathcal{I})} - \mathbf{x}\big\|_2^2\Big\} dP(\mathbf{A}) \\ &+ \int_{\mathcal{A}_{\epsilon}^{(2\alpha)}} \|\mathbf{x}\|_2^2 \mathbb{E}_{\mathbf{n}}\big\{\mathbb{I}(E_{\mathcal{I}}^c)\big\} dP(\mathbf{A}) \\ &+ \int_{\mathcal{A}_{\epsilon}^{(2\alpha)}} \sum_{\mathcal{J}\neq\mathcal{I}} \mathbb{E}_{\mathbf{n}|\mathbf{A}}\Big\{\big\|\hat{\mathbf{x}}^{(\mathcal{J})} - \mathbf{x}\big\|_2^2 \mathbb{I}(E_{\mathcal{J}})\Big\} dP(\mathbf{A}) \qquad (2.14)\end{aligned}$$

where $dP(\mathbf{A})$ and $\mathbb{E}_{\mathbf{A}}$ denote the conditional Gaussian probability measure and the expectation operator defined on the set $\mathcal{A}_{\epsilon}^{(2\alpha)}$, respectively.

The first term on the right hand side of Eq. (7.22) can be written as:

$$\begin{aligned}\int_{\mathcal{A}_{\epsilon}^{(2\alpha)}} \mathbb{E}_{\mathbf{n}|\mathbf{A}}\Big\{\Big\|(\mathbf{A}_{\mathcal{I}}^*\mathbf{A}_{\mathcal{I}})^{-1}\mathbf{A}_{\mathcal{I}}^*\mathbf{y}\big|_{\mathcal{I}} - \mathbf{x}\Big\|_2^2\Big\} dP(\mathbf{A}) &= \mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{\big\|(\mathbf{A}_{\mathcal{I}}^*\mathbf{A}_{\mathcal{I}})^{-1}\mathbf{A}_{\mathcal{I}}^*\mathbf{n}\big\|_2^2\Big\} \\ &= \mathbb{E}_{\mathbf{A}}\Big\{\sigma^2 \operatorname{Tr}(\mathbf{A}_{\mathcal{I}}^*\mathbf{A}_{\mathcal{I}})^{-1}\Big\}. \qquad (2.15)\end{aligned}$$

The second term on the right hand side of Eq. (7.22) can be upper-bounded as

$$\int_{\mathcal{A}_{\epsilon}^{(2\alpha)}} \|\mathbf{x}\|_2^2 \mathbb{E}_{\mathbf{n}}\big\{\mathbb{I}(E_{\mathcal{I}}^c)\big\} dP(\mathbf{A}) \leq \frac{2\|\mathbf{x}\|_2^2}{P_{\mathcal{A}}\Big(\mathcal{A}_{\epsilon}^{(2\alpha)}\Big)} \exp\left(-\frac{\delta^2}{4\sigma^4}\frac{N^2}{N-L+\frac{2\delta}{\sigma^2}N}\right) \qquad (2.16)$$

where the inequality follows from Lemma 2.3.5 and Eq. (2.12).

Now, let $\delta' = \delta N/(N-L) = \zeta\mu^2(\mathbf{x})$ for some $2/3 < \zeta < 1$. Recall that by hypothesis, $\mu^2(\mathbf{x})$ goes to zero slower than $\sqrt{\frac{\log L}{L}}$. Suppose that L is large enough, so that $\sigma^2 \geq 2\zeta\mu^2(\mathbf{x})$. Then, it can be shown that the right hand side of Eq. (2.16) decays faster than $1/L^c$, where $c := \zeta^2 C_0/2\sigma^4$. Now, by the hypothesis of $N > (18\kappa\sigma^4+1)L$ we have $C_0 > 9\kappa\sigma^4/2$, therefore $c > \kappa$, and hence the right hand side of Eq. (2.16) goes to zero faster than $1/L^{(c-\kappa)}$, as $M, N \to \infty$.

Remark. If we assume the weaker condition that $\mu(\mathbf{x}) = \mu_0$, constant independent of M, N and L, then δ will be constant and the error exponent in Eq. (2.16)

decays to zero exponentially fast. Hence, as long as $\|\mathbf{x}\|_2^2$ grows polynomially, the whole term on the right hand side of Eq. (2.16) decays to zero exponentially fast. Therefore, the claim of the theorem holds with overwhelming probability, *i.e.*, probability of failure exponentially small in L (rather than with high probability, which refers to the case of failure probability polynomially small in L).

Finally, consider the term corresponding to $\mathcal{J}$ in the third expression on the right hand side of Eq. (7.22). This term can be simplified as

$$\begin{aligned}\mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{\big\|\hat{\mathbf{x}}^{(\mathcal{J})}-\mathbf{x}\big\|_2^2\mathbb{I}(E_{\mathcal{J}})\Big\} &= \mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{\big\|\hat{\mathbf{x}}_{\mathcal{J}}^{(\mathcal{J})}-\mathbf{x}_{\mathcal{J}}\big\|_2^2\mathbb{I}(E_{\mathcal{J}})\Big\}+\mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{\big\|\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2\mathbb{I}(E_{\mathcal{J}})\Big\}\\ &= \underbrace{\mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{\big\|(\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{J}})^{-1}\mathbf{A}_{\mathcal{J}}^*\mathbf{A}\mathbf{x}-\mathbf{x}_{\mathcal{J}}\big\|_2^2\mathbb{I}(E_{\mathcal{J}})\Big\}}_{T_1}\\ &\quad+\underbrace{\mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{\big\|(\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{J}})^{-1}\mathbf{A}_{\mathcal{J}}^*\mathbf{n}\big\|_2^2\mathbb{I}(E_{\mathcal{J}})\Big\}}_{T_2}\\ &\quad+\mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{\big\|\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2\mathbb{I}(E_{\mathcal{J}})\Big\}. \qquad (2.17)\end{aligned}$$

where the first equality follows from the fact that $\hat{\mathbf{x}}^{(\mathcal{J})}$ is zero outside of the index set $\mathcal{J}$, and the second equality follows from the assumption that $\mathbf{n}$ is zero-mean and independent of $\mathbf{A}$ and $\mathbf{x}$. Now, the term T_1 can be further simplified as

$$\begin{aligned}T_1 &= \mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{\big\|(\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{J}})^{-1}\mathbf{A}_{\mathcal{J}}^*(\mathbf{A}_{\mathcal{J}}\mathbf{x}_{\mathcal{J}}+\mathbf{A}_{\mathcal{I}\setminus\mathcal{J}}\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}})-\mathbf{x}_{\mathcal{J}}\big\|_2^2\mathbb{I}(E_{\mathcal{J}})\Big\}\\ &= \mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{\big\|(\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{J}})^{-1}\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{I}\setminus\mathcal{J}}\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2\mathbb{I}(E_{\mathcal{J}})\Big\}\end{aligned}$$

Invoking the sub-multiplicative property of matrix norms and applying the Cauchy-Shwarz inequality to the mean of the product of non-negative random variables, we

can further upper bound T_1 as:

$$\begin{aligned} T_1 &\leq \mathbb{E}_{\mathbf{A}}\Big\{ \big\|(\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{J}})^{-1}\big\|_2^2 \big\|\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2 \big\|\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2 \mathbb{I}(E_{\mathcal{J}}) \Big\} \\ &\leq \mathbb{E}_{\mathbf{A}}\Big\{ \big\|(\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{J}})^{-1}\big\|_2^2 \Big\} \mathbb{E}_{\mathbf{A}}\Big\{ \big\|\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2 \Big\} \big\|\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2 \mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{ \mathbb{I}(E_{\mathcal{J}}) \Big\} \end{aligned} \tag{2.18}$$

For simplicity, let $\epsilon = \sqrt{2\alpha}$. Now, consider the term $\mathbb{E}_{\mathbf{A}}\Big\{ \big\|(\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{J}})^{-1}\big\|_2^2 \Big\}$, the average of the squared spectral norm of the matrix $(\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{J}})^{-1}$. Since $\mathbf{A} \in \mathcal{A}_\epsilon$, the minimum eigenvalue of $\frac{1}{N}\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{J}}$ is greater than or equal to $(1-2\sqrt{2\alpha})^2$, since $|\mathcal{J}| = L < 2L$, and $\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{J}}$ is a sub-matrix of $\mathbf{A}_{\mathcal{K}}^*\mathbf{A}_{\mathcal{K}}$ for some $\mathcal{K}$ such that $\mathcal{J} \subset \mathcal{K}$ and $|\mathcal{K}| = 2L$ [67, Theorem 4.3.15]. Similarly, for the second term, $\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{I}\setminus\mathcal{J}}$ is a sub-matrix of $\mathbf{A}_{\mathcal{J}\cup\mathcal{I}}^*\mathbf{A}_{\mathcal{J}\cup\mathcal{I}} - \mathbf{I}$. Since $|\mathcal{J}\cup\mathcal{I}| \leq 2L$, the spectral norm of $\frac{1}{N}\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{I}\setminus\mathcal{J}}$ is upper bounded by $(1+2\sqrt{2\alpha})^2 - 1 = 8\alpha + 4\sqrt{2\alpha}$ [67, Theorem 4.3.15], [95]. Finally, since the averaging is over $\mathcal{A}_\epsilon^{(2\alpha)}$, the same bounds hold for the average of the matrix norms. Hence, T_1 can be upper bounded by

$$T_1 \leq \frac{(8\alpha + 4\sqrt{2\alpha})^2}{(1-2\sqrt{2\alpha})^4} \mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{ \mathbb{I}(E_{\mathcal{J}}) \Big\} \|\mathbf{x}\|_2^2.$$

Note that by the hypothesis of $N > C_2 L$, we have $\alpha < 1/9$. Hence, $2\sqrt{2\alpha} = \frac{2\sqrt{2}}{3} < 1$. Also, T_2 is similarly upper bounded by $\sigma^2 \mathbb{E}_{\mathbf{A}}\big\{ \mathrm{Tr}(\mathbf{A}_{\mathcal{J}}^*\mathbf{A}_{\mathcal{J}})^{-1} \big\} \mathbb{E}_{\mathbf{n},\mathbf{A}}\big\{ \mathbb{I}(E_{\mathcal{J}}) \big\}$, which can be further bounded by

$$T_2 \leq \frac{\alpha\sigma^2}{(1-2\sqrt{2\alpha})^2} \mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{ \mathbb{I}(E_{\mathcal{J}}) \Big\}.$$

Therefore, Eq. (2.17) can be further bounded as

$$\mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{ \big\|\hat{\mathbf{x}}^{(\mathcal{J})} - \mathbf{x}\big\|_2^2 \mathbb{I}(E_{\mathcal{J}}) \Big\} \leq \underbrace{\left[\left(1 + \tfrac{(8\alpha+4\sqrt{2\alpha})^2}{(1-2\sqrt{2\alpha})^4} \right) \|\mathbf{x}\|_2^2 + \tfrac{\alpha\sigma^2}{(1-2\sqrt{2\alpha})^2} \right]}_{\mathscr{C}(\mathbf{x})} \mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{ \mathbb{I}(E_{\mathcal{J}}) \Big\}$$

Now, from Lemma 2.3.5 and Eq. (2.12), we have

$$\begin{aligned}\sum_{\mathcal{J}\neq\mathcal{I}} \mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{ \big\|\hat{\mathbf{x}}^{(\mathcal{J})} - \mathbf{x}\big\|_2^2 \mathbb{I}(E_{\mathcal{J}})\Big\} \\ &\leq \mathscr{C}(\mathbf{x}) \sum_{\mathcal{J}\neq\mathcal{I}} \mathbb{E}_{\mathbf{n},\mathbf{A}}\Big\{\mathbb{I}(E_{\mathcal{J}})\Big\} \\ &\leq \frac{\mathscr{C}(\mathbf{x})}{P_{\mathcal{A}}\big(\mathcal{A}_\epsilon^{(2\alpha)}\big)} \sum_{\mathcal{J}\neq\mathcal{I}} \exp\left(-\frac{N-L}{4}\left(\frac{\sum_{k\in\mathcal{I}\setminus\mathcal{J}}|x_k|^2 - \delta'}{\sum_{k\in\mathcal{I}\setminus\mathcal{J}}|x_k|^2 + \sigma^2}\right)^2\right),\end{aligned} \tag{2.19}$$

where x_k denotes the kth component of $\mathbf{x}$. The number of index sets $\mathcal{J}$ such that $|\mathcal{J}\cap\mathcal{I}| = K < L$ is upper-bounded by $\binom{L}{L-K}\binom{M-L}{L-K}$. Also, $\sum_{k\in\mathcal{I}\setminus\mathcal{J}}|x_k|^2 \geq (L-K)\mu^2(\mathbf{x})$. Therefore, we can rewrite the right hand side of Eq. (2.19) as

$$\begin{aligned}&\frac{\mathscr{C}(\mathbf{x})}{P_{\mathcal{A}}\big(\mathcal{A}_\epsilon^{(2\alpha)}\big)} \sum_{K=0}^{L-1} \binom{L}{L-K}\binom{M-L}{L-K} \exp\left(-\frac{N-L}{4}\left(\frac{(L-K)\mu^2(\mathbf{x}) - \delta'}{(L-K)\mu^2(\mathbf{x}) + \sigma^2}\right)^2\right) \\ &= \frac{\mathscr{C}(\mathbf{x})}{P_{\mathcal{A}}\big(\mathcal{A}_\epsilon^{(2\alpha)}\big)} \sum_{K'=1}^{L} \binom{L}{K'}\binom{M-L}{K'} \exp\left(-\frac{N-L}{4}\left(\frac{K'\mu^2(\mathbf{x}) - \delta'}{K'\mu^2(\mathbf{x}) + \sigma^2}\right)^2\right)\end{aligned} \tag{2.20}$$

We use the inequality

$$\binom{L}{K'} \leq \exp\left(K' \log\left(\frac{Le}{K'}\right)\right) \tag{2.21}$$

in order to upper-bound the K'th term of the summation in Eq. (2.20) by

$$\exp\left(L\frac{K'}{L}\log\left(\frac{e}{\frac{K'}{L}}\right) + L\frac{K'}{L}\log\frac{(\beta-1)e}{\frac{K'}{L}} - C_0 L\Big(\frac{L\frac{K'}{L}\mu^2(\mathbf{x}) - \delta'}{L\frac{K'}{L}\mu^2(\mathbf{x}) + \sigma^2}\Big)^2\right) \tag{2.22}$$

where $C_0 := (N-L)/4L$. We define

$$f(z) := Lz\log\frac{e}{z} + Lz\log\frac{(\beta-1)e}{z} - C_0 L\Big(\frac{Lz\mu^2(\mathbf{x}) - \delta'}{Lz\mu^2(\mathbf{x}) + \sigma^2}\Big)^2 \tag{2.23}$$

By Lemmas 3.4, 3.5 and 3.6 of [4], it can be shown that $f(z)$ asymptotically attains its maximum at either $z = \frac{1}{L}$ or $z = 1$, if $L\mu^4(\mathbf{x})/\log L \to \infty$ as $N \to \infty$. Thus, we

can upper-bound the right hand side of Eq. (2.19) by

$$\frac{\mathscr{C}(\mathbf{x})}{P_{\mathcal{A}}\big(\mathcal{A}_\epsilon^{(2\alpha)}\big)} \sum_{K'=1}^{L} \exp\Big(\max\{f(1/L), f(1)\}\Big). \tag{2.24}$$

The values of $f(1/L)$ and $f(1)$ can be written as:

$$f(1/L) = 2\log L + 2 + \log(\beta-1) - C_0 L\Big(\frac{\mu^2(\mathbf{x}) - \delta'}{\mu^2(\mathbf{x}) + \sigma^2}\Big)^2 \tag{2.25}$$

and

$$f(1) = L(2 + \log(\beta-1)) - C_0 L\Big(\frac{L\mu^2(\mathbf{x}) - \delta'}{L\mu^2(\mathbf{x}) + \sigma^2}\Big)^2. \tag{2.26}$$

Since $C_0 > 2 + \log(\beta-1)$ due to the assumption of $\alpha < 1/(9 + 4\log(\beta-1))$, both $f(1)$ and $f(1/L)$ will grow to $-\infty$ linearly as $N \to \infty$. Hence, the exponent in Eq. (2.24) will grow to $-\infty$ as $N \to \infty$, since $\|\mathbf{x}\|_2^2$ in $\mathscr{C}(\mathbf{x})$ grows polynomially in L and $P_{\mathcal{A}}\big(\mathcal{A}_\epsilon^{(2\alpha)}\big)$ tends to 1 exponentially fast. Let L be large enough such that

$$\frac{\mathscr{C}(\mathbf{x})}{P_{\mathcal{A}}\big(\mathcal{A}_\epsilon^{(2\alpha)}\big)} \sum_{K'=1}^{L} \exp\Big(\max\{f(1/L), f(1)\}\Big) \le \frac{1}{L^{c-\kappa}},$$

where $c = \zeta^2 C_0 / 2\sigma^4$. Now, by Markov's inequality we have:

$$P\left(\left|\check{e}_\delta^{(M)} - \sigma^2 \operatorname{Tr}(\mathbf{A}_{\mathcal{I}}^* \mathbf{A}_{\mathcal{I}})^{-1}\right| > \frac{1}{L^{\frac{c-\kappa}{2}}}\right) \le \frac{2}{L^{\frac{c-\kappa}{2}}}. \tag{2.27}$$

Hence, for any measurement matrix $\mathbf{A}$ from the restricted ensemble of i.i.d. Gaussian matrices $\mathcal{A}_\epsilon^{(2\alpha)}$, we have

$$\left|\check{e}_\delta^{(M)} - \sigma^2 \operatorname{Tr}(\mathbf{A}_{\mathcal{I}}^* \mathbf{A}_{\mathcal{I}})^{-1}\right| \le \frac{1}{L^{\frac{c-\kappa}{2}}} \tag{2.28}$$

with probability exceeding

$$1 - \frac{2}{L^{\frac{c-\kappa}{2}}}. \tag{2.29}$$

Now, since the probability measure is defined over $\mathcal{A}_{\epsilon}^{(2\alpha)}$, the statement of Eq. (2.28) holds with probability exceeding

$$1 - 2\exp\left(-\frac{1}{2} M \frac{\sqrt{2H(2/\beta)}}{\sqrt{\beta}} \right) - \frac{2}{L^{\frac{c-\kappa}{2}}} \tag{2.30}$$

over all the measurement matrices in the Gaussian ensemble $\mathcal{A}$.

Recall that the expression $\sigma^2 \operatorname{Tr}(\mathbf{A}_{\mathcal{I}}^{*}\mathbf{A}_{\mathcal{I}})^{-1}$ is indeed the Cramér-Rao bound of the GAE. Noting that $\check{e}_{\delta}^{(M)}$ is an upper bound on $e_{\delta}^{(M)}$ concludes the proof the the main theorem. □

Chapter 3

An Information-Theoretic Universal Sufficiency Condition for Noisy Sparse Recovery

3.1 Introduction

In this chapter, we develop universal sufficient conditions for asymptotically reliable recovery of sparse signals with respect to the $0-1$ loss metric. This metric is defined as

$$p(\mathbf{x}, \hat{\mathbf{x}}) = 1 - \mathbb{I}(\{\hat{x}_i \neq 0, \forall i \in \mathcal{I})\} \times \mathbb{I}(\{\hat{x}_j = 0, \forall j \notin \mathcal{I})\}$$

where $\mathbb{I}(\cdot)$ is the indicator function. This metric is referred to as Error Metric 1 in [4].

Asymptotically reliable sparse recovery in the information theoretic setting under the $0-1$ loss metric was first studied by Wainwright in an excellent paper [123]. Using an optimal decoder that decodes to the closest subspace, it was shown that in the pres-

ence of additive Gaussian noise, $N = O(L)$ and $N = O(L\log(M-L))$ measurements were necessary and sufficient for the linear and sublinear sparsity regimes respectively. For the linear regime, it was also required that $\sqrt{L}\big(\min_i |x_i^{(M)}|\big)/\sqrt{\log L} \to \infty$ as $M \to \infty$, leading to $P := \|\mathbf{x}\|_2^2 \to \infty$. The reason for this requirement is that at high dimensions, the requirement of exact support recovery becomes too stringent for an average case analysis. In [4], Akcakaya and Tarokh proved that for a given sequence of $\mathbf{x}^{(M)} \in \mathbb{C}^M$, asymptotically reliable sparse recovery is possible given $N = \mathcal{O}(L)$ and $N = \mathcal{O}(L\log(M-L))$ measurements, under the hypothesis of $\sqrt[4]{L}\big(\min_i |x_i^{(M)}|\big)/\sqrt[4]{\log L} \to \infty$, as $M \to \infty$. Akcakaya and Tarokh consider two other performance metrics: a statistical version of the exact support recovery and a metric based on the energy of the recovered signal:

$$p_2(\hat{\mathbf{x}}, \mathbf{x}, \alpha) = 1 - \mathbb{I}\left(\frac{|\{i \mid \hat{x}_i \neq 0\} \cap \mathcal{I}|}{|\mathcal{I}|} > 1-\alpha\right)$$

$$p_3(\hat{\mathbf{x}}, \mathbf{x}, \gamma) = 1 - \mathbb{I}\left(\sum_{k \in \{i|\hat{x}_i \neq 0\} \cap \mathcal{I}} |x_k|^2 > (1-\gamma)P\right)$$

for which necessary and sufficient conditions for sparse recovery are established in [4].

In [57], asymptotically reliable sparse recovery was considered for the finite P regime. It was shown that in this regime $N = O(L\log(M-L))$ measurements are necessary, which improves on previous results. The statistical version Error Metric 2 was also considered in [102], where methods developed in [123] were used. Sparse measurement matrix ensembles instead of Gaussian measurement ensembles were considered in [97, 125]. Necessary conditions for recovery with respect to the $0-1$ loss metric were derived in [125]. Sufficient conditions for LASSO to asymptotically recover the whole support were obtained in [97]. We also note that there is other work

that characterizes the average distortion associated with compressed sensing [107], as well as that which is associated with the broader problem of sparsely representing a given vector $\mathbf{y}$ [3, 57].

The above results establish sufficient/necessary conditions on the asymptotic achievability of reliable sparse recovery for a fixed sequence of unknown sparse vectors $\mathbf{x}^{(M)}$. It is unclear whether there exists a sequence of measurement matrices for which reliable sparse recovery (and the Cramér-Rao bound) can be achieved asymptotically for *all* sequences of unknown sparse vectors $\mathbf{x}^{(M)}$. In other words, is it possible to prove similar results for a universal compressed sensing measurement setting?

In what follows, we prove that asymptotically reliable sparse recovery is possible for *all* sequences $\mathbf{x}^{(M)}$ under the hypothesis of $\min_i |x_i^{(M)}| \geq \mu_0$, for some constant μ_0, given $N > CL$ measurements, with C a constant which only depends on β, σ^2 and μ_0, in the linear sparsity regime. Moreover, we prove the parallel result for the asymptotic achievability of the Cramér-Rao bound for all sequences $\mathbf{x}^{(M)}$. These results extend the information theoretic approach to characterize sufficient conditions for *all* unknown sparse sequences.

The outline of this chapter follows next. In Section 3.2, we state the main results of this chapter and present their proofs in Section 7.3. Finally, proofs of some technical lemmas are presented in Appendix 3.4.

3.2 Main Results

Let $\{\mathbf{x}^{(M)} \in \mathbb{C}^M\}_M$ be a sequence of L-sparse vectors such that $\mathsf{supp}(\mathbf{x}^{(M)}) = \mathcal{I}^{(M)}$, and let $\{\mathbf{A}^{(M)} \in \mathbb{C}^{N \times M}\}_M$ be a sequence of measurement matrices. Let $\hat{\mathbf{x}}^{(M)}$ be an

estimate of $\mathbf{x}^{(M)}$, under the compressive sensing model of $\mathbf{y}^{(M)} = \mathbf{A}^{(M)}\mathbf{x}^{(M)} + \mathbf{n}^{(M)}$, where $\mathbf{y}^{(M)} \in \mathbb{C}^N$ is the observed vector and $\mathbf{n}^{(M)} \in \mathbb{C}^N$ is the additive i.i.d. Gaussian noise. Let $p_{\mathsf{err}}(\mathbf{A}^{(M)}) := \mathbb{E}_{\mathbf{n}}\left\{p(\hat{\mathbf{x}}^{(M)}, \mathbf{x}^{(M)})\right\}$ denote the detection error probability corresponding to the measurement matrix $\mathbf{A}^{(M)}$, where $\mathbb{E}_{\mathbf{n}}\{\cdot\}$ denotes the expectation operator defined over the density of $\mathbf{n}^{(M)}$. We say that asymptotically reliable sparse recovery is achieved by $\{\mathbf{A}^{(M)}\}_M$, if $p_{\mathsf{err}}(\mathbf{A}^{(M)}) \to 0$, as $M \to \infty$. We denote $\mathbf{A}^{(M)}$ by an i.i.d. Gaussian measurement matrix, if each element of $\mathbf{A}^{(M)}$ is drawn i.i.d. according to the normal distribution $\mathcal{N}(0,1)$. For ease of notation, we will suppress the dependence on M when there is no ambiguity. Let $H(x) := -x\log x - (1-x)\log(1-x)$ denote the binary entropy function.

Finally, let

$$X^{(M)} := \Big\{\mathbf{x} \in \mathbb{C}^M \ : \ \min_i |x_i| \geq \mu_0, \text{ and } \|\mathbf{x}\|_0 = L\Big\},$$

where μ_0 is an arbitrary but fixed positive constant.

The main result of this chapter is the following theorem:

Theorem 3.2.1 (Main Theorem)**.** *Let* $\left\{\mathbf{x}^{(M)} \in X^{(M)}\right\}_M$ *be a sequence of unknown L-sparse vectors. If*

$$N > CL$$

where $C = \max\{C_1, C_2, C_3\}$*, with*

$$C_1 := \Big(\frac{1}{2} + \frac{\delta}{\sigma^2}\Big)^{-1}\left(1 - \sqrt{1 - \frac{1}{\beta H\big(\frac{1}{\beta}\big)\big(2 + \frac{\sigma^2}{\delta}\big)^2}}\right)^{-1},$$

$$C_2 := 9 + 4\log(\beta - 1) + 4\beta H\big(\tfrac{1}{\beta}\big),$$

$$C_3 := \left(\frac{\sqrt{2}}{\sqrt{1+\epsilon}-1}(1+\epsilon)\left(1+\sqrt{\beta H(\tfrac{2}{\beta})}\right)\right)^2,$$

$\epsilon \in (0, \frac{3-\sqrt{5}}{2})$ *a fixed constant and* $\delta := \frac{1}{2}\frac{(1-\epsilon-\frac{\epsilon}{1-\epsilon})^2}{(1+\epsilon)}\mu_0^2$, *then any sequence of* $N \times M$ *i.i.d Gaussian measurement matrices* $\left\{\mathbf{A}^{(M)}\right\}_M$ *achieves asymptotically reliable sparse recovery for any sequence* $\left\{\mathbf{x}^{(M)} \in X^{(M)}\right\}_M$, *with overwhelming probability.*

We also have the following corollary regarding the asymptotic achievability of the Cramér-Rao bound:

Corollary 3.2.2 (Achievability of the Cramér-Rao Bound)**.** *Let* $\left\{\mathbf{x}^{(M)} \in X^{(M)}\right\}_M$ *be any sequence of unknown L-sparse vectors, such that* $\|\mathbf{x}^{(M)}\|_2^2$ *grows at most polynomially in L. Let* $e_\delta^{(M)}$ *and* $e_G^{(M)}$ *denote the MSE of the joint typicality decoder and the Genie-aided decoder, respectively. Then, under the conditions of Theorem 7.2, we have*

$$\limsup_M \left|e_\delta^{(M)} - e_G^{(M)}\right| = 0.$$

with overwhelming probability, for any sequence of $N \times M$ *i.i.d. Gaussian measurement matrices* $\{\mathbf{A}^{(M)}\}_M$.

3.3 Proofs of the Main Results

Recall the definition of the joint typicality as in [4]: we say an $N \times 1$ noisy observation vector, $\mathbf{y} = \mathbf{A}\mathbf{x}+\mathbf{n}$ and a set of indices $\mathcal{J} \subset \{1, 2, \cdots, M\}$, with $|\mathcal{J}| = L$, are δ-jointly typical if $\text{rank}(\mathbf{A}_\mathcal{J}) = L$ and

$$\left|\frac{1}{N}\|\mathbf{\Pi}_{\mathbf{A}_\mathcal{J}}^\perp \mathbf{y}\|^2 - \frac{N-L}{N}\sigma^2\right| < \delta \tag{3.1}$$

where $\mathbf{A}_{\mathcal{J}}$ is the sub-matrix of $\mathbf{A}$ with columns corresponding to the index set $\mathcal{J}$ and $\mathbf{\Pi}^{\perp}_{\mathbf{A}_{\mathcal{J}}} := \mathbf{I} - \mathbf{A}_{\mathcal{J}}(\mathbf{A}^*_{\mathcal{J}}\mathbf{A}_{\mathcal{J}})^{-1}\mathbf{A}^*_{\mathcal{J}}$. We also define the following event:

$$E_{\mathcal{J}}(\mathbf{x}) = \{\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \text{ is } \delta\text{-jointly typical with } \mathcal{J}\}, \tag{3.2}$$

for all $\mathcal{J} \in \{1, 2, \cdots, M\}$ with $|\mathcal{J}| = L$.

Our strategy for proving the Main Theorem has two steps: 1) we show that any sequence of i.i.d. Gaussian measurement matrices achieves asymptotically reliable sparse recovery for a carefully chosen *canonical* set of unknown sparse vectors, with overwhelming probability, and 2) we show that by this particular choice of the *canonical* set, and by using the asymptotic Restricted Isometry Property (RIP) of i.i.d. Gaussian matrices [29], the property proved in step (1) is carried over to the set $X^{(M)}$, asymptotically. Note that this proof technique is reminiscent of the proofs for achievability of the capacity of compound channels (See, for example [130]).

3.3.1 The Canonical Set

Let $\mathbf{x}_{\mathcal{J}}$ denote the restriction of a vector $\mathbf{x}$ to the index set $\mathcal{J} \subset \{1, 2, \cdots, M\}$. Let

$$X_c^{(M)} := \Big\{\mathbf{x}_c^{(\mathcal{K})} \in \mathbb{R}^M \quad : \quad (\mathbf{x}_c^{(\mathcal{K})})_i = c_\epsilon \mu_0 \text{ for all } i \in \mathcal{K}, \\ (\mathbf{x}_c^{(\mathcal{K})})_i = 0 \text{ for all } i \notin \mathcal{K}, \forall \mathcal{K} \text{ s.t. } |\mathcal{K}| = L\Big\},$$

where

$$c_\epsilon := \sqrt{\frac{1 - \epsilon - \frac{\epsilon}{1-\epsilon}}{1+\epsilon}}. \tag{3.3}$$

For instance, for $M = 3$ and $L = 2$, $X_c^{(3)}$ consists of the vectors $(0, c_\epsilon\mu_0, c_\epsilon\mu_0)^T$, $(c_\epsilon\mu_0, 0, c_\epsilon\mu_0)^T$ and $(c_\epsilon\mu_0, c_\epsilon\mu_0, 0)^T$. We denote $X_c^{(M)}$ by the *canonical* set.

The following lemma establishes step (1) of the proof of the Main Theorem:

Lemma 3.3.1. *If*

$$N > CL$$

where $C = \max\{C_1, C_2, C_3\}$, *with*

$$C_1 := \Big(\frac{1}{2} + \frac{\delta}{\sigma^2}\Big)^{-1}\left(1 - \sqrt{1 - \frac{1}{\beta H\big(\frac{1}{\beta}\big)\big(2 + \frac{\sigma^2}{\delta}\big)^2}}\right)^{-1},$$

$$C_2 := 9 + 4\log(\beta - 1) + 4\beta H\big(\tfrac{1}{\beta}\big),$$

$$C_3 := \left(\frac{\sqrt{2}}{\sqrt{1+\epsilon} - 1}(1+\epsilon)\Big(1 + \sqrt{\beta H(\tfrac{2}{\beta})}\Big)\right)^2,$$

$\epsilon \in (0, \frac{3-\sqrt{5}}{2})$ *a fixed constant and* $\delta := \frac{1}{2}\frac{(1-\epsilon-\frac{\epsilon}{1-\epsilon})^2}{(1+\epsilon)}\mu_0^2$, *then any sequence of i.i.d. Gaussian measurement matrices* $\mathbf{A}^{(M)}$ *achieves asymptotically reliable sparse recovery for all* $\mathbf{x} \in X_c^{(M)}$ *and* $\frac{1}{\sqrt{N}}\mathbf{A}^{(M)}$ *satisfies the Restricted Isometry Property (RIP) of order* $2L$ *with* $\epsilon_{2L} = \epsilon$, *with overwhelming probability.*

Proof. Clearly,

$$\big|X_c^{(M)}\big| = \binom{M}{L} \leq \exp\Big(\beta H(1/\beta)L\Big) \tag{3.4}$$

using the Stirling's bound. According to the proof of Theorem 1.1 of [4], the average error probability of the joint typicality decoder can be bounded as follows:

$$\mathbb{E}_{\mathbf{A}}\Big\{p_{\mathsf{err}}\big(\mathbf{A}^{(M)}\big)\Big\} \leq \exp\big(-\phi(L)\big) := \sum_{\mathbf{x}\in X_c^{(M)}} 2\exp\left(-\frac{\delta^2}{4\sigma^4}\frac{N^2}{N - L + \frac{2\delta}{\sigma^2}N}\right)$$
$$+ \sum_{\mathbf{x}\in X_c^{(M)}} \exp\Big(\log L - g(L)\Big), \tag{3.5}$$

where $g(L) := \min\{f(1/L), f(1)\}$, with

$$f(1/L) := -2\log L - 2 - \log(\beta - 1) + C_0 L\left(\Big(\frac{\mu^2(\mathbf{x}) - \delta'}{\mu^2(\mathbf{x}) + \sigma^2}\Big)^2\right)$$

and

$$f(1) = -L(2 + \log(\beta - 1)) + C_0 L\bigg(\Big(\frac{L\mu^2(\mathbf{x}) - \delta'}{L\mu^2(\mathbf{x}) + \sigma^2} \Big)^2 \bigg),$$

$C_0 := (N - L)/4L$, $\mu(\mathbf{x}) := \min_i |x_i|$, and $\delta' = \delta/(1 - \alpha)$, given

$$\delta \leq (1 - \alpha) c_\epsilon^2 \mu_0^2. \tag{3.6}$$

Clearly, the choice of $\delta = \frac{1}{2} \frac{(1 - \epsilon - \frac{\epsilon}{1-\epsilon})^2}{(1+\epsilon)} \mu_0^2$ satisfies Eq. (3.6) due to the necessary bound of $\alpha < 1/2$. Note that the summation over $\mathbf{x} \in X_c^{(M)}$ results from the union bound in bounding the error for *all* $\mathbf{x} \in X_c^{(M)}$. Hence,

$$\begin{aligned} \mathbb{E}_{\mathbf{A}} \Big\{ p_{\mathsf{err}}\big(\mathbf{A}^{(M)}\big) \Big\} \;\; \leq \;\; & 2 \exp\Big(- \frac{\delta^2}{4\sigma^4} \frac{N^2}{N - L + \frac{2\delta}{\sigma^2} N} + L\beta H\big(\tfrac{1}{\beta}\big) \Big) \\ & + \exp\Big(\log L - g(L) + L\beta H\big(\tfrac{1}{\beta}\big) \Big). \end{aligned} \tag{3.7}$$

Now, by the hypothesis of $N > C_1 L$, we get

$$\frac{\delta^2}{4\sigma^4} \frac{N^2}{N - L + \frac{2\delta}{\sigma^2} N} > L\beta H\big(\tfrac{1}{\beta}\big). \tag{3.8}$$

Hence, the exponent of the first term on the right hand side of Eq. (7.35) tends to $-\infty$, as $M \to \infty$. Moreover, the hypothesis of $N > C_2 L$ yields

$$C_0 > 2 + \log(\beta - 1) + \beta H\big(\tfrac{1}{\beta}\big), \tag{3.9}$$

which implies that the exponent of the second term on the right hand side of Eq. (7.35) tends to $-\infty$ as well.

Now, by the Markov's inequality we have:

$$\mathbb{P}\Big(p_{\mathsf{err}}\big(\mathbf{A}^{(M)}\big) > \exp\big(- \phi(L)/2 \big) \Big) \leq \exp\big(- \phi(L)/2 \big). \tag{3.10}$$

for $\mathbf{A}$ in the Gaussian ensemble, where $\phi(L)$ is as in Eq. (3.5), and tends to ∞ as $L \to \infty$. Furthermore, by the hypothesis of $N > C_3 L$ and the results of [29], $\frac{1}{\sqrt{N}}\mathbf{A}^{(M)}$ satisfies the RIP or order $2L$ with $\epsilon_{2L} = \epsilon$. Therefore, we have

$$N(1-\epsilon)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}_{\mathcal{K}}\mathbf{x}\|_2^2 \leq N(1+\epsilon)\|\mathbf{x}\|_2^2, \tag{3.11}$$

for any $\mathcal{K}$ such that $|\mathcal{K}| = 2L$, with probability exceeding

$$1 - 2\exp\Big(-\frac{1}{2}L\beta H\big(\tfrac{2}{\beta}\big)\epsilon\Big). \tag{3.12}$$

Hence, any i.i.d. Gaussian measurement matrix $\mathbf{A}^{(M)}$ satisfies

$$p_{\mathsf{err}}\big(\mathbf{A}^{(M)}\big) \leq \exp\big(-\phi(L)/2\big) \tag{3.13}$$

and $\frac{1}{\sqrt{N}}\mathbf{A}^{(M)}$ satisfies the RIP of order $2L$ with $\epsilon_{2L} = \epsilon$, with probability exceeding

$$1 - \exp\big(-\phi(L)/2\big) - 2\exp\Big(-\frac{1}{2}L\beta H\big(\tfrac{2}{\beta}\big)\epsilon\Big). \tag{3.14}$$

This establishes the claim of the lemma. □

3.3.2 Two Technical Lemmas

In order to move on with the proof of the Main Theorem, we need to state the following technical lemmas:

Lemma 3.3.2. *Let $\frac{1}{\sqrt{N}}\check{\mathbf{A}} \in \mathbb{C}^{N\times M}$ be a matrix satisfying the RIP of order $2L$ with $\epsilon_{2L} = \epsilon$. Let*

$$\lambda(\mathbf{x}) := \frac{1}{\sigma^2}\big\|\mathbf{K}_{\mathcal{I},\mathcal{J}}\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2,$$

where $\mathbf{K}_{\mathcal{I},\mathcal{J}} := \mathbf{\Pi}^{\perp}_{\check{\mathbf{A}}_{\mathcal{J}}}\check{\mathbf{A}}_{\mathcal{I}\setminus\mathcal{J}}$, for some $\mathcal{I}$ and $\mathcal{J} \neq \mathcal{I}$ such that $|\mathcal{I}| = |\mathcal{J}| = L$. Then, for all $\mathbf{x} \in X^{(M)}$ with $\mathsf{supp}(\mathbf{x}) = \mathcal{I}$, we have:

$$\frac{N}{\sigma^2}\frac{\big(1-\epsilon-\frac{\epsilon}{1-\epsilon}\big)^2}{(1+\epsilon)}\mu_0^2 \leq \lambda\big(\mathbf{x}_c^{(\mathcal{I})}\big) \leq \lambda(\mathbf{x}).$$

Proof. The proof is given in Appendix 3.4. □

Lemma 3.3.3. *Let $F(z;\nu,\lambda)$ and $f(z;\nu,\lambda)$ denote the cumulative distribution function and density of a non-central χ^2 random variable of order ν with non-centrality parameter λ, respectively. Let $0 \leq \lambda_1 \leq \lambda_2$ be two non-negative constants. Let a and b be two constants satisfying*

$$0 \leq a \leq b \leq \nu + \frac{\nu}{\nu+2}\lambda_1.$$

Then, we have

$$F(b;\nu,\lambda_2) - F(a;\nu,\lambda_2) \leq F(b;\nu,\lambda_1) - F(a;\nu,\lambda_1).$$

Proof. The proof is given in Appendix 3.4. □

3.3.3 Proof of the Main Theorem

Let $\{\check{\mathbf{A}}^{(M)}\}_M$ be a sequence of measurement matrices for which asymptotically reliable sparse recovery can be achieved for all $\mathbf{x} \in X_c^{(M)}$, and that $\frac{1}{\sqrt{N}}\check{\mathbf{A}}^{(M)}$ satisfies the RIP of order $2L$ with $\epsilon_{2L} = \epsilon$. This choice of $\{\check{\mathbf{A}}^{(M)}\}_M$ is clearly possible by the virtue of Lemma 3.3.1.

Recall the error events $E_{\mathcal{I}}^c$ and $E_{\mathcal{J}}$ given by:

$$E_{\mathcal{I}}^c(\mathbf{x}) = \{\mathbf{y} = \check{\mathbf{A}}^{(M)}\mathbf{x} + \mathbf{n} \text{ is not } \delta\text{-jointly typical with } \mathcal{I}\}$$

and

$$E_{\mathcal{J}}(\mathbf{x}) = \{\mathbf{y} = \check{\mathbf{A}}^{(M)}\mathbf{x} + \mathbf{n} \text{ is } \delta\text{-jointly typical with } \mathcal{J}\}.$$

We will show that the following statement holds: if the joint typicality decoder asymptotically achieves reliable sparse recovery for all $\mathbf{x} \in X_c^{(M)}$ using the sequence of measurement matrices $\{\check{\mathbf{A}}^{(M)}\}_M$, so it does for all $\mathbf{x} \in X^{(M)}$. More formally, let $\mathbf{x} \in X^{(M)}$, with $\mathsf{supp}(\mathbf{x}) = \mathcal{I}$. Then, we have the following proposition:

Proposition 3.3.4. *For any $\mathbf{x} \in X^{(M)}$, we have*

$$\mathbb{P}\big(E_{\mathcal{I}}^c(\mathbf{x})\big) = \mathbb{P}\big(E_{\mathcal{I}}^c(\mathbf{x}_c^{(\mathcal{I})})\big),$$

and

$$\mathbb{P}\big(E_{\mathcal{J}}(\mathbf{x})\big) \leq \mathbb{P}\big(E_{\mathcal{J}}(\mathbf{x}_c^{(\mathcal{I})})\big).$$

Clearly, Proposition 3.3.4 establishes the second step of the proof of the Main Theorem, which in turn completes the proof. In what follows, we prove the two claims of Proposition 3.3.4.

Proof of Proposition 3.3.4. First, consider the error event $E_{\mathcal{I}}^c$. We have

$$\mathbb{P}\big(E_{\mathcal{I}}^c(\mathbf{x})\big) = \mathbb{P}\Big(\big|\theta - (N-L)\sigma^2\big| > N\delta\Big) \tag{3.15}$$

where θ is a χ^2 random variable with $N-L$ degrees of freedom, with mean $(N-L)\sigma^2$ and variance $2(N-L)\sigma^4$. Note that this probability is independent of $\check{\mathbf{A}}^{(M)}$ and $\mathbf{x}$. Hence,

$$\mathbb{P}\big(E_{\mathcal{I}}^c(\mathbf{x})\big) = \mathbb{P}\big(E_{\mathcal{I}}^c(\mathbf{x}_c^{(\mathcal{I})})\big), \tag{3.16}$$

which establishes the first claim of the proposition.

Next, consider the error event $E_{\mathcal{J}}$, for some $\mathcal{J} \neq \mathcal{I}$. That is,

$$\left|\frac{1}{N}\left\|\mathbf{\Pi}^{\perp}_{\check{\mathbf{A}}_{\mathcal{J}}}\mathbf{y}\right\|_2^2 - \frac{N-L}{N}\sigma^2\right| < \delta. \tag{3.17}$$

It is easy to show that

$$\mathbb{P}\big(E_{\mathcal{J}}(\mathbf{x})\big) = \mathbb{P}\Big(\big|\vartheta_{\mathcal{J}} - (N-L)\big| < N\frac{\delta}{\sigma^2}\Big), \tag{3.18}$$

where $\vartheta_{\mathcal{J}}$ is a non-central χ^2 random variable with non-centrality parameter

$$\lambda(\mathbf{x}) := \frac{1}{\sigma^2}\Big\|\mathbf{K}_{\mathcal{I},\mathcal{J}}\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}}\Big\|_2^2,$$

where

$$\mathbf{K}_{\mathcal{I},\mathcal{J}} := \mathbf{\Pi}^{\perp}_{\check{\mathbf{A}}_{\mathcal{J}}}\check{\mathbf{A}}_{\mathcal{I}\setminus\mathcal{J}}. \tag{3.19}$$

Letting $F(z;\nu,\lambda)$ denote the cumulative distribution function of a non-central χ^2 random variable of order ν and non-centrality parameter λ, we have:

$$\mathbb{P}\big(E_{\mathcal{J}}(\mathbf{x})\big) = F\Big(N-L+N\frac{\delta}{\sigma^2}; N-L, \lambda(\mathbf{x})\Big) - F\Big(N-L-N\frac{\delta}{\sigma^2}; N-L, \lambda(\mathbf{x})\Big).$$

Next, we let

$$a = N-L-N\frac{\delta}{\sigma^2}, \quad b = N-L+N\frac{\delta}{\sigma^2},$$

$$\lambda_1 = \lambda\big(\mathbf{x}_c^{(\mathcal{I})}\big), \quad \lambda_2 = \lambda(\mathbf{x}),$$

and $\nu = N-L$, in Lemma 3.3.3. We have

$$N\frac{\delta}{\sigma^2} = \frac{1}{2\sigma^2}N\frac{\big(1-\epsilon-\frac{\epsilon}{1-\epsilon}\big)^2}{(1+\epsilon)}\mu_0^2 \le \frac{1}{2}\lambda\big(\mathbf{x}_c^{(\mathcal{I})}\big), \tag{3.20}$$

due to the lower bound of Lemma 3.3.2. Hence,

$$\begin{aligned} b = N-L+N\frac{\delta}{\sigma^2} &\le N-L+\frac{1}{2}\lambda\big(\mathbf{x}_c^{(\mathcal{I})}\big) \\ &\le N-L+\frac{N-L}{N-L+2}\lambda\big(\mathbf{x}_c^{(\mathcal{I})}\big) \\ &= \nu+\frac{\nu}{\nu+2}\lambda_1 \end{aligned}$$

for any $L \geq 2$. Moreover, by Lemma 3.3.2, we have $\lambda(\mathbf{x}) \geq \lambda\big(\mathbf{x}_c^{(\mathcal{I})}\big)$. Hence, Lemma 3.3.3 implies that:

$$\begin{aligned}\mathbb{P}\big(E_{\mathcal{J}}(\mathbf{x})\big) &= F\Big(N - L + N\frac{\delta}{\sigma^2}; N - L, \lambda(\mathbf{x})\Big) - F\Big(N - L - N\frac{\delta}{\sigma^2}; N - L, \lambda(\mathbf{x})\Big) \\ &\leq F\Big(N - L + N\frac{\delta}{\sigma^2}; N - L, \lambda\big(\mathbf{x}_c^{(\mathcal{I})}\big)\Big) - F\Big(N - L - N\frac{\delta}{\sigma^2}; N - L, \lambda\big(\mathbf{x}_c^{(\mathcal{I})}\big)\Big) \\ &= \mathbb{P}\Big(E_{\mathcal{J}}\big(\mathbf{x}_c^{(\mathcal{I})}\big)\Big),\end{aligned}$$

which establishes the second claim of Proposition 3.3.4. □

3.3.4 Proof of Corollary 3.2.2

Proof of Corollary 3.2.2 closely parallels that of the main result of Chapter 1, with the corresponding modifications in the proof of the Main Theorem of this chapter, and is thus omitted for brevity.

3.4 Proof of Lemmas 3.3.2 and 3.3.3

Proof of Lemma 3.3.2. First, we obtain lower and upper bounds on the eigenvalues of the operator $\mathbf{K}_{\mathcal{I},\mathcal{J}}$. We have:

$$\begin{aligned}\big\|\mathbf{\Pi}^{\perp}_{\check{\mathbf{A}}_{\mathcal{J}}}\check{\mathbf{A}}_{\mathcal{I}\setminus\mathcal{J}}\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2 \leq \big\|\check{\mathbf{A}}_{\mathcal{I}\setminus\mathcal{J}}\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2 &\leq N(1+\epsilon_{|\mathcal{I}\setminus\mathcal{J}|})\big\|\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2 \\ &\leq N(1+\epsilon_L)\big\|\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2 \\ &\leq N(1+\epsilon)\big\|\mathbf{x}_{\mathcal{I}\setminus\mathcal{J}}\big\|_2^2, \qquad (3.21)\end{aligned}$$

and

$$\begin{aligned}\left\|\mathbf{\Pi}^{\perp}_{\check{\mathbf{A}}_{\mathcal{J}}}\check{\mathbf{A}}_{\mathcal{I}\backslash\mathcal{J}}\mathbf{x}_{\mathcal{I}\backslash\mathcal{J}}\right\|_2^2 &= \left\|\check{\mathbf{A}}_{\mathcal{I}\backslash\mathcal{J}}\mathbf{x}_{\mathcal{I}\backslash\mathcal{J}}\right\|_2^2 - \left\|\mathbf{\Pi}_{\check{\mathbf{A}}_{\mathcal{J}}}\check{\mathbf{A}}_{\mathcal{I}\backslash\mathcal{J}}\mathbf{x}_{\mathcal{I}\backslash\mathcal{J}}\right\|_2^2 \\ &\geq N(1-\epsilon_{|\mathcal{I}\backslash\mathcal{J}|})\left\|\mathbf{x}_{\mathcal{I}\backslash\mathcal{J}}\right\|_2^2 - \left\|(\check{\mathbf{A}}^*_{\mathcal{J}}\check{\mathbf{A}}_{\mathcal{J}})^{-1/2}\check{\mathbf{A}}^*_{\mathcal{J}}\check{\mathbf{A}}_{\mathcal{I}\backslash\mathcal{J}}\mathbf{x}_{\mathcal{I}\backslash\mathcal{J}}\right\|_2^2 \\ &\geq N(1-\epsilon_{|\mathcal{I}\backslash\mathcal{J}|})\left\|\mathbf{x}_{\mathcal{I}\backslash\mathcal{J}}\right\|_2^2 - N\frac{\epsilon_{L+|\mathcal{I}\backslash\mathcal{J}|}}{1-\epsilon_L}\left\|\mathbf{x}_{\mathcal{I}\backslash\mathcal{J}}\right\|_2^2 \\ &\geq N\Big(1-\epsilon_L-\frac{\epsilon_{2L}}{1-\epsilon_L}\Big)\left\|\mathbf{x}_{\mathcal{I}\backslash\mathcal{J}}\right\|_2^2 \\ &\geq N\Big(1-\epsilon-\frac{\epsilon}{1-\epsilon}\Big)\left\|\mathbf{x}_{\mathcal{I}\backslash\mathcal{J}}\right\|_2^2, \end{aligned} \tag{3.22}$$

since $\epsilon_L < \epsilon_{2L} = \epsilon$, by hypothesis. Now, we can lower-bound $\lambda\big(\mathbf{x}_c^{(\mathcal{I})}\big)$ as follows:

$$\begin{aligned}\lambda(\mathbf{x}_c^{(\mathcal{I})}) \geq \frac{N}{\sigma^2}\Big(1-\epsilon-\frac{\epsilon}{1-\epsilon}\Big)\left\|(\mathbf{x}_c^{(\mathcal{I})})_{\mathcal{I}\backslash\mathcal{J}}\right\|_2^2 &= \frac{N}{\sigma^2}\Big(1-\epsilon-\frac{\epsilon}{1-\epsilon}\Big)|\mathcal{I}\backslash\mathcal{J}|c_\epsilon^2\mu_0^2 \\ &\geq \frac{N}{\sigma^2}\frac{\big(1-\epsilon-\frac{\epsilon}{1-\epsilon}\big)^2}{(1+\epsilon)}\mu_0^2. \end{aligned} \tag{3.23}$$

since $|\mathcal{I}\backslash\mathcal{J}| \geq 1$. Similarly, an upper bound for $\lambda\big(\mathbf{x}_c^{(\mathcal{I})}\big)$ can be obtained as

$$\begin{aligned}\lambda\big(\mathbf{x}_c^{(\mathcal{I})}\big) \leq \frac{1}{\sigma^2}N(1+\epsilon)\left\|(\mathbf{x}_c^{(\mathcal{I})})_{\mathcal{I}\backslash\mathcal{J}}\right\|_2^2 &= \frac{1}{\sigma^2}N|\mathcal{I}\backslash\mathcal{J}|(1+\epsilon)c_\epsilon^2\mu_0^2 \\ &= \frac{1}{\sigma^2}N|\mathcal{I}\backslash\mathcal{J}|(1+\epsilon)\frac{\big(1-\epsilon-\frac{\epsilon}{1-\epsilon}\big)}{\big(1+\epsilon\big)}\mu_0^2 \\ &= \frac{1}{\sigma^2}N|\mathcal{I}\backslash\mathcal{J}|\Big(1-\epsilon-\frac{\epsilon}{1-\epsilon}\Big)\mu_0^2. \end{aligned} \tag{3.24}$$

Finally, for any $\mathbf{x} \in X^{(M)}$, we have

$$\lambda(\mathbf{x}) \geq \frac{1}{\sigma^2}N\Big(1-\epsilon-\frac{\epsilon}{1-\epsilon}\Big)\|\mathbf{x}_{\mathcal{I}\backslash\mathcal{J}}\|_2^2 \geq \frac{1}{\sigma^2}N|\mathcal{I}\backslash\mathcal{J}|\Big(1-\epsilon-\frac{\epsilon}{1-\epsilon}\Big)\mu_0^2,$$

which establishes the claim of the lemma. □

Proof of Lemma 3.3.3. It can be shown that [39]

$$\frac{\partial F(z;\nu,\lambda)}{\partial\lambda} = -f(z;\nu+2,\lambda) \tag{3.25}$$

for all non-negative z and λ. Therefore, we have

$$\Big(F(b;\nu,\lambda_1)-F(a;\nu,\lambda_1)\Big)-\Big(F(b;\nu,\lambda_2)-F(a;\nu,\lambda_2)\Big) \tag{3.26}$$

$$\begin{aligned}
&=\Big(F(b;\nu,\lambda_1)-F(b;\nu,\lambda_2)\Big)-\Big(F(a;\nu,\lambda_1)-F(a;\nu,\lambda_2)\Big)\\
&=\int_{\lambda_2}^{\lambda_1}\frac{\partial F(b;\nu,\xi)}{\partial\xi}d\xi-\int_{\lambda_2}^{\lambda_1}\frac{\partial F(a;\nu,\xi)}{\partial\xi}d\xi\\
&=\int_{\lambda_1}^{\lambda_2}f(b;\nu+2,\xi)d\xi-\int_{\lambda_1}^{\lambda_2}f(a;\nu+2,\xi)d\xi\\
&=\int_{\lambda_1}^{\lambda_2}\Big(f(b;\nu+2,\xi)-f(a;\nu+2,\xi)\Big)d\xi. \qquad (3.27)
\end{aligned}$$

Therefore, it is enough to show that

$$f(b;\nu+2,\xi)\geq f(a;\nu+2,\xi) \tag{3.28}$$

for all $\xi\in[\lambda_1,\lambda_2]$.

Let $M(\nu,\lambda)$ denote the mode of the density $f(z;\nu,\lambda)$. It can be shown that [110]

$$M(\nu+2,\xi)\geq\nu+\frac{\nu}{\nu+2}\xi. \tag{3.29}$$

Also, it is known that the non-central χ^2 distribution is uni-modal [42]. Therefore, for any a and b satisfying $a\leq b\leq\nu+\frac{\nu}{\nu+2}\lambda_1\leq\nu+\frac{\nu}{\nu+2}\xi$, we have

$$f(b;\nu+2,\xi)\geq f(a;\nu+2,\xi), \tag{3.30}$$

for all $\xi\in[\lambda_1,\lambda_2]$, which proves the statement of the lemma. □

Chapter 4

SPARLS: The Sparse RLS Algorithm

4.1 Introduction

This chapter deals with adaptive identification of sparse systems. Many real-life systems admit sparse representations with few non-zero coefficients. Examples include multipath wireless communication channels where reflections reach the receiver with long delays, imaging, video, etc. [23]. Many of the above applications require adaptive estimation techniques with minimum computational complexity due to time-varying dynamics and a large number of potential parameters. Wireless communication channels provide a typical representative of the above setup. The wireless channel is described by sparse fading rays and long zero samples and thus admits a sparse representation [16]. If power amplifiers at the transmitter and receiver ends operate in the linear regime, the channel is represented by a time-varying linear filter

whose unit sample response is a sparse vector. Adaptive algorithms such as the Least Mean Squares (LMS), the Recursive Least Squares (RLS) and fast variants thereof have been widely applied for estimation/equalization of such channels.

Recently, compressive sensing methods have been developed for the estimation of the multipath channels taking into account the sparseness characteristic [7, 12, 16, 33, 36, 122]. Bajwa et. al [16] used the Dantzig Selector [30] and Least Squares (LS) estimates for the problem of sparse channel sensing. Although the Dantzig Selector and the LS method produce sparse estimates with improved MSE, they do not exploit the sparsity of the underlying signal in order to reduce the computational complexity. Moreover, they are not appropriate for the setting of streaming data. In [36], two different sparsity constraints are incorporated into the quadratic cost function of the LMS algorithm, to take into account the sparse channel coefficient vector. An alternative viewpoint to sparse LMS, is the proportionate Normalized LMS and its variants [51], which assign different step sizes to different coefficients based on their optimal magnitudes. An ℓ_1-regularized RLS type algorithm based on a low complexity Expectation-Maximization is derived in [12]. In [7] the weighted Least Absolute Shrinkage and Selection Operator (LASSO) estimates are retrieved recursively using a system of normal equations or via iterative subgradient methods. Previously reported algorithms for sparse system identification using Kalman filtering rely on the Dantzig Selector [122] and on the pseudo-measurement technique [33]. In [122] the proposed algorithm is not exclusively based on Kalman filtering as it requires the implementation of an additional optimization algorithm (the Dantzig selector), which leads to increased complexity and execution time. The sparse Kalman filtering approach

in [33], first performs a Kalman iteration then it generates a fictitious observation from the ℓ_1-reguralization constraint and carries out a Kalman filter pseudo-measurement update. The computational cost of the Kalman filter pseudo-measurement update is avoided by the proposed technique.

We introduce a Recursive ℓ_1-Regularized Least Squares (SPARLS) algorithm for adaptive filtering setup [10, 12]. The SPARLS algorithm is based on an Expectation Maximization (EM) type algorithm presented in [55] and produces successive improved estimates based on streaming data. We present analytical results for the convergence and the steady state Mean Squared Error (MSE), which reveal the significant MSE gain of the SPARLS algorithm. Simulation studies show that the SPARLS algorithm significantly outperforms the RLS algorithm in terms of MSE, for both static (with finite samples) and time-varying signals. Moreover, these simulation results suggest that the computational complexity of the SPARLS algorithm (with slight modifications) can be less than that of the RLS algorithm, for tap-weight vectors with fixed support. In particular, for estimating a time-varying Rayleigh fading wireless channel with 5 nonzero coefficients, the SPARLS algorithm gains about 7dB over the RLS algorithm in MSE and has about 80% less computational complexity.

The outline of the chapter is as follows: we will present the mathematical preliminaries and problem statement in Section 4.2. We will formally define the SPARLS algorithm in Section 4.3, followed by analytical results regarding convergence, steady state error, error performance comparison between SPARLS and RLS, complexity and storage issues, and parameter adjustments in Section 4.4. Finally, simulation studies are presented in Section 4.5.

4.2 Mathematical Preliminaries and Problem Statement

4.2.1 Adaptive Filtering Setup

Consider the conventional adaptive filtering setup, consisting of a transversal filter followed by an adaptation block. The tap-input vector at time i is defined by

$$\mathbf{x}(i) := [x(i), x(i-1), \cdots, x(i-M+1)]^T \tag{4.1}$$

where $x(k)$ is the input at time k, $k = 1, \cdots, n$. The tap-weight vector at time n is defined by

$$\hat{\mathbf{w}}(n) := [\hat{w}_0(n), \hat{w}_1(n), \cdots, \hat{w}_{M-1}(n)]^T. \tag{4.2}$$

The output of the filter at time i is given by

$$y(i) := \hat{\mathbf{w}}^*(n)\mathbf{x}(i). \tag{4.3}$$

where $(\cdot)^*$ denotes the conjugate transpose operator. Let $d(i)$ be the desired output of the filter at time i. We can define the instantaneous error of the filter by

$$e(i) := d(i) - y(i) = d(i) - \hat{\mathbf{w}}^*(n)\mathbf{x}(i). \tag{4.4}$$

The operation of the adaptation block at time n can therefore be stated as the following optimization problem:

$$\min_{\hat{\mathbf{w}}(n)} \quad f\big(e(1), e(2), \cdots, e(n)\big), \tag{4.5}$$

where $f \geq 0$ is a certain cost function. In particular, if $d(i)$ is generated by an unknown tap-weight $\mathbf{w}(n)$, *i.e.*, $d(i) = \mathbf{w}^*(n)\mathbf{x}(i)$, with an appropriate choice of f,

one can possibly obtain a good approximation to $\mathbf{w}(n)$ by solving the optimization problem given in (4.5). This is, in general, a system identification problem and is the topic of interest in this chapter[1].

As an example, one can define the cost function as follows:

$$f_{RLS}\big(e(1), e(2), \cdots, e(n)\big) := \sum_{i=1}^{n} \lambda^{n-i} |e(i)|^2. \tag{4.6}$$

with λ a non-negative constant. The parameter λ is commonly referred to as *forgetting factor*. The solution to the optimization problem in Eq. (4.5) with f_{RLS} gives rise to the well-known Recursive Least Squares (RLS) algorithm (See, for example, [65]). The RLS Algorithm can be summarized as follows:

Algorithm 1 RLS

Inputs: $\mathbf{P}(0) = \delta^{-1}\mathbf{I}$ for some small positive constant δ and $\hat{\mathbf{w}}(0) = 0$.

Output: $\hat{\mathbf{w}}(N)$.

1: **for** $n = 1, 2, \cdots, N$ **do**

2: $\quad \mathbf{k}(n) = \dfrac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n)}{1 + \lambda^{-1}\mathbf{x}^*(n)\mathbf{P}(n-1)\mathbf{x}(n)}.$

3: $\quad e(n) = d(n) - \hat{\mathbf{w}}^*(n-1)\mathbf{x}(n).$

4: $\quad \hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)e^*(n).$

5: $\quad \mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^*(n)\mathbf{P}(n-1).$

6: **end for**

The cost function f_{RLS} given in (4.6) corresponds to a least squares identification problem. Let

$$\mathbf{D}(n) := \mathrm{diag}(\lambda^{n-1}, \lambda^{n-2}, \cdots, 1), \tag{4.7}$$

[1]Our discussion will focus on single channel complex valued signals. The extension to the multi-variable case presents no difficulties.

$$\mathbf{d}(n) := [d^*(1), d^*(2), \cdots, d^*(n)]^T \tag{4.8}$$

and $\mathbf{X}(n)$ be an $n \times M$ matrix whose ith row is $\mathbf{x}^*(i)$, *i.e.*,

$$\mathbf{X}(n) := \begin{pmatrix} \mathbf{x}^*(1) \\ \vdots \\ \mathbf{x}^*(n-1) \\ \mathbf{x}^*(n) \end{pmatrix}. \tag{4.9}$$

The RLS cost function can be written in the following form:

$$f_{RLS}\big(e(1), e(2), \cdots, e(n)\big) = \left\|\mathbf{D}^{1/2}(n)\mathbf{d}(n) - \mathbf{D}^{1/2}(n)\mathbf{X}(n)\hat{\mathbf{w}}(n)\right\|_2^2 \tag{4.10}$$

where $\mathbf{D}^{1/2}(n)$ is a diagonal matrix with entries $D_{ii}^{1/2}(n) := \sqrt{D_{ii}(n)}$.

The canonical form of the problem typically assumes that the input-output sequences are generated by a time varying system with parameters represented by $\mathbf{w}(n)$. In most applications however, stochastic uncertainties are also present. Thus a more pragmatic data generation process is described by the noisy model

$$d(i) = \mathbf{w}^*(n)\mathbf{x}(i) + \eta(i) \tag{4.11}$$

where $\eta(i)$ is the observation noise. Note that $\mathbf{w}(n)$ reflects the true parameters which may or may not vary with time. The noise will be assumed to be i.i.d. Gaussian, i.e., $\eta(i) \sim \mathcal{N}(0, \sigma^2)$. The estimator has only access to the streaming data $x(i)$ and $d(i)$.

4.2.2 Estimation of Sparse Vectors

Let $\mathbf{x}$ be a vector in $\mathbb{C}^M$. We define the ℓ_0 quasi-norm of $\mathbf{x}$ as follows:

$$\|\mathbf{x}\|_0 = |\{x_i | x_i \neq 0\}| \tag{4.12}$$

A vector $\mathbf{x} \in \mathbb{C}^M$ is called *sparse*, if $\|\mathbf{x}\|_0 \ll M$. A wide range of interesting estimation problems deal with the estimation of sparse vectors. Many signals of interest can naturally be modeled as sparse. For example, the wireless channel usually has a few significant multi-path components. One needs to estimate such signals for various purposes.

Suppose that $\|\mathbf{w}(n)\|_0 = L \ll M$. Also, let $\mathcal{I} := \mathsf{supp}(\mathbf{w}(n))$. Given a matrix $\mathbf{A} \in \mathbb{C}^{N\times M}$ and an index set $\mathcal{J} \subseteq \{1, 2, \cdots, M\}$, we denote the sub-matrix of $\mathbf{A}$ with columns corresponding to the index set $\mathcal{J}$ by $\mathbf{A}_{\mathcal{J}}$. Similarly, we denote the sub-vector of $\mathbf{x} \in \mathbb{C}^M$ corresponding to the index set $\mathcal{J}$ by $\mathbf{x}_{\mathcal{J}}$.

A sparse approximation to $\mathbf{w}(n)$ can be obtained by solving the following optimization problem:

$$\min_{\hat{\mathbf{w}}(n)} \|\hat{\mathbf{w}}(n)\|_0 \;\; \text{s.t.} \;\; f\big(e(1), e(2), \cdots, e(n)\big) \leq \epsilon \tag{4.13}$$

where ϵ is a positive constant controlling the cost error in (4.5). The above optimization problem is, in general, computationally intractable. A considerable amount of recent research in statistical signal processing is focused on efficient estimation methods for estimating an unknown sparse vector based on noiseless/noisy observations (See, for example, [29], [30], [46], [56] and [64]). In particular, convex relaxation techniques provide a viable alternative, whereby the ℓ_0 quasi-norm in (4.13) is replaced by the convex ℓ_1 norm so that (4.13) becomes

$$\min_{\hat{\mathbf{w}}(n)} \|\hat{\mathbf{w}}(n)\|_1 \;\; \text{s.t.} \;\; f\big(e(1), e(2), \cdots, e(n)\big) \leq \epsilon \tag{4.14}$$

A convex problem results when f is convex, as in the RLS case. Note that we employ

the following definition of the ℓ_1 norm on the complex vector space $\mathbb{C}^M$:

$$\|\mathbf{w}\|_1 := \sum_{i=1}^{M} \left(|\Re\{w_i\}| \ + |\Im\{w_i\}| \right) \tag{4.15}$$

The Lagrangian formulation shows that if $f = f_{RLS}$, the optimum solution can be equivalently derived from the following optimization problem

$$\min_{\hat{\mathbf{w}}(n)} \left\{ \frac{1}{2\sigma^2} \left\| \mathbf{D}^{1/2}(n)\mathbf{d}(n) - \mathbf{D}^{1/2}(n)\mathbf{X}(n)\hat{\mathbf{w}}(n) \right\|_2^2 + \gamma \|\hat{\mathbf{w}}(n)\|_1 \right\}. \tag{4.16}$$

The parameter γ represents a trade off between estimation error and sparsity of the parameter coefficients. The optimization problem (4.16) is known in the compressed sensing literature as *basis pursuit denoising* [35] or *LASSO* in the context of linear regression [114]. It has been shown in [20, 116] that both techniques achieve performance close to the genie-aided estimator, which knows the locations of the non-zero components. Sufficient as well as necessary conditions for the existence and uniqueness of a global minimizer are derived in [116]. These conditions require that the input signal must be properly chosen so that the matrix $\mathbf{D}^{1/2}(n)\mathbf{X}(n)$ is sufficiently incoherent (we will explicitly use some of these results later on in Section 4.4.2). Suitable probing signals for exact recovery in a multi-path environment are analyzed in [16] and [63].

Since (4.16) is convex, many standard "off the shelf" optimization tools, like linear programming [35], projected gradient methods [56], and iterative thresholding [19] can be employed. Techniques based on convex optimization although quite efficient, suffer from the polynomial runtime and operate in a *batch* mode (all information are a priori required). Quite often in applications like communications, the measurements arrive sequentially and in many cases the system response is time-varying. In such cases

the batch estimators have the disadvantage of requiring greater runtime complexity and larger memory requirements. These challenges may be addressed by developing sparse adaptive filters, which have the potential of allowing online operation.

4.2.3 Low-Complexity Expectation Maximization Algorithm

The convex program in Eq. (4.16) can be solved with the conventional convex programming methods. Here, we adopt an efficient solution presented by Figueirado and Nowak [55] in the context of Wavelet-based image restoration, which we will modify to an online and adaptive setting. Consider the noisy observation model:

$$\mathbf{d}(n) = \mathbf{X}(n)\mathbf{w}(n) + \boldsymbol{\eta}(n). \tag{4.17}$$

where $\boldsymbol{\eta}(n) \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, with the following cost function

$$\begin{aligned} f_n(\mathbf{w}) &= \frac{1}{2\sigma^2}\left\|\mathbf{D}^{1/2}(n)\mathbf{d}(n) - \mathbf{D}^{1/2}(n)\mathbf{X}(n)\mathbf{w}\right\|_2^2 + \gamma\|\mathbf{w}\|_1 \\ &= \frac{1}{2\sigma^2}\Big(\mathbf{d}(n) - \mathbf{X}(n)\mathbf{w}\Big)^* \mathbf{D}(n)\Big(\mathbf{d}(n) - \mathbf{X}(n)\mathbf{w}\Big) + \gamma\|\mathbf{w}\|_1 \end{aligned} \tag{4.18}$$

If we consider the alternative observation model:

$$\mathbf{d}(n) = \mathbf{X}(n)\mathbf{w}(n) + \boldsymbol{\xi}(n). \tag{4.19}$$

with $\boldsymbol{\xi}(n) \sim \mathcal{N}(0, \sigma^2\mathbf{D}^{-1}(n))$, the convex program in Eq. (4.16) can be identified as the following penalized Maximum Likelihood (ML) problem:

$$\max_{\mathbf{w}(n)} \Big\{ \log p(\mathbf{d}(n)|\mathbf{w}(n)) - \gamma\|\mathbf{w}(n)\|_1 \Big\} \tag{4.20}$$

where $p(\mathbf{d}(n)|\mathbf{w}(n)) := \mathcal{N}(\mathbf{X}(n)\mathbf{w}(n), \sigma^2\mathbf{D}^{-1}(n))$. This ML problem is in general hard to solve. The clever idea of [55] is to decompose the noise vector $\boldsymbol{\xi}(n)$ in order

to divide the optimization problem into a denoising and a filtering problem. We adopt the same method with appropriate modifications for the cost function given in Eq. (4.20). Consider the following decomposition for $\boldsymbol{\xi}(n)$:

$$\boldsymbol{\xi}(n) = \alpha \mathbf{X}(n)\boldsymbol{\xi}_1(n) + \mathbf{D}^{-1/2}(n)\boldsymbol{\xi}_2(n) \tag{4.21}$$

where $\boldsymbol{\xi}_1(n) \sim \mathcal{N}(0, \mathbf{I})$ and $\boldsymbol{\xi}_2(n) \sim \mathcal{N}(0, \sigma^2\mathbf{I} - \alpha^2\mathbf{D}^{1/2}(n)\mathbf{X}(n)\mathbf{X}^*(n)\mathbf{D}^{1/2}(n))$. We need to choose $\alpha^2 \leq \sigma^2/s_1$, where s_1 is the largest eigenvalue of

$$\mathbf{D}^{1/2}(n)\mathbf{X}(n)\mathbf{X}^*(n)\mathbf{D}^{1/2}(n),$$

in order for $\boldsymbol{\xi}_2(n)$ to have a positive semi-definite covariance matrix (we will talk about how to choose the parameter α in practice in Section 4.4.5). We can therefore rewrite the model in Eq. (4.19) as

$$\begin{cases} \mathbf{v}(n) = \mathbf{w}(n) + \alpha\boldsymbol{\xi}_1(n) \\ \mathbf{d}(n) = \mathbf{X}(n)\mathbf{v}(n) + \mathbf{D}^{-1/2}(n)\boldsymbol{\xi}_2(n) \end{cases} \tag{4.22}$$

The Expectation Maximization (EM) algorithm can be used to solve the penalized ML problem of (4.20), with the help of the following alternative penalized ML problem

$$\max_{\mathbf{w}(n)} \Big\{ \log p(\mathbf{d}(n), \mathbf{v}(n)|\mathbf{w}(n)) - \gamma\|\mathbf{w}(n)\|_1 \Big\}, \tag{4.23}$$

which is easier to solve, employing $\mathbf{v}(n)$ as the auxiliary variable. The ℓth iteration of the EM algorithm is as follows:

$$\begin{cases} \text{E-step: } Q(\mathbf{w}|\hat{\mathbf{w}}(n)) := -\frac{1}{2\alpha^2}\|\mathbf{r}^{(\ell)} - \mathbf{w}\|_2^2 - \gamma\|\mathbf{w}\|_1, \\ \qquad \text{where } \mathbf{r}^{(\ell)}(n) := \big(\mathbf{I} - \frac{\alpha^2}{\sigma^2}\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{X}(n)\big)\hat{\mathbf{w}}^{(\ell)}(n) + \frac{\alpha^2}{\sigma^2}\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{d}(n) \\ \\ \text{M-step: } \hat{\mathbf{w}}^{(\ell+1)}(n) := \arg\max_{\mathbf{w}} Q(\mathbf{w}|\hat{\mathbf{w}}(n)) = \mathcal{S}(\mathbf{r}^{(\ell)}) \end{cases} \tag{4.24}$$

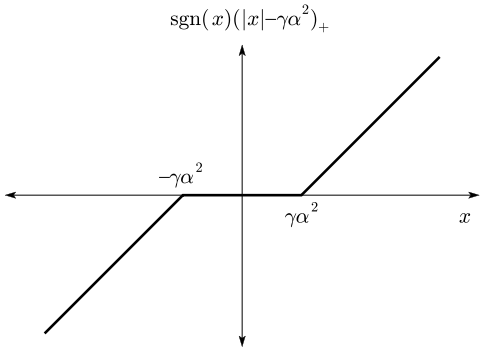


Figure 4.1: Soft thresholding function

where $\mathcal{S}(\cdot) : \mathbb{C}^M \mapsto \mathbb{C}^M$ is the element-wise *soft thresholding* function defined as

$$\big(\mathcal{S}(\mathbf{w})\big)_i := \operatorname{sgn}\big(\Re\{w_i\}\big)\big(|\Re\{w_i\}| - \gamma\alpha^2\big)_+ + i\operatorname{sgn}\big(\Im\{w_i\}\big)\big(|\Im\{w_i\}| - \gamma\alpha^2\big)_+ \quad (4.25)$$

for all $i = 1, 2, \cdots, M$ and the $(\cdot)_+$ operator is defined as $(x)_+ := \max(x, 0)$.

Note that the above algorithm belongs to a class of pursuit algorithms denoted by iterated shrinkage algorithms (See [23] for a detailed discussion). It is known that the EM algorithm converges to a local maximum (See for example, [45], [91] and [115]). Moreover, under the hypothesis of $\mathbf{X}_{\mathcal{I}}(n)$ being maximal rank, the maximizer is unique and therefore the EM algorithm converges to the unique maximizer of the cost function [116]. The latter hypothesis can be satisfied by appropriately designing the input sequence $x(n)$. For example, a Gaussian i.i.d. input sequence $x(n)$ (as well as the designs given in [16] and [63]) will guarantee this property with high probability.

The soft thresholding function is plotted in Figure 4.1. Note that the soft thresholding function tends to decrease the support of the estimate $\hat{\mathbf{w}}(n)$, since it shrinks the support to those elements whose absolute value is greater than $\gamma\alpha^2$. We can use this observation to express the double iteration given in Eq. (4.24) in a low com-

plexity fashion. Note that the M-step applies soft thresholding independently on the real and imaginary parts of the vector $\mathbf{r}^{(\ell)}(n)$. In order to simplify the notation in what follows, we present the low complexity implementation of the EM algorithm for $\mathbf{r}^{(\ell)}(n) \in \mathbb{R}^M$. Generalization to $\mathbf{r}^{(\ell)}(n) \in \mathbb{C}^M$ is straightforward, since the low complexity implementation can be applied to the real and imaginary parts of $\mathbf{r}^{(\ell)}(n)$ independently.

Let $\mathcal{I}^{(\ell)}$ be the support of $\mathbf{r}^{(\ell)}(n)$ at the ℓth iteration. Let

$$\begin{cases} \mathcal{I}_+^{(\ell)} := \{i : r_i^{(\ell)}(n) > \gamma\alpha^2\} \subseteq \mathcal{I}^{(\ell)} \\ \mathcal{I}_-^{(\ell)} := \{i : r_i^{(\ell)}(n) < -\gamma\alpha^2\} \subseteq \mathcal{I}^{(\ell)} \end{cases}, \tag{4.26}$$

$$\mathbf{B}(n) := \mathbf{I} - \frac{\alpha^2}{\sigma^2}\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{X}(n), \tag{4.27}$$

and

$$\mathbf{u}(n) := \frac{\alpha^2}{\sigma^2}\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{d}(n). \tag{4.28}$$

Note that the second iteration in Eq. (4.24) can be written as

$$\hat{w}_i^{(\ell+1)}(n) = \begin{cases} r_i^{(\ell)}(n) - \gamma\alpha^2 & i \in \mathcal{I}_+^{(\ell)} \\ r_i^{(\ell)}(n) + \gamma\alpha^2 & i \in \mathcal{I}_-^{(\ell)} \\ 0 & i \notin \mathcal{I}_+^{(\ell)} \cup \mathcal{I}_-^{(\ell)} \end{cases} \tag{4.29}$$

for $i = 1, 2, \cdots, M$. We then have

$$\mathbf{B}(n)\hat{\mathbf{w}}^{(\ell+1)}(n) = \mathbf{B}_{\mathcal{I}_+^{(\ell)}}(n)\big(\mathbf{r}^{(\ell)}_{\mathcal{I}_+^{(\ell)}}(n) - \gamma\alpha^2\mathbf{1}_{\mathcal{I}_+^{(\ell)}}\big) + \mathbf{B}_{\mathcal{I}_-^{(\ell)}}(n)\big(\mathbf{r}^{(\ell)}_{\mathcal{I}_-^{(\ell)}}(n) + \gamma\alpha^2\mathbf{1}_{\mathcal{I}_-^{(\ell)}}\big) \tag{4.30}$$

which allows us to express the EM iteration as follows:

$$\begin{cases} \mathbf{r}^{(\ell+1)}(n) = \mathbf{B}_{\mathcal{I}_+^{(\ell)}}(n)\big(\mathbf{r}^{(\ell)}_{\mathcal{I}_+^{(\ell)}}(n) - \gamma\alpha^2\mathbf{1}_{\mathcal{I}_+^{(\ell)}}\big) + \mathbf{B}_{\mathcal{I}_-^{(\ell)}}(n)\big(\mathbf{r}^{(\ell)}_{\mathcal{I}_-^{(\ell)}}(n) + \gamma\alpha^2\mathbf{1}_{\mathcal{I}_-^{(\ell)}}\big) + \mathbf{u}(n) \\ \mathcal{I}_+^{(\ell+1)} = \{i : r_i^{(\ell+1)}(n) > \gamma\alpha^2\} \\ \mathcal{I}_-^{(\ell+1)} = \{i : r_i^{(\ell+1)}(n) < -\gamma\alpha^2\} \end{cases} \tag{4.31}$$

This new set of iteration has a lower computational complexity, since it restricts the matrix multiplications to the instantaneous support of the estimate $\mathbf{r}^{(\ell)}(n)$, which is expected to be close to the support of $\mathbf{w}(n)$ [116]. We denote the iterations given in Eq. (4.31) by Low-Complexity Expectation Maximization (LCEM) algorithm.

4.3 The SPARLS Algorithm

4.3.1 The Main Algorithm

Upon the arrival of the nth input, $\mathbf{B}(n)$ and $\mathbf{u}(n)$ can be obtained via the following rank-one update rules:

$$\begin{cases} \mathbf{B}(n) = \lambda \mathbf{B}(n-1) - \frac{\alpha^2}{\sigma^2}\mathbf{x}(n)\mathbf{x}^*(n) + (1-\lambda)\mathbf{I} \\ \mathbf{u}(n) = \lambda \mathbf{u}(n-1) + \frac{\alpha^2}{\sigma^2} d^*(n)\mathbf{x}(n) \end{cases} \tag{4.32}$$

Upon the arrival of the nth input, $x(n)$, the LCEM algorithm computes the estimate $\hat{\mathbf{w}}(n)$ given $\mathbf{B}(n)$, $\mathbf{u}(n)$ and $\mathbf{s}^{(0)}(n)$. The LCEM algorithm is summarized in Algorithm 2. Note that the input argument K denotes the number of EM iterations.

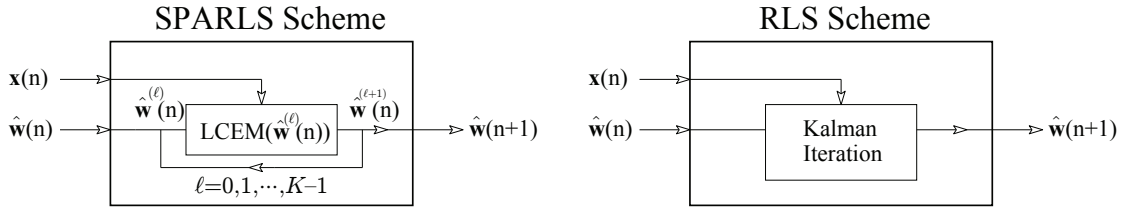


Figure 4.2: Schematic realizations of SPARLS and RLS algorithms.

The SPARLS algorithm is formally defined in Algorithm 3. Without loss of generality, we can set the time index $n = 1$ such that $x(1) \neq 0$, in order for the initialization

Algorithm 2 LCEM $\left(\mathbf{B}, \mathbf{u}, \hat{\mathbf{w}}, \mathcal{I}_+^{(K-1)} \cup \mathcal{I}_-^{(K-1)}, K\right)$

Inputs: $\mathbf{B}$, $\mathbf{u}$, $\hat{\mathbf{w}}$, $\mathcal{I}_+^{(K-1)} \cup \mathcal{I}_-^{(K-1)}$, and K.

Outputs: $\hat{\mathbf{w}}$, $\mathcal{I}_+^{(K-1)}$ and $\mathcal{I}_-^{(K-1)}$.

$\mathbf{r}^{(0)} = \mathbf{B}_{\mathcal{I}_+^{(K-1)}} \hat{\mathbf{w}}_{\mathcal{I}_+^{(K-1)}} + \mathbf{B}_{\mathcal{I}_-^{(K-1)}} \hat{\mathbf{w}}_{\mathcal{I}_-^{(K-1)}} + \mathbf{u}$.

$\mathcal{I}_+^{(0)} = \{i : r_i^{(0)} > \gamma\alpha^2\}$.

$\mathcal{I}_-^{(0)} = \{i : r_i^{(0)} < -\gamma\alpha^2\}$.

for $\ell = 1, 2, \cdots, K-1$ **do**

  $\mathbf{r}^{(\ell)} = \mathbf{B}_{\mathcal{I}_+^{(\ell-1)}} \left(\mathbf{r}^{(\ell-1)}_{\mathcal{I}_+^{(\ell-1)}} - \gamma\alpha^2 \mathbf{1}_{\mathcal{I}_+^{(\ell-1)}}\right) + \mathbf{B}_{\mathcal{I}_-^{(\ell-1)}} \left(\mathbf{r}^{(\ell-1)}_{\mathcal{I}_-^{(\ell-1)}} + \gamma\alpha^2 \mathbf{1}_{\mathcal{I}_-^{(\ell-1)}}\right) + \mathbf{u}$.

  $\mathcal{I}_+^{(\ell)} = \{i : r_i^{(\ell)} > \gamma\alpha^2\}$.

  $\mathcal{I}_-^{(\ell)} = \{i : r_i^{(\ell)} < -\gamma\alpha^2\}$.

end for

for $i = 1, 2, \cdots, M$ **do**

$$\hat{w}_i = \begin{cases} r_i^{(K-1)} - \gamma\alpha^2 & i \in \mathcal{I}_+^{(K-1)} \\ r_i^{(K-1)} + \gamma\alpha^2 & i \in \mathcal{I}_-^{(K-1)} \\ 0 & i \notin \mathcal{I}_+^{(K-1)} \cup \mathcal{I}_-^{(K-1)} \end{cases}.$$

end for

to be well-defined. The schematic realizations of the SPARLS and RLS algorithm are depicted in Figure 4.2. Both algorithms perform in an online fashion and update the estimate $\hat{\mathbf{w}}(n)$ upon the arrival of the new data input $x(n)$.

4.3.2 Low Complexity Update Scheme

The update equation for $\mathbf{B}(n)$ can be implemented in a low complexity fashion. This is due to the fact that the LCEM algorithm only needs the columns of $\mathbf{B}(n)$

Algorithm 3 SPARLS

Inputs: $\mathbf{B}(1) = \mathbf{I} - \frac{\alpha^2}{\sigma^2}\mathbf{x}(1)\mathbf{x}^*(1)$, $\mathbf{u}(1) = \frac{\alpha^2}{\sigma^2}\mathbf{x}(1)d^*(1)$ and K.

Output: $\hat{\mathbf{w}}(n)$.

1: **for all** Input $x(n)$ **do**

2: $\mathbf{B}(n) = \lambda\mathbf{B}(n-1) - \frac{\alpha^2}{\sigma^2}\mathbf{x}(n)\mathbf{x}^*(n) + (1-\lambda)\mathbf{I}$.

3: $\mathbf{u}(n) = \lambda\mathbf{u}(n-1) + \frac{\alpha^2}{\sigma^2}d^*(n)\mathbf{x}(n)$.

4: Run LCEM $\big(\mathbf{B}(n), \mathbf{u}(n), \hat{\mathbf{w}}(n-1), \mathcal{I}_+^{(K-1)}(n-1) \cup \mathcal{I}_-^{(K-1)}(n-1), K\big)$.

5: Update $\hat{\mathbf{w}}(n)$.

6: **end for**

corresponding to the index set $\mathcal{I}_+ \cup \mathcal{I}_-$. Thus, given the hypothesis that the subset $\mathcal{I}^{(0)}(n)$ does not vary much with n, *i.e.*, $|\mathcal{I}^{(0)}(n) \backslash \mathcal{I}^{(0)}(n-1)| \ll |\mathcal{I}^{(0)}(n)|$, one can implement the update step for $\mathbf{B}(n)$ in a low complexity fashion as follows.

First, we consider the updating procedure for $\mathbf{B}(n)$ when the new input data $x(n)$ has arrived. Clearly, $\mathcal{I}^{(0)}(n) = \mathcal{I}_+^{(K-1)}(n-1) \cup \mathcal{I}_-^{(K-1)}(n-1)$, if we run the LCEM algorithm a total of K times for each new input $x(n)$. The columns of $\mathbf{B}(n)$ required for the LCEM algorithm clearly correspond to $\mathcal{I}^{(0)}(n)$. We also assign a variable $t_i \in \{1, 2, \cdots, n-1\}$ to each column of $\mathbf{B}(n)$, which denotes the last time index when the ith column of the matrix $\mathbf{B}$ was in the index set $\mathcal{I}^{(0)}$. Upon the arrival of $x(n)$, we only update the columns of $\mathbf{B}(n)$ corresponding to the index set $\mathcal{I}^{(0)}(n)$ and denote the resulting matrix by $\tilde{\mathbf{B}}(n)$:

$$\tilde{\mathbf{B}}_i(n) = \lambda^{n-t_i}\tilde{\mathbf{B}}_i(n-1) - \frac{\alpha^2}{\sigma^2}\sum_{m=0}^{n-t_i-1}\lambda^m\Big((\mathbf{x}(n-m)\mathbf{x}^*(n-m))_i + (1-\lambda)\mathbf{I}_i\Big) \quad (4.33)$$

for all $i \in \mathcal{I}^{(0)}(n)$. For example, if the ith column of $\tilde{\mathbf{B}}(n)$ has been last updated at

time $n-3$, then $t_i = n-3$, hence the update equation simply becomes:

$$\begin{aligned}\tilde{\mathbf{B}}_i(n) &= \lambda^3\tilde{\mathbf{B}}_i(n-1) - \frac{\alpha^2}{\sigma^2}\Big(\mathbf{x}(n)\mathbf{x}^*(n) + \lambda\mathbf{x}(n-1)\mathbf{x}^*(n-1) + \lambda^2\mathbf{x}(n-2)\mathbf{x}^*(n-2)\Big)_i \\ &\quad + (1-\lambda)(1+\lambda+\lambda^2)\mathbf{I}_i\end{aligned}$$

Algorithm 4 LCU($\tilde{\mathbf{B}}(n-1), \mathcal{J}, \{t_i\}_{i=1}^M$)

Inputs: $\tilde{\mathbf{B}}(n-1)$, $\mathcal{J}$ and $\{t_i\}_{i=1}^M$.

Output: $\mathbf{B}_{\mathcal{J}}$ and $\{t_i\}_{i=1}^M$.

1: **for all** i in $\mathcal{J}$ **do**

2: $\quad \tilde{\mathbf{B}}_i(n) = \lambda\Big\{\tilde{\mathbf{B}}_i(n-1) - \frac{\alpha^2}{\sigma^2}\sum_{m=0}^{n-t_i-1}\lambda^m\Big((\mathbf{x}(n-m)\mathbf{x}^*(n-m))_i + (1-\lambda)\mathbf{I}_i\Big)\Big\}$.

3: $\quad t_i \leftarrow n$.

4: **end for**

5: $\mathbf{B}_{\mathcal{J}} \leftarrow \tilde{\mathbf{B}}_{\mathcal{J}}$

Subsequently, the time indices t_i will be updated as $t_i = n$ for all $i \in \mathcal{I}^{(0)}(n)$ and remain unchanged otherwise. We can formally define the sub-routine Low Complexity Update (LCU) for updating $\mathbf{B}(n)$ as in Algorithm 4. Note that if $\mathcal{I}^{(0)}(n) = \{1, 2, \cdots, M\}$ for all times, then the above update equation for $\tilde{\mathbf{B}}(n)$ is equivalent to the update equation in Eq. (4.32). But, due to the sparsifying nature of the estimator, the index set $\mathcal{I}^{(0)}(n)$ is expected to be very close to the true index set $\mathcal{I}$. In that case the number of column updates at each time is $\mathcal{I}^{(0)}(n)$. Moreover, these updates are usually very simple in the steady state, since most of the t_is are equal to n, for all $i \in \mathcal{I}^{(0)}(n)$, with the hypothesis of $|\mathcal{I}^{(0)}(n) \backslash \mathcal{I}^{(0)}(n-1)| \ll |\mathcal{I}^{(0)}(n)|$. This way, we can exploit the sparseness of the estimate in order to reduce the complexity of the update process for $\mathbf{B}(n)$. Therefore, one can use the LCU subroutine

$\text{LCU}(\mathbf{B}(n-1), \mathcal{I}_+^{(K-1)}(n-1) \cup \mathcal{I}_-^{(K-1)}(n-1), \{t_i\}_{i=1}^M)$ on line 2 of the SPARLS algorithm. Similarly, the LCU subroutine can be used in the LCEM algorithm (right before lines 1 and 5), when the algorithm needs to access sub-matrices such as $\mathbf{B}_{\mathcal{I}_+^{(\ell)}}(n)$ or $\mathbf{B}_{\mathcal{I}_-^{(\ell)}}(n)$. Nevertheless, the hypothesis of $|\mathcal{I}^{(0)}(n) \backslash \mathcal{I}^{(0)}(n-1)| \ll |\mathcal{I}^{(0)}(n)|$ may be violated, in which case using the LCU subroutine might result in drawbacks (See Section IV-D for a detailed discussion). Nevertheless, one can always resort to the original form of the SPARLS algorithm.

4.4 Analysis of the SPARLS Algorithm

In this section, we will study the convergence of SPARLS to a fixed point in a stationary environment in Section 4.4.1, the steady state error of the SPARLS in Section 4.4.2, comparison of the error performance of SPARLS and RLS in a stationary environment for finite sample size, i.e., $n < \infty$ in Section 4.4.3, the complexity and storage issues of SPARLS (with and without the LCU subroutine) in Section 4.4.4, and finally, adjusting the parameters of SPARLS in Section 4.4.5.

4.4.1 Convergence Analysis

In order to study the convergence of the SPARLS algorithm, we need to make a number of additional assumptions. First of all, we consider the case of constant unknown vector $\mathbf{w}(n)$, *i.e.*, $\mathbf{w}(n) = \mathbf{w}_0$ for all $n = 1, 2, \cdots$. Moreover, we analyze the convergence in a stationary environment: the input sequence $\{x(n)\}_{n=1}^{\infty}$ and the output sequence $\{d(n)\}_{n=1}^{\infty}$ are realizations of a jointly stationary random process.

Before moving on to the convergence analysis of SPARLS, we briefly overview

the convergence properties of the EM algorithm. The global and componentwise convergence of the EM algorithm has been widely studied in the statistics literature (See, for example, [45] and [91]). According to the original paper of Dempster et al. [45], the EM algorithm can be represented by a mapping $\mathcal{M}_n : \mathbb{C}^M \longmapsto \mathbb{C}^M$, defined as

$$\hat{\mathbf{w}}^{(\ell+1)}(n) = \mathcal{M}_n(\hat{\mathbf{w}}^{(\ell)}(n)) \tag{4.34}$$

where the mapping $\mathcal{M}_n$ is the composition of the E and M steps at time n. Moreover, if the minimizer of the objective function

$$f_n(\mathbf{w}) := \frac{1}{2\sigma^2}\|\mathbf{D}^{1/2}\mathbf{d}(n) - \mathbf{D}^{1/2}\mathbf{X}(n)\mathbf{w}\|_2^2 + \gamma\|\mathbf{w}\|_1 \tag{4.35}$$

is unique, we have

$$f_n(\mathbf{w}^{(\ell+1)}(n)) < f_n(\mathbf{w}^{(\ell)}(n)). \tag{4.36}$$

From Lemma 3 of Tropp [116], we know that the minimizer of the objective function given in Eq. (4.35) is unique if $\mathbf{X}_{\mathcal{I}}(n)$ is maximal rank, where $\mathcal{I} = \mathsf{supp}(\mathbf{w}_0)$. We denote this minimizer by $\tilde{\mathbf{w}}(n)$. The hypothesis of $\mathbf{X}_{\mathcal{I}}(n)$ being maximal rank can be achieved if the input sequence is persistently exciting (In other words, the input must be sufficiently rich to properly excite all modes of the system). For example, if the input sequence $x(n)$ is drawn from an i.i.d. random process, the columns of $\mathbf{X}_{\mathcal{I}}(n)$ form an orthogonal set with probability 1. Hence, we can assume throughout the analysis that the minimizer of the objective function is unique.

The SPARLS algorithm only performs the EM algorithm a finite (K) number of times for each n. Hence, it does not exactly solve the minimization problem in (4.16). Furthermore, the cost function varies at each step (with n). Hence, it is not trivial

that performing the EM algorithm a finite number of times ($K < \infty$) at each step, results in convergence to the unique minimizer of $f_n(\mathbf{x})$, as $n \to \infty$. Indeed, the following theorem establishes the convergence of the SPARLS algorithm under the above assumptions:

Theorem 4.4.1 (Convergence)**.** *Given a stationary environment and a constant target sparse vector* $\mathbf{w}_0$*, the SPARLS algorithm (with* $K < \infty$*) converges almost surely to the unique minimizer of the cost function* $f_n(\mathbf{w})$*, as* $n \to \infty$*.*

Idea of proof: The idea of proof is to relate the convergence behavior of the EM algorithm along one specific function $f_n(\mathbf{w})$ to the convergence of the SPARLS algorithm across different functions $f_n(\mathbf{w})$. The proof is formally given in section 7.3.

Note that the case of $n \to \infty$ is not of particular interest in our analysis of the stationary scenario, since it defeats the purpose of compressive sampling. However, the convergence proof guarantees that we can get to an arbitrarily small neighborhood of the fixed point (i.e., limit of the unique minimizer of $f_n(\mathbf{w})$) for finite n. This fact will be used later in the performance comparison of SPARLS and RLS (See Theorem 4.2). Next, we study the steady state error of the SPARLS algorithm.

4.4.2 Steady State Error Analysis

We define the average instantaneous error of the SPARLS algorithm as follows:

$$\epsilon(n) := \mathbb{E}_\eta\Big\{ \|\hat{\mathbf{w}}(n) - \mathbf{w}(n)\|_2 \Big\}. \tag{4.37}$$

As it is shown in section 4.7, $\epsilon(n)$ obeys the following recurrence relation:

$$\begin{aligned}\epsilon(n+1) \leq \rho(n)^K \epsilon(n) + \mathbb{E}_\eta\Big\{\big\|(\mathbf{D}^{1/2}(n)\mathbf{X}_{\mathcal{I}}(n))^+ \boldsymbol{\eta}_{\mathcal{I}}(n)\big\|_2\Big\} \\ + \gamma\sigma^2 \Big\|\Big(\mathbf{X}^*_{\mathcal{I}}(n)\mathbf{D}(n)\mathbf{X}_{\mathcal{I}}(n)\Big)^{-1}\Big\|_{2,\infty} \\ + \|\mathbf{w}(n+1) - \mathbf{w}(n)\|_2. \end{aligned} \tag{4.38}$$

where $\mathbf{A}^+$ is the Moore-Penrose pseudo-inverse of matrix $\mathbf{A}$ and $\rho(n)$ is defined as $\rho(n) := 1 - \frac{\alpha^2}{\sigma^2} s_M(n)$, with $s_M(n)$ being the minimum eigenvalue of $\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{X}(n)$ and the $(2,\infty)$-norm of a matrix $\mathbf{A}$ is defined as $\|\mathbf{A}\|_{2,\infty} := \max_{\mathbf{x}:\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_\infty}$.

The first term on the right hand side corresponds to the linear convergence of the EM algorithm, the second term corresponds to the observation noise, the third term corresponds to the error bias with respect to the genie-aided (oracle) estimate, and the fourth term corresponds to the evolution of the true vector $\mathbf{w}(n)$. Note that we are allowing the target $\mathbf{w}(n)$ to change with time in the steady state. A popular model to describe the evolution of the parameter vector in statistical signal processing is the random walk model of the form:

$$\mathbf{w}(n+1) = \mathbf{w}(n) \ + \kappa\boldsymbol{\delta}(n) \tag{4.39}$$

where $\boldsymbol{\delta}(n)$ is a white Gaussian random vector with covariance matrix $\Delta(n)$ and κ is a scaling constant (See, for example, [85]). The scaling constant κ represents the speed of the time evolution of $\mathbf{w}(n)$. In order for the error recurrence relation to remain valid, we need to assume $\kappa \ll 1$, so that the estimate $\hat{\mathbf{w}}(n)$ remains in a small neighborhood of the target $\tilde{\mathbf{w}}(n)$.

If we further assume that the last three terms on the right hand side do not change rapidly with n, using the Cauchy-Schwarz inequality and averaging over $\boldsymbol{\delta}(n)$

(assuming independence between $\boldsymbol{\delta}(n)$ and $\boldsymbol{\eta}(n)$), we get:

$$\epsilon(n) \lessapprox \frac{1}{1-\rho(n)^K}\left(\frac{\sigma\sqrt{\mathrm{Tr}\left((\mathbf{X}_{\mathcal{I}}^*(n)\mathbf{D}(n)\mathbf{X}_{\mathcal{I}}(n))^{-1}\right)}+\gamma\alpha^2}{s_{\min}\big(\mathbf{X}_{\mathcal{I}}^*(n)\mathbf{D}(n)\mathbf{X}_{\mathcal{I}}(n)\big)}+\kappa\sqrt{\mathrm{Tr}(\Delta(n))}\right) \quad (4.40)$$

where $s_{\min}(\mathbf{A})$ denotes the minimum eigenvalue of the matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$. The first term on the right hand side demonstrates the trade-off between the denoising of the estimate and the additional cost due to ℓ_1-regularization. The second term corresponds to the regeneration of the unknown vector $\mathbf{w}(n)$. Finally, the factor of $1/(1-\rho(n)^K)$ in the error bound is due to the linear convergence of the EM algorithm.

4.4.3 Error Performance Comparison of SPARLS and RLS

In the time-invariant scenario, choosing $\lambda < 1$, will result in a persistent steady state MSE error as $n \to \infty$, unlike RLS which converges to the true vector as the number of measurements tend to infinity (with $\lambda = 1$). However, the steady state MSE error of SPARLS can be sufficiently reduced by choose λ close enough to 1 in the low sparsity regime. In fact, in the following theorem, we show that for L/M small enough and for large enough but *finite* number of measurements$n$, $\lambda < 1$ sufficiently close to 1, and an appropriate choice of γ, the MSE performance of SPARLS is superior to that of RLS (with $\lambda = 1$). This is indeed in line with the premises of compressive sampling, which guarantee superior performance with significantly lower number of measurements:

Theorem 4.4.2. *Consider a stationary environment, for which the RLS algorithm operates with $\lambda = 1$ and recovers the true tap-weight vector $\mathbf{w}_0$ as $n \to \infty$. Let $\epsilon(n)$ and $\epsilon_{RLS}(n)$ denote the average instantaneous errors of the SPARLS and RLS*

algorithms at the nth iteration, respectively. Then, for a given n_0 large enough, there exist constants $0 < a < 1$, $\lambda_0 \in (0,1)$ sufficiently close to 1 and γ_0 such that for $\lambda = \lambda_0$ and $\gamma = \gamma_0$ we have

$$\epsilon(n_0) < \epsilon_{RLS}(n_0), \tag{4.41}$$

for $L/M < a$.

Idea of proof: The proof uses basic ideas regarding the Basis Pursuit algorithms in compressed sensing (See, for example, [116] and [20]) and is given in section 4.8.

In fact, the MSE of SPARLS can be significantly lower than that of RLS for finite n in the low sparsity regime, i.e., $L \ll M$. This is evident in the fact that only the components of noise corresponding to the index set $\mathcal{I}$ appear in the error expression of SPARLS in Eq. (4.38), whereas all the noise coordinates contribute to the MSE of RLS. This can also be observed from Fig. 5. Here, we have $L = 5$ and $M = 100$. For $n_0 \approx 120$, SPARLS achieves its steady state error level, while it takes a much longer time for RLS to achieve the same MSE (in about 500 iterations). Finally, as simulation studies reveal, the SPARLS algorithm has significant MSE advantages over the RLS algorithm, especially in low SNR and low sparsity regimes.

4.4.4 Complexity and Storage Issues

The SPARLS algorithm has a computational complexity of $\mathcal{O}(M^2)$ multiplications per step, which coincides with the order of complexity of the RLS algorithm [65]. In what follows, we motivate the use of the LCU subroutine and its role in potentially decreasing the computational complexity of the SPARLS algorithm under the hypothesis that the index set $\mathcal{I}^{(0)}(n)$ does not vary much across different n in the steady

state, *i.e.*, $|\mathcal{I}^{(0)}(n)\backslash\mathcal{I}^{(0)}(n-1)| \ll |\mathcal{I}^{(0)}(n)|$.

If the LCU sub-routine is used in lines 1 and 5 of the LCEM algorithm, it will be called a total of K times for each new input $x(n)$ and requires a total of $\sum_{\ell=0}^{K-1}\big(|\mathcal{I}_+^{(\ell)}(n)| + |\mathcal{I}_-^{(\ell)}(n)|\big)$ column updates overall. Recall that t_i denotes the last time index when the ith column of the matrix $\tilde{\mathbf{B}}(n)$ appeared in the index set $\mathcal{I}^{(0)}(t_i)$. For each $i \in \mathcal{I}_+^{(\ell)}(n) \cup \mathcal{I}_-^{(\ell)}(n)$, the ith column of $\tilde{\mathbf{B}}(n)$ requires a total of $M(n-t_i)+2$ multiplications. Hence, the total number of multiplications required for K runs of the LCU sub-routine is given by $\sum_{\ell=0}^{K-1}\sum_{i\in\mathcal{I}_+^{(\ell)}(n)\cup\mathcal{I}_-^{(\ell)}(n)}(M(n-t_i)+2)$. The hypothesis of $|\mathcal{I}^{(0)}(n)\backslash\mathcal{I}^{(0)}(n-1)| \ll |\mathcal{I}^{(0)}(n)|$, implies that the indices t_i are very close to n. In other words, $n - t_i \approx \mathcal{O}(1)$, for all $t_i \in \mathcal{I}^{(0)}(n)$. Therefore, the total number of multiplications will be $\mathcal{O}(KMN)$, where $N := \frac{1}{K}\sum_{\ell=0}^{K-1}\big(|\mathcal{I}_+^{(\ell)}(n)| + |\mathcal{I}_-^{(\ell)}(n)|\big)$.

Moreover, the LCEM algorithm requires $M\big(|\mathcal{I}_+^{(\ell)}(n)| + |\mathcal{I}_-^{(\ell)}(n)|\big)$ multiplications at the ℓth iteration in order to perform the E step. Thus, for a total of K iterations, the number of multiplications carried out by the LCEM algorithm will be KMN. For a sparse signal $\mathbf{w}(n)$, one expects to have $N \approx \mathcal{O}(\|\mathbf{w}(n)\|_0) = \mathcal{O}(L)$. Therefore, the overall complexity of the LCEM algorithm is roughly of the order $\mathcal{O}(KLM)$. Thus under the hypothesis of $|\mathcal{I}^{(0)}(n)\backslash\mathcal{I}^{(0)}(n-1)| \ll |\mathcal{I}^{(0)}(n)|$, the SPARLS algorithm has a lower computational complexity than the RLS algorithm, which requires $\mathcal{O}(M^2)$ multiplications for each step.

Note that the assumption of $|\mathcal{I}^{(0)}(n)\backslash\mathcal{I}^{(0)}(n-1)| \ll |\mathcal{I}^{(0)}(n)|$ may be violated at some steps of the algorithm. This can, for example, happen when the support of the true vector changes over time. However, even when the support of the true vector is constant over time, a new component, say i, may arise in $\mathcal{I}^{(0)}(n)$ after a

long time ($t_i \ll n$). Therefore, the LCU routine needs to update the corresponding column of $\tilde{\mathbf{B}}(n)$ using all the previous regressors from time t_i to n. Moreover, the LCU subroutine requires storing all the regressors $x(j)$ from time $j = \min_i t_i$ to n. However, simulation studies reveal that such events are very rare (a component being inactive for a long time which suddenly arises in $\mathcal{I}^{(0)}(n)$). Although this is a drawback compared to RLS (in terms of storage requirements), the cost of storing a finite number of regressors is traded off with potential computational complexity reduction. Finally, note that in any case the cumulative computational complexity of SPARLS using the LCU subroutine (from time 1 to n) will always be lower or equal to that of RLS.

4.4.5 Adjusting the Parameters of SPARLS

Parameter α: As mentioned earlier in Section 4.2.3, the parameter α in the SPARLS algorithm must be chosen such that $\alpha^2 \leq \sigma^2/s_1$, where s_1 is the largest eigenvalue of $\mathbf{D}^{1/2}(n)\mathbf{X}(n)\mathbf{X}^*(n)\mathbf{D}^{1/2}$. This constraint clearly depends on the underlying statistical characteristics of the input sequence $x(n)$. Here, we investigate this constraint for the Gaussian i.i.d. input sequence, *i.e.*, $x(i) \sim \mathcal{N}(0, \nu^2)$, for $i = 1, 2, \cdots, n$, for simplicity. Generalization to other stationary input sequences is possible.

First, note that the maximum eigenvalue of the above matrix is equal to the maximum eigenvalue of $\mathbf{C}(n) := \mathbf{X}^*(n)\mathbf{D}(n)\mathbf{X}(n)$. Recall that the rows of the matrix $\mathbf{X}(n)$ are the tap inputs at times $1, 2, \cdots, n$. Hence, $\mathbf{X}(n)$ has a Toeplitz structure [63]. We have

$$\mathbf{C}(n) = \sum_{k=1}^{n} \lambda^{n-k} \mathbf{x}(k)\mathbf{x}^*(k) \tag{4.42}$$

where $\mathbf{x}(k)$ is the tap input at time k. Hence, the (i,j)th element of the $\mathbf{C}(n)$ can be expressed as $C_{ij}(n) = \sum_{k=1}^{n} \lambda^{n-k} x_i(k) x_j^*(k)$. In order to obtain bounds on the eigen-values of the matrix $\mathbf{C}(n)$, we adopt the methodology of [63] which uses the Gershgorin's disc theorem together with concentration bounds on sums of independent random variables. In our case, however, we are dealing with weighted sums of random variables. We first state a lemma from [84] which we will use later:

Lemma 4.4.3. *Let $x_1, x_2, \cdots, x_n$ be i.i.d. Gaussian variables with mean zero and variance 1. Let $a_1, a_2, \cdots, a_n$ be nonnegative. Let*

$$|a|_\infty := \sup_i a_i, \quad |a|_2^2 = \sum_{i=1}^{n} a_i^2$$

and

$$Z := \sum_{i=1}^{n} a_i (x_i^2 - 1)$$

Then, the following inequalities hold for all t:

$$\mathbb{P}\big(Z \geq 2|a|_2\sqrt{t} + 2|a|_\infty t\big) \leq \exp(-t)$$

$$\mathbb{P}\big(Z \leq -2|a|_2\sqrt{t}\big) \leq \exp(-t)$$

Proof. The proof is given in [84]. □

Now, the ith diagonal element of $\mathbf{C}(n)$ is given by

$$C_{ii}(n) = \sum_{k=1}^{n} \lambda^{n-k} x_i^2(k)$$

with $\mathbb{E}(C_{ii}(n)) = n_\lambda \nu^2$, where $n_\lambda := \frac{1-\lambda^{n+1}}{1-\lambda}$. Using lemma 4.4.3 with $a_i := \lambda^{n-i}$, $i = 1, 2, \cdots, n$ yields:

$$\mathbb{P}\big(|C_{ii}(n) - n_\lambda \nu^2| \geq 4\nu^2\sqrt{n_{\lambda^2} t}\big) \leq 2\exp(-t)$$

for $0 \leq t \leq 1$. Also, a slight modification of Lemma 6 in [63] yields:

$$\mathbb{P}\big(|C_{ij}(n)| \geq t\big) \leq 2\exp\bigg(-\frac{t^2}{4\nu^2(n_{\lambda^2}\nu^2 + t/2)}\bigg)$$

Similar to [63], we seek conditions on λ, n and ν^2 such that the eigenvalues of $\mathbf{C}(n)$ lie in the interval $[1-\delta, 1+\delta]$, where $\delta < 1$ is a positive constant. It can be shown that if $n_\lambda \nu^2 = 1$, and n is large enough so that $n_\lambda \approx \frac{1}{1-\lambda}$, the eigen-values of $\mathbf{C}(n)$ lie in the above interval with probability exceeding

$$1 - 3M^2 \exp\bigg(-\frac{n_\lambda \delta^2}{54M^2}\bigg). \tag{4.43}$$

In particular, for any $c < \frac{\delta^2}{54}$, by choosing

$$\lambda \geq 1 - \frac{\delta^2 - 54c}{162M^2 \log M},$$

the exponent in the above expression goes to 0 at a polynomial rate. A similar treatment can be applied to other input sequences, *e.g.*, bounded input sequences [63], using the straightforward variant of the Hoeffding's lemma for weighted sums [66]. The above bound on λ is not optimal. In fact, comparing to the Gaussian i.i.d. case, one expects to obtain bounded eigen-values given $\lambda \geq 1 - \frac{C_0}{M}$, for some constant C_0. Studying the eigen-values of structured and exponentially weighted covariance estimates is a hard problem in general. We will state an interesting result in this class.

Suppose that the exponentially weighted random matrix $\mathbf{C}(n)$ is formed by the set $\{x_{ik}\}$, where x_{ij} are i.i.d. Gaussian random variables distributed as $\mathcal{N}(0, \nu^2)$. This is denoted by the *independence assumption* in the jargon of adaptive signal processing [65]. Note that $\mathbf{C}(n)$ can be identified as the empirical estimate of the

covariance matrix through an exponentially weighted moving average. Such random matrices often arise in portfolio optimization techniques (See, for example, [98]). In [98], the eigen-distribution of such matrices is studied and compared to those of Wishart ensembles. Using the resolvent technique (See, for example, [111]), it is shown in [98] that in the limit of $M \to \infty$ and $\lambda \to 1$, with $Q := 1/M(1-\lambda)$ fixed, and $n \to \infty$, the eigenvalues of the matrix $(1-\lambda)\mathbf{C}(n)$ are distributed according to the density

$$\rho(s) = \frac{Qv}{\pi} \tag{4.44}$$

where v is the solution to the non-algebraic equation $\frac{s}{\nu^2} - \frac{vs}{\tan(vs)} + \log(v\nu^2) - \log\sin(vs) - \frac{1}{Q} = 0$.

For example, by solving the above equation numerically for $Q = 2$ and $\nu = 1$, the minimum and maximum eigenvalues in the spectrum of $(1-\lambda)\mathbf{C}(n)$ are found to be 0.30 and 2.37, respectively. As it is shown in [98], for finite but large values of M, the empirical eigen-distribution is very similar to the asymptotic case.

Finally, note that the asymptotic value of $\rho(n) = 1 - \alpha^2/\sigma^2 s_M(n)$ as $n \to \infty$, can be estimated using the minimum eigenvalue of $\mathbf{C}(n)$, which is bounded below by $1-\delta$, with probability exceeding the expression of Eq. (6.5).

Parameter γ: The parameter γ is an additional degree of freedom which controls the trade-off between sparseness of the output (computational complexity) and the MSE. For very small values of γ, the SPARLS algorithm coincides with the RLS algorithm. For very large values of γ, the output will be the zero vector. Thus, there are intermediate values for γ which result in low MSE and sparsity level which is desired. The parameter γ can be fine-tuned according to the application we are

interested in. For example, for estimating the wireless multi-path channel, γ can be optimized with respect to the number of channel taps (sparsity), temporal statistics of the channel and noise level via exhaustive simulations or experiments. Note that γ can be fine-tuned offline for a certain application. Theoretical bounds on γ for near-oracle recovery are discussed in [20] and [116]. There are also some heuristic methods for choosing γ which are discussed in [55]. The noise variance σ^2 can be estimated in various ways, which are discussed in [55] and [74].

Parameter λ: The parameter λ can be fine-tuned based on the time-variation rate of the true vector, as it is done for the RLS algorithm. However, for the SPARLS algorithm we assume that $\lambda \in (0, 1)$, in the cost function given in Eq. (4.16), even when the true vector is constant over time. This is due to the fact that with $\lambda = 1$, which is used for RLS algorithm when the true vector is constant over time, for large values of n, the quadratic term in Eq. (4.16) grows unboundedly and dominates the ℓ_1-penalty term. Hence, the minimizer of the cost function, for large values of n, coincides with that obtained by the RLS algorithm, which is not necessarily sparse. Restricting λ to lie in the open interval $(0, 1)$ maintains a proper scaling between the quadratic and ℓ_1-penalty terms, since the quadratic term will remain bounded over time. The lack of scalability of the Laplacian prior induced by the ℓ_1-penalty term, has led some researchers to employ the Gaussian Scale Mixture (GSM) densities, which are known to be scale invariant (See [6] and [103]). However, there are a number of well-established performance results that show potential near-oracle performance when the Laplacian prior is used (See [20] and [116]). In this regard, we have chosen to use the Laplacian prior. Nevertheless, generalization of the SPARLS algorithm

equipped with other penalization schemes (such as the GSM prior) is possible.

4.5 Simulation Studies

We consider the estimation of a sparse multi-path wireless channel generated by the Jake's model [73]. In the Jake's model, each component of the tap-weight vector is a sample path of a Rayleigh random process with autocorrelation function given by

$$R(n) = \mathrm{J}_0(2\pi n f_d T_s) \tag{4.45}$$

where $\mathrm{J}_0(\cdot)$ is the zeroth order Bessel function, f_d is the Doppler frequency shift and T_s is the channel sampling interval. The dimensionless parameter $f_d T_s$ gives a measure of how fast each tap is changing over time. Note that the case $f_d T_s = 0$ corresponds to a constant tap-weight vector. Thus, the Jake's model covers constant tap-weight vectors as well. For the purpose of simulations, T_s is normalized to 1.

We consider two different input sequences $\{x(i)\}_{i=1}^{\infty}$ for simulations: Gaussian i.i.d. input sequence, where each $x(i)$ is distributed according to $\mathcal{N}(0, 1/M)$, and i.i.d. random Rademacher input sequence, where each $x(i)$ takes the values $\pm 1/\sqrt{M}$ with equal probability. The SNR is defined as $\mathbb{E}\{\|\mathbf{w}\|_2^2\}/\sigma^2$, where σ^2 is the variance of the Gaussian zero-mean observation noise. The locations of the nonzero elements of the tap-weight vector are randomly chosen in the set $\{1, 2, \cdots, M\}$ and the SPARLS algorithm has no knowledge of these locations. Also, all the simulations are done with $K = 1$, *i.e.*, a single LCEM iteration per new data and the column updates are performed using the LCU subroutine. Finally, a choice of $\alpha = \sigma/2$ has been used

(Please see Section 4.4.5).

We compare the performance of the SPARLS and RLS with respect to two performance measures. The first measure is the MSE defined as

$$\text{MSE} := \frac{\mathbb{E}\{\|\hat{\mathbf{w}} - \mathbf{w}\|_2^2\}}{\mathbb{E}\{\|\mathbf{w}\|_2^2\}} \tag{4.46}$$

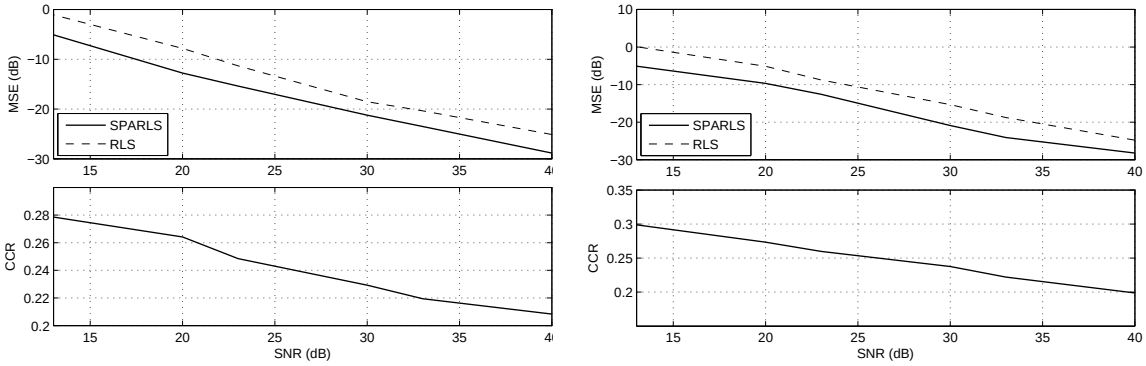
where the averaging is carried out by 50000 Monte Carlo samplings. The number of samples has been chosen large enough to ensure that the uncertainty in the measurements is less than 1%. The second measure is the computational complexity ratio (CCR) which is defined by

$$\text{CCR} := \frac{\text{average number of multiplications for SPARLS}}{\text{average number of multiplications for RLS}} \tag{4.47}$$

4.5.1 Time-invariant Scenario: $f_d = 0$

In this case, the best choice of λ for the RLS algorithm is $\lambda = 1$. As mentioned earlier in Section 4.4.5, in order to maintain the scaling between the quadratic and ℓ_1-penalty terms of the cost function, we choose $\lambda < 1$ for SPARLS. A value of $\lambda = 0.999$ has been chosen for the SPARLS algorithm. The corresponding values of γ are obtained by exhaustive simulations and are listed in Tables I and II. Moreover, we have $L = 5$ and $M = 100$, and both RLS and SPARLS algorithms are run for Gaussian and Rademacher i.i.d. input sequences of length 500.

Figures 4.3(a) and 4.3(b) show the mean squared error and computational complexity ratio of the SPARLS and RLS algorithm for Gaussian and Rademacher i.i.d. sequences, respectively. The SPARLS algorithm gains about 5 dB in MSE and about 75% less computational complexity.



(a) Gaussian i.i.d. input (b) Rademacher i.i.d. input

Figure 4.3: MSE and CCR of RLS and SPARLS vs. SNR for $f_d T_s = 0$.

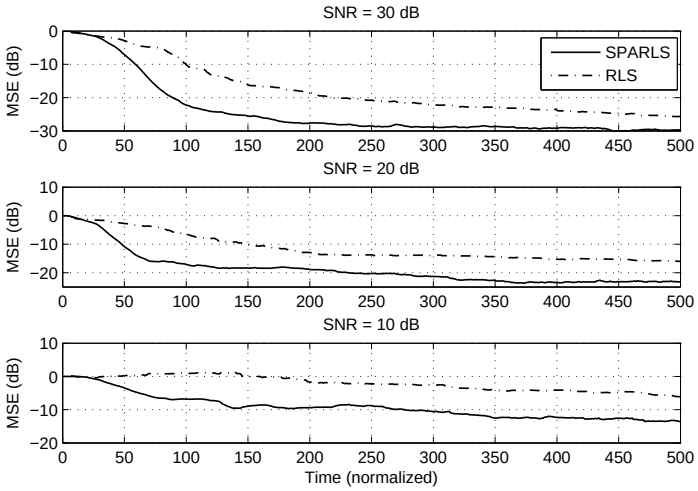


Figure 4.4: MSE of RLS and SPARLS vs. time for SNR $= 10, 20$ and 30 dB and i.i.d. Gaussian input sequence. The time scale is normalized to the signaling interval of the input sequence.

Figure 4.4 shows the time-domain behavior of the SPARLS and RLS algorithms for three different SNR levels of 10 dB, 20 dB and 30 dB, with Gaussian i.i.d. input (the case of Rademacher i.i.d. input is very similar, and thus omitted for brevity).

A value of $\lambda = 0.999$ has been used for the SPARLS algorithm in this case. As it is clear from the figure, for low number of measurements, the SPARLS algorithm significantly outperforms the RLS algorithm in terms of MSE.

4.5.2 Time-varying Scenario: $f_d \neq 0$

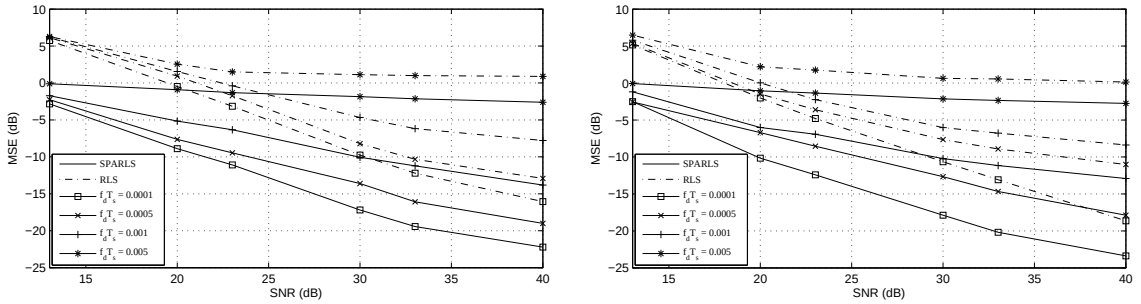
In order to compare the performance of the SPARLS and RLS algorithms, we first need to optimize the RLS algorithm for the given time-varying channel. By exhaustive simulations, the optimum forgetting factor, λ, of the RLS algorithm can be obtained for various choices of SNR and $f_d T_s$. As for the SPARLS algorithm, we perform a partial optimization as follows: we use the values of Tables 1 and 2 for λ and optimize over γ with exhaustive simulations. Note that with such choices of parameters λ and γ, we are comparing a near-optimal parametrization of SPARLS with the optimal parametrization of RLS. The performance of the SPARLS can be further enhanced by simultaneous optimization over both λ and γ. The pairs of (λ, γ) corresponding to the optimal values of γ and λ vs. σ^2 and $f_d T_s$ are summarized in Tables 4.1 and 4.2, for i.i.d. Gaussian and Rademacher input sequences, respectively.

Table 4.1: Optimal values of λ for the RLS algorithm and the corresponding values of γ for the SPARLS algorithm vs. σ^2 and $f_d T_s$, for i.i.d. Gaussian input.

σ^2 \ $f_d T_s$	0	0.0001	0.0005	0.001	0.005
0.0001	(0.999, 100)	(0.97, 100)	(0.96, 100)	(0.97, 100)	(0.99, 200)
0.0005	(0.999, 50)	(0.97, 50)	(0.97, 50)	(0.98, 40)	(0.99, 100)
0.001	(0.999, 35)	(0.98, 35)	(0.98, 30)	(0.99, 25)	(0.99, 60)
0.005	(0.999, 15)	(0.99, 15)	(0.99, 15)	(0.99, 10)	(0.99, 30)
0.01	(0.999, 13)	(0.99, 10)	(0.99, 8)	(0.99, 8)	(0.99, 15)
0.05	(0.999, 3)	(0.99, 3)	(0.99, 3)	(0.99, 3)	(0.99, 5)

Table 4.2: Optimal values of λ for the RLS algorithm and the corresponding values of γ for the SPARLS algorithm vs. σ^2 and f_dT_s, for i.i.d. Rademacher input.

σ^2 \ f_dT_s	0	0.0001	0.0005	0.001	0.005
0.0001	(0.999, 100)	(0.97, 90)	(0.96, 90)	(0.97, 90)	(0.99, 250)
0.0005	(0.999, 50)	(0.97, 50)	(0.97, 45)	(0.98, 45)	(0.99, 100)
0.001	(0.999, 35)	(0.98, 35)	(0.98, 35)	(0.99, 20)	(0.99, 70)
0.005	(0.999, 10)	(0.99, 10)	(0.99, 10)	(0.99, 10)	(0.99, 30)
0.01	(0.999, 8)	(0.99, 5)	(0.99, 5)	(0.99, 5)	(0.99, 10)
0.05	(0.999, 5)	(0.99, 4)	(0.99, 4)	(0.99, 4)	(0.99, 7)

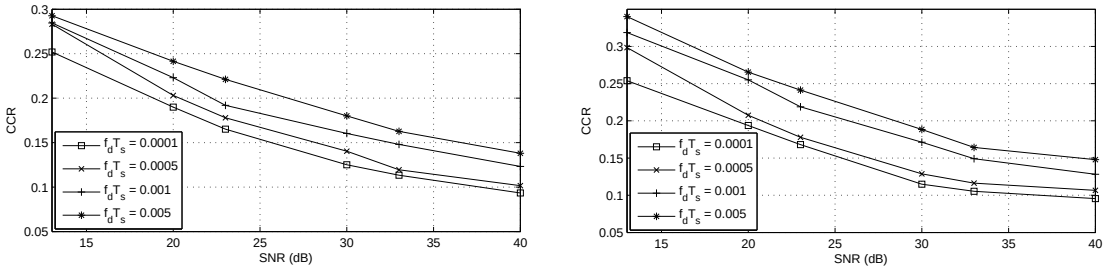


(a) Gaussian i.i.d. input (b) Rademacher i.i.d. input

Figure 4.5: MSE of RLS and SPARLS vs. SNR for $f_dT_s = 0.0001$,0.0005, 0.001 and 0.005.

Figures 4.5(a) and 4.5(b) show the mean squared error of the RLS and SPARLS algorithms for $f_dT_s = 0.0001, 0.0005, 0.001$ and 0.005, with $L = 5$ and $M = 100$, for i.i.d. Gaussian and Rademacher inputs, respectively. Similarly, Figures 4.6(a) and 4.6(b) show the computational complexity ratio of the SPARLS algorithm compared RLS for the same values of f_dT_s, for i.i.d. Gaussian and Rademacher inputs, respectively.

In both cases, the SPARLS algorithm outperforms the RLS algorithm with about 7 dB gain in the MSE performance. Moreover, the computational complexity of the



(a) Gaussian i.i.d. input (b) Rademacher i.i.d. input

Figure 4.6: CCR of RLS and SPARLS vs. SNR for $f_dT_s = 0.0001$,0.0005, 0.001 and 0.005.

SPARLS (using the LCU subroutine) is about 80% less than that of RLS on average.

4.6 Proof of Theorem 4.1

Suppose that we perform the LCEM algorithm a total of K times in each step. The estimate at time $n+1$ can be written as

$$\hat{\mathbf{w}}(n+1) = \underbrace{\mathcal{M}_n \circ \mathcal{M}_n \circ \cdots \circ \mathcal{M}_n}_{K \text{ times}}(\hat{\mathbf{w}}(n)) = \mathcal{M}_n^K(\hat{\mathbf{w}}(n)). \tag{4.48}$$

Now, consider the objective function $f_n(\mathbf{w})$:

$$\begin{aligned} f_n(\mathbf{w}) \;\; = \;\; \text{const.} + \frac{1}{2\sigma^2}\Big\{ & \mathbf{w}^*\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{X}(n)\mathbf{w} - 2\,\mathrm{Re}\left\{\mathbf{w}_0^*\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{X}(n)\mathbf{w}\right\} \\ & -2\,\mathrm{Re}\left\{\boldsymbol{\eta}^*(n)\mathbf{D}(n)\mathbf{X}(n)\mathbf{w}\right\}\Big\} + \gamma\|\mathbf{w}\|_1 \end{aligned} \tag{4.49}$$

Using the stationarity hypothesis, we assume that the input vector $\mathbf{x}(i)$ at time i is a random vector with zero mean entries and covariance $\mathbf{R}_x$. For n large enough, the entries of the matrix $\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{X}(n)$ can be written as

$$(\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{X}(n))_{ij} = \sum_{k=0}^{n-1} \lambda^k x_i(k)x_j^*(k) \to \frac{1}{1-\lambda}(\mathbf{R}_x)_{ij}, \tag{4.50}$$

where we have invoked the strong law of large numbers for weighted sums [38]. If we take the expectation of the objective function with respect to $\boldsymbol{\eta}(n)$, we get:

$$\mathbb{E}_{\eta}\{f_n(\mathbf{w})\} \to f(\mathbf{w}) := \text{const.} + \frac{1}{2\sigma^2(1-\lambda)}\Big\{\mathbf{w}^*\mathbf{R}_x\mathbf{w} - 2\,\text{Re}\left\{\mathbf{w}_0^*\mathbf{R}_x\mathbf{w}\right\}\Big\} + \gamma\|\mathbf{w}\|_1 \tag{4.51}$$

as $n \to \infty$. Note that $f(\mathbf{w})$ is independent of n. From the continuity of the minimizer of $f_n(\mathbf{w})$ in $\boldsymbol{\eta}(n)$, we conclude that

$$\mathbb{E}_{\eta}\{\tilde{\mathbf{w}}(n)\} \to \tilde{\mathbf{w}}_0 \tag{4.52}$$

as $n \to \infty$ almost surely. The above limit process implies the existence of a limit genie-aided estimate as the number of observations n tends to infinity.

We want to show that the SPARLS algorithm converges to $\tilde{\mathbf{w}}_0$ almost surely. Throughout the rest of the proof, we drop the expectation with respect to $\boldsymbol{\eta}$ for notational simplicity and assume it implicitly in our derivations.

Consider Kn_0 successive iterations of the EM algorithm on a single cost function $f_n(\mathbf{w})$ at time n, resulting in the set of estimates $\{\mathcal{M}_n^i(\hat{\mathbf{w}}(n))\}_{i=1}^{Kn_0}$. It is possible to choose n_0 large enough such that

$$\left|f_n\big(\mathcal{M}_n^{Kn_0}(\hat{\mathbf{w}}(n))\big) - f_n(\tilde{\mathbf{w}}(n))\right| < \epsilon/3 \tag{4.53}$$

due to the guaranteed convergence of the EM algorithm applied to a single cost function $f_n(\mathbf{w})$ [45]. In other words, due to the continuity of $f_n(\mathbf{w})$, we can reach an arbitrarily small neighborhood of $\tilde{\mathbf{w}}(n)$ in finite time by successively applying the EM iteration across the curve $f_n(\mathbf{w})$.

Now, consider applying the SPARLS iterations from time n to $n+n_0-1$, resulting in the estimates $\{\hat{\mathbf{w}}(n+i)\}_{i=1}^{n_0}$, where $\hat{\mathbf{w}}(n+i) := \mathcal{M}_{n+i-1}^{K}(\hat{\mathbf{w}}(n+i-1))$. By the

continuity of the mapping $\mathcal{M}_n$ in the linear and quadratic coefficients of $\mathbf{w}$, and by the continuity of the function $f_n(\mathbf{w})$ in $\mathbf{w}$, we can choose n large enough such that

$$\begin{aligned}&\left|f_{n+n_0}(\hat{\mathbf{w}}(n+n_0)) - f_{n+n_0}(\mathcal{M}_n^{Kn_0}(\hat{\mathbf{w}}(n)))\right| = \\ &\left|f_{n+n_0}\Big(\mathcal{M}_{n+n_0-1}^{K} \circ \mathcal{M}_{n+n_0-2}^{K} \circ \cdots \circ \mathcal{M}_{n}^{K}(\hat{\mathbf{w}}(n))\Big) - f_{n+n_0}\big(\mathcal{M}_n^{Kn_0}(\hat{\mathbf{w}}(n))\big)\right| \leq \epsilon/3\end{aligned} \tag{4.54}$$

Since the coefficients of the linear and quadratic terms in $f_n(\mathbf{w})$ are independent of n in the limit of $n \to \infty$, $f_n(\mathbf{w})$ tends to $f(\mathbf{w})$ in a point-wise fashion. Let

$$W := \mathcal{B}_{2\|\tilde{\mathbf{w}}_0\|_2}(0) := \left\{\mathbf{w} \in \mathbb{C}^M : \|\mathbf{w}\|_2 \leq 2\|\tilde{\mathbf{w}}_0\|_2\right\} \tag{4.55}$$

Since $\mathbb{C}^M$ is a separable metric space, by the Egorov's theorem [127], the point-wise convergence of the continuous bounded functions $f_n(\mathbf{w})$ to $f(\mathbf{w})$ in the compact set W, implies uniform convergence everywhere except on some subset of arbitrarily small measure. Hence, for any positive $\epsilon > 0$, there exists an integer N such that for all $n > N$ we have

$$\max_{\mathbf{w}\in W} |f_n(\mathbf{w}) - f(\mathbf{w})| < \epsilon/12. \tag{4.56}$$

By Eqs. (4.53) and (4.54), it is implied that for ϵ small enough, $\hat{\mathbf{w}}(n+n_0)$ and $\mathcal{M}_n^{Kn_0}(\hat{\mathbf{w}}(n))$ are in a small neighborhood of $\tilde{\mathbf{w}}(n)$ (due to the continuity of $f_n(\cdot)$ and $f_{n+n_0}(\cdot)$). Hence, by choosing n large enough, the points $\hat{\mathbf{w}}(n+n_0)$ and $\mathcal{M}_n^{Kn_0}(\hat{\mathbf{w}}(n))$ lie inside the set W. We thus have

$$\begin{aligned}\left|f_n(\hat{\mathbf{w}}(n+n_0)) - f_n(\mathcal{M}_n^{Kn_0}(\hat{\mathbf{w}}(n)))\right| \leq &\left|f_{n+n_0}(\hat{\mathbf{w}}(n+n_0)) - f_n(\hat{\mathbf{w}}(n+n_0))\right| \\ &+ \left|f_{n+n_0}(\mathcal{M}_n^{Kn_0}(\hat{\mathbf{w}}(n))) - f_n(\mathcal{M}_n^{Kn_0}(\hat{\mathbf{w}}(n)))\right| \\ &+ \left|f_{n+n_0}(\mathcal{M}_n^{Kn_0}(\hat{\mathbf{w}}(n))) - f_{n+n_0}(\hat{\mathbf{w}}(n+n_0))\right| \\ \leq &4\epsilon/12 + \epsilon/3 = 2\epsilon/3\end{aligned} \tag{4.57}$$

Hence, after n_0 iterations of the SPARLS algorithm, we have

$$\begin{aligned}\big|f_n(\hat{\mathbf{w}}(n+n_0)) - f_n(\tilde{\mathbf{w}}(n))\big| \leq & \big|f_n(\hat{\mathbf{w}}(n+n_0)) - f_n(\mathcal{M}_n^{Kn_0}(\hat{\mathbf{w}}(n)))\big| \\ & + \big|f_n(\mathcal{M}_n^{Kn_0}(\hat{\mathbf{w}}(n))) - f_n(\tilde{\mathbf{w}}(n))\big| < 2\epsilon/3 + \epsilon/3 = \epsilon.\end{aligned}$$

Therefore, after n_0 iterations, we can reach an arbitrarily small neighborhood of $\tilde{\mathbf{w}}(n)$ for all n, due to the continuity of $f_n(\mathbf{w})$. Since $\tilde{\mathbf{w}}(n) \to \tilde{\mathbf{w}}_0$, we can reach an arbitrarily small neighborhood of $\tilde{\mathbf{w}}_0$ in finite time for all n. Therefore, the SPARLS algorithm converges to $\tilde{\mathbf{w}}_0$ almost surely.

4.7 Steady State Error Analysis: Derivations

First, we briefly overview the convergence properties of the EM algorithm. The global and componentwise convergence of the EM algorithm has been widely studied in the statistics literature (See, for example, [45] and [91]). Suppose, for the moment, that the mapping $\mathcal{M}_n$ is differentiable at $\tilde{\mathbf{w}}(n)$, the maximizer of the objective function in Eq. (4.18). We can therefore write the Taylor expansion as follows:

$$\hat{\mathbf{w}}^{(\ell+1)}(n) - \tilde{\mathbf{w}}(n) = D\mathcal{M}_n(\tilde{\mathbf{w}}(n))\big(\hat{\mathbf{w}}^{(\ell)}(n) - \tilde{\mathbf{w}}(n)\big) + \mathcal{O}\big(\big\|\hat{\mathbf{w}}^{(\ell)}(n) - \tilde{\mathbf{w}}(n)\ big\|^2\big), \quad (4.58)$$

where $D\mathcal{M}_n$ is the Jacobian of the mapping $\mathcal{M}_n$ and we have used the fact that $\tilde{\mathbf{w}}(n)$ is a fixed point for the mapping $\mathcal{M}_n$. Hence, in a sufficiently small neighborhood of $\tilde{\mathbf{w}}(n)$, the EM algorithm is simply a linear mapping. However, in our case the mapping $\mathcal{M}_n$ is not differentiable, since the soft thresholding function is not differentiable at points $-\gamma\alpha^2$ and $\gamma\alpha^2$. We can therefore use the sub-differential of the mapping $\mathcal{M}_n$ in order to study its behavior in a neighborhood of $\tilde{\mathbf{w}}(n)$. Let $\mathcal{E} : \mathbb{C}^M \longmapsto \mathbb{C}^M$ be a

mapping defined as:

$$\mathcal{E}(\mathbf{w}) := \big(\mathbf{I} - \frac{\alpha^2}{\sigma^2}\mathbf{X}^*\mathbf{D}\mathbf{X}\big)\mathbf{w} + \frac{\alpha^2}{\sigma^2}\mathbf{X}^*\mathbf{D}\mathbf{d}. \tag{4.59}$$

Note that we have dropped the dependence on n for notational convenience. The mapping $\mathcal{M}$ is then simply given by $\mathcal{M}(\mathbf{w}) = \mathcal{S} \circ \mathcal{E}(\mathbf{w})$, where $\mathcal{S}(\cdot)$ is the elementwise soft thresholding function, defined in Eq. (4.25). Although the mapping $\mathcal{E}$ is differentiable, the mapping $\mathcal{S}$ is not. However, as we will see later on, the restriction on the convergence properties of the EM algorithm does not arise from the M step and is mainly due to the E step. In order to simplify the notational presentation, we assume that $\mathbf{w} \in \mathbb{R}^M$. Due to the trivial isomorphism of the vector spaces $\mathbb{C}^M$ and $\mathbb{R}^{2M}$ over the field of real numbers, generalization to $\mathbf{w} \in \mathbb{C}^M$ is straightforward. We can define the sub-differential of the mapping $\mathcal{S}$ as follows (See, for example, [104]):

$$\partial\mathcal{S}(\mathbf{w}) = \mathrm{diag}(h_1, h_2, \cdots, h_M) \tag{4.60}$$

where

$$h_i := \begin{cases} 1 & |w_i| > \gamma\alpha^2 \\ 0 \le h_i \le 1 & |w_i| = \gamma\alpha^2 \\ 0 & |w_i| < \gamma\alpha^2 \end{cases} \tag{4.61}$$

In addition, from the chain rule for sub-differentials [104], we have

$$\partial\mathcal{M}(\mathbf{w}) = \partial(\mathcal{S} \circ \mathcal{E}(\mathbf{w})) = \big(\partial\mathcal{S}(\mathcal{E}(\mathbf{w}))\big)^*\big(\mathbf{I} - \frac{\alpha^2}{\sigma^2}\mathbf{X}^*\mathbf{D}\mathbf{X}\big) \tag{4.62}$$

Therefore, by an appropriate choice of the sub-differential of $\mathcal{S}$ at $\tilde{\mathbf{w}}(n)$, we can locally approximate the EM iteration by

$$\hat{\mathbf{w}}^{(\ell+1)}(n) - \tilde{\mathbf{w}}(n) \approx \big(\partial\mathcal{S}(\mathcal{E}(\tilde{\mathbf{w}}(n)))\big)^*\big(\mathbf{I} - \frac{\alpha^2}{\sigma^2}\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{X}(n)\big)\big(\hat{\mathbf{w}}^{(\ell)}(n) - \tilde{\mathbf{w}}(n)\big) \tag{4.63}$$

From the convergence results of [45] and [91], it is known that the linear convergence rate of the EM algorithm is governed by the maximum eigenvalue of the Jacobian $D\mathcal{M}$. In our case, we need to consider the maximum eigenvalue of $\partial(\mathcal{S}\circ\mathcal{E}(\tilde{\mathbf{w}}(n)))$. Clearly, the maximum eigenvalue of the diagonal matrix $\partial\mathcal{S}(\mathcal{E}(\tilde{\mathbf{w}}(n))$ is bounded above by 1, since all its diagonal elements h_i are bounded as $0 \leq h_i \leq 1$. In fact, the maximum eigenvalue of $\partial\mathcal{S}(\mathcal{E}(\tilde{\mathbf{w}}(n))$ is equal to 1, unless all the elements of $\tilde{\mathbf{w}}(n)$ are in the range $-\gamma\alpha^2 \leq w_i \leq \gamma\alpha^2$, which is very unlikely to happen. This account for the earlier claim that the maximum eigenvalue of $\partial\mathcal{S}$ does not play a significant role in the convergence rate, since it most likely is equal to 1. Therefore, the rate of convergence is governed by the maximum eigenvalue of the matrix $\mathbf{I} - \frac{\alpha^2}{\sigma^2}\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{X}(n)$, which is given by

$$\rho(n) := 1 - \frac{\alpha^2}{\sigma^2} s_M(n), \tag{4.64}$$

where $s_M(n)$ is the minimum eigenvalue of $\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{X}(n)$ (there is more to say about the asymptotic behavior of $\rho(n)$, as $n \to \infty$, in Section 4.4.5). If we perform the EM iteration a total of K times, we can write:

$$\begin{aligned}\|\hat{\mathbf{w}}(n+1) - \tilde{\mathbf{w}}(n)\|_2 \leq & \left\| \left(\partial\mathcal{M}(\tilde{\mathbf{w}}(n))\right)^K \right\|_2 \|\hat{\mathbf{w}}(n) - \tilde{\mathbf{w}}(n)\|_2 \\ \leq & \rho(n)^K \|\hat{\mathbf{w}}(n) - \tilde{\mathbf{w}}(n)\|_2,\end{aligned} \tag{4.65}$$

for $\hat{\mathbf{w}}(n)$ in a small neighborhood of $\tilde{\mathbf{w}}(n)$.

Recall that from Lemma 3 of [116], we know that the maximizer of the objective function given in Eq. (4.18) is unique if $\mathbf{X}_{\mathcal{I}}(n)$ is maximal rank, where $\mathcal{I} = \mathsf{supp}(\mathbf{w}(n))$. Moreover, Lemma 6 of Tropp in [116] establishes that if γ sat-

isfies

$$\gamma \geq \frac{\|\mathbf{X}^*(n)\mathbf{D}^{1/2}(n)(\mathbf{D}^{1/2}(n)\mathbf{X}_{\mathcal{I}}(n))^{+}\mathbf{D}^{1/2}(n)\boldsymbol{\eta}_{\mathcal{I}}(n)\|_{\infty}}{1-\max\limits_{i\notin\mathcal{I}}|\mathbf{X}_i^*(n)\mathbf{D}(n)\mathbf{X}(n)\mathbf{g}(n)|}, \tag{4.66}$$

we have

$$\hat{\mathbf{w}}_g(n)-\tilde{\mathbf{w}}(n)=\gamma\sigma^2\Big(\mathbf{X}_{\mathcal{I}}^*(n)\mathbf{D}(n)\mathbf{X}_{\mathcal{I}}(n)\Big)^{-1}\mathbf{g}(n) \tag{4.67}$$

where $\hat{\mathbf{w}}_g(n)$ is the *genie-aided* estimate of $\mathbf{w}(n)$ given by

$$\hat{\mathbf{w}}_g(n):=\big(\mathbf{D}^{1/2}(n)\mathbf{X}_{\mathcal{I}}(n)\big)^{+}\mathbf{D}^{1/2}(n)\mathbf{d}_{\mathcal{I}}(n)=\mathbf{w}(n)+\big(\mathbf{D}^{1/2}(n)\mathbf{X}_{\mathcal{I}}(n)\big)^{+}\boldsymbol{\eta}_{\mathcal{I}}(n) \tag{4.68}$$

and $\mathbf{g}(n)$ is in the sub-gradient set of $\|\tilde{\mathbf{w}}(n)\|_1$. The genie-aided estimate corresponds to the least square solution when a genie has provided the support of $\mathbf{w}(n)$ to the estimator and is considered to be a theoretical performance benchmark for the estimation of sparse vectors. Using the relations between $\hat{\mathbf{w}}_g(n)$, $\tilde{\mathbf{w}}(n)$ and $\mathbf{w}(n)$ and triangle inequality we can write:

$$\begin{aligned}\epsilon(n+1)=&\mathbb{E}_{\eta}\Big\{\|\hat{\mathbf{w}}(n+1)-\tilde{\mathbf{w}}(n)+\tilde{\mathbf{w}}(n)-\mathbf{w}(n)+\mathbf{w}(n)-\mathbf{w}(n+1)\|\Big\}\\ \leq&\rho(n)^K\epsilon(n)+\mathbb{E}_{\eta}\Big\{\big\|(\mathbf{D}^{1/2}(n)\mathbf{X}_{\mathcal{I}}(n))^{+}\boldsymbol{\eta}_{\mathcal{I}}(n)\big\|_2\Big\}\\ &+\gamma\sigma^2\Big\|\Big(\mathbf{X}_{\mathcal{I}}^*(n)\mathbf{D}(n)\mathbf{X}_{\mathcal{I}}(n)\Big)^{-1}\Big\|_{2,\infty}\\ &+\|\mathbf{w}(n+1)-\mathbf{w}(n)\|_2.\end{aligned} \tag{4.69}$$

where the $(2,\infty)$-norm of a matrix $\mathbf{A}$ is defined as $\|\mathbf{A}\|_{2,\infty}:=\max_{\mathbf{x}:\mathbf{x}\neq\mathbf{0}}\frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_{\infty}}$.

4.8 Proof of Proposition 4.2

For the RLS algorithm (with $\lambda=1$), the error expression is given by

$$\epsilon_{RLS}(n+1):=\mathbb{E}_{\eta}\Big\{\big\|\hat{\mathbf{w}}_{RLS}(n+1)-\mathbf{w}(n+1)\big\|_2\Big\}=\mathbb{E}_{\eta}\Big\{\big\|\mathbf{X}^{+}(n)\boldsymbol{\eta}(n)\big\|_2\Big\}. \tag{4.70}$$

According to Eq. (4.38) the corresponding error expression for the SPARLS algorithm in a stationary environment is upper bounded as

$$\epsilon(n+1) \leq \rho^K(n)\epsilon(n) + \mathbb{E}_\eta\Big\{\big\|(\mathbf{D}^{1/2}(n)\mathbf{X}_\mathcal{I}(n))^+\boldsymbol{\eta}_\mathcal{I}(n)\big\|_2\Big\} + \gamma\alpha^2\Big\|\big(\mathbf{X}^*_\mathcal{I}(n)\mathbf{D}(n)\mathbf{X}_\mathcal{I}(n)\big)^{-1}\Big\|_{2,\infty}. \tag{4.71}$$

Let $\mu(n)$ be the coherence of the matrix $\mathbf{X}^*(n)\mathbf{D}(n)\mathbf{X}(n)$. Now, we claim that for $n_0 < \infty$ and $L < aM < 1/(3\mu(n_0))$ for some $a > 0$, one can choose γ_0 and $\lambda_0 < 1$ such that

$$\mathbb{E}_\eta\Big\{\big\|(\mathbf{D}^{1/2}(n_0)\mathbf{X}_\mathcal{I}(n_0))^+\boldsymbol{\eta}_\mathcal{I}(n_0)\big\|_2\Big\} + \gamma\alpha^2\Big\|\big(\mathbf{X}^*_\mathcal{I}(n_0)\mathbf{D}(n_0)\mathbf{X}_\mathcal{I}(n_0)\big)^{-1}\Big\|_{2,\infty} < \mathbb{E}_\eta\Big\{\big\|\mathbf{X}^+(n_0)\boldsymbol{\eta}(n_0)\big\|_2\Big\}. \tag{4.72}$$

with high probability. First, note that the claim is obviously true for $\lambda = 1$, for an appropriate choice of γ and a sufficiently incoherent measurement matrix $\mathbf{X}(n)$, thanks to the results of Tropp [116] and Ben-Haim et al. [20] on the near-oracle performance of Subspace Pursuit. Next, by the continuity of the pseudo-inverse operator in the argument $\mathbf{D}^{1/2}(n)$, the continuity of the coherence $\mu(n)$ in λ, and finally the continuity of the lower bound on γ in λ (See Eq. (4.66) or Lemma 6 of [116]), there exist $\lambda_0 < 1$ and γ_0 such that the above inequality holds.

Note that with the appropriate choice of γ_0 as in [116] and [20], $|\mathcal{I}| \leq L$ with high probability. Hence, for $L \ll M$ (low sparsity regime), the left hand side of Eq. (4.72) can be significantly smaller than the right hand side. Now, given that the SPARLS algorithm converges to a fixed point (Theorem 4.1), for n_0 large enough, the average instantaneous error of SPARLS, $\epsilon(n_0)$, is a factor of $1/(1-\rho(n_0)^K)$ away from the left hand side of Eq. (4.72). By choosing K appropriately, one can guarantee that

$\rho(n_0)^K \ll 1$. Hence, there exists $0 < a < \min\{\frac{1}{3\mu(n_0)M}, 1\}$ such that for $L/M < a$, we have $\epsilon(n_0) < \epsilon_{RLS}(n_0)$. This establishes the statement of the theorem.

Chapter 5

Adaptive Algorithms for Sparse System Identification

5.1 Introduction

This chapter presents a compendium of adaptive algorithms for sparse system identification, which are based on the Expectation-Maximization framework [77, 78]. The expectation step is carried out by Kalman filtering while the maximization step corresponds to a soft-thresholding function due to the ℓ_1-regularization. This algorithmic separation and the particular form of the assumed parameter dynamics, enables us to develop several sparse variants including RLS, LMS and fast RLS schemes.

As a particular application, adaptive identification of sparse nonlinear systems described by finite Volterra models is considered. Very often power amplifiers located at an access point of a downlink channel (e.g., base stations in cellular systems and repeaters for satellite links) operate close to saturation in order to achieve power effi-

ciency. In such cases nonlinearities cannot be ignored without significant loss of performance. A widely used method to describe nonlinear systems is based on Volterra models [21, Ch. 14]. Volterra models employ a large number of parameters which increases exponentially with the order of nonlinearity. Therefore, there is a strong need to reduce the number of terms by only considering those terms that actually contribute to the output. All previous algorithms for estimating Volterra kernels treat each kernel equally and identify the complete set of kernels. The major drawback of estimating the complete set of kernels is the large computational/implementation cost. It is shown that significant performance gains are achieved if insignificant kernels are ignored. The proposed algorithms are applied to two channel settings. The first setting takes into account the Power Amplifier (PA) dynamic nonlinearities but ignores multipath fading. The second setting includes fading as well.

The rest of this chapter is organized as follows. Sparse problem formulation is addressed in Section 5.2. The proposed algorithms for adaptive tracking of sparse systems are given in section 5.3. In section 5.4 two important sparse nonlinear channels are discussed. Simulations results are presented in section 5.5.

5.2 Sparse Problem Formulation

Recall the problem of estimating a *sparse* vector, $\mathbf{w}$, from noisy linear measurements of the form

$$y(i) = \mathbf{x}^*(i)\mathbf{w} + v(i), \quad 1 \leq i \leq n, \tag{5.1}$$

where $y(i)$ and $\mathbf{x}(i)$ are the system output and input (at time i) and $v(i)$ is zero mean, independent and identically distributed (i.i.d) Gaussian noise (of variance σ_v^2).[1] The complex parameter vector $\mathbf{w} \in \mathbb{C}^M$ is said to be $L-$sparse if it has s or fewer non-zero elements and $L \ll M$. Let $\mathsf{supp}(\mathbf{w})$ denote the support set of $\mathbf{w}$, $\mathsf{supp}(\mathbf{w}) = \{j : h_j \neq 0\}$, namely the set of nonzero coefficients. The number of elements of the support of $\mathbf{w}$ defines the ℓ_0 quasi-norm of $\mathbf{w}$. Then $\mathbf{w}$ is $L-$sparse if $\|\mathbf{w}\|_0 \leq L \ll M$.

Collecting n successive samples from the above equation into a column format results in the following system of linear equations

$$\mathbf{y}(n) = \mathbf{X}(n)\mathbf{w} + \mathbf{v}(n) \tag{5.2}$$

where $\mathbf{y}(n) = [y(1), \cdots, y(n)]^T$, $\mathbf{X}(n) = [\mathbf{x}^*(1), \cdots, \mathbf{x}^*(n)]^T$, $\mathbf{v}(n) = [v(1), \cdots, v(n)]^T$. Eq. (5.2) provides a noisy representation of a block of successive received samples in terms of the columns of $\mathbf{X}(n) \in \mathbb{C}^{n \times M}$.

As discussed in Chapter 4, we intend to adaptively estimate $\mathbf{w}$ by solving the following optimization problem:

$$\min_{\mathbf{w}} \left\{ \frac{1}{2}\|\mathbf{y}(n) - \mathbf{X}(n)\mathbf{w}\|_2^2 + \gamma\|\mathbf{w}\|_1 \right\}. \tag{P_{ℓ_1}}$$

5.3 Adaptive algorithms

In this section, we propose a family of sparse adaptive algorithms. Generalizing the stationary environment described by Eq. (5.2), we now consider time-varying parameters. Let $\mathbf{w}(n)$ denote the parameter vector at time n. Motivated by the adaptive filtering practice we shall assume that the parameter dynamics are governed

[1] Our analysis assumes a system with complex valued variables.

by the first order model of Eq. (5.3). The derived family of sparse adaptive algorithms is based on the Expectation Maximization (EM) framework.

Let us start by considering a model that captures the dynamics of the unknown parameter vector $\mathbf{w}(n)$. A popular technique in the adaptive filtering literature is to describe parameter dynamics by the first-order model [85, p. 60], [108, p. 272].

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{q}_{|\Lambda_0}(n) = \mathbf{w}_0 + \sum_{i=1}^{n} \mathbf{q}_{|\Lambda_0}(i); \quad \mathbf{w}_0 \sim \mathcal{N}(\overline{\mathbf{w}}_0, \sigma_0^2 \mathbf{I}_{|\Lambda_0}). \tag{5.3}$$

The initial parameter $\mathbf{w}_0$ is normally distributed with mean $\overline{\mathbf{w}}_0$ and uncorrelated coefficients with variance σ_0^2. Λ_0 denotes the true support set of $\mathbf{w}_0$, i.e. the set of the non-zero coefficients of $\mathbf{w}_0$. The noise term $\mathbf{q}(n)$ is zero outside Λ_0 ($q_i(n) \neq 0$ iff $i \in \Lambda_0$) and zero-mean Gaussian inside Λ_0 with diagonal covariance matrix $\mathbf{R}_{|\Lambda_0}(n)$ having diagonal elements $\sigma_{q_1}^2(n), \ldots, \sigma_{q_L}^2(n)$, where L is the ℓ_0-norm of $\mathbf{w}_0$. The variances $\{\sigma_{q_i}^2(n)\}_{i=1}^{L}$ are in general allowed to vary with time. The stochastic processes $\mathbf{v}(n)$, $\mathbf{q}(n)$ and the random variable $\mathbf{w}_0$ are mutually independent.

Another way of expressing the mathematical program (P_{ℓ_1}) is as a Maximum Likelihood (ML) estimation problem augmented by an ℓ_1-penalty, i.e.

$$\widehat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} \big\{ \log p(\mathbf{y}(n); \boldsymbol{\theta}) - \gamma \|\boldsymbol{\theta}\|_1 \big\} \tag{5.4}$$

where $\boldsymbol{\theta}$ is the unknown parameter vector. Parameter estimation is carried out by maximizing the penalized log-likelihood, which in many cases is complex and is either difficult or impossible to compute directly or optimize. In such cases, the Expectation-Maximization (EM) algorithm [45, 53, 55] can be used to maximize $\log p(\mathbf{y}(n); \boldsymbol{\theta}) - \gamma \|\boldsymbol{\theta}\|_1$, without explicitly computing it. Although the EM algorithm is a framework of iterative algorithms, the derived adaptive algorithms employ only one iteration per

time update for computational and storage efficiency [53]. This is highly desirable in time-varying systems, where the algorithm is expected to track system variations.

In order to apply the Expectation-Maximization algorithm we have to specify the complete and incomplete data. The vector $\mathbf{w}(n)$ at time n is taken to represent the complete data vector, whereas $\mathbf{y}(n)$ accounts for the incomplete data [53, p. 31-33]. The EM algorithm iterates between estimating the conditional expectation of the complete data given the incomplete data and a current estimate of the parameters (the **E-step**), and maximizing the conditional expectation of the complete log-likelihood minus an ℓ_1-penalty (the **M-step**). The two steps are summarized by the following equation:

$$\widehat{\boldsymbol{\theta}}(n) = \arg\max_{\boldsymbol{\theta}} \left\{ \mathbb{E}_{p(\mathbf{w}(n)|\mathbf{y}(n);\widehat{\boldsymbol{\theta}}(n-1))} \big[\log p(\mathbf{w}(n);\boldsymbol{\theta}) \big] - \gamma \|\boldsymbol{\theta}\|_1 \right\}. \tag{5.5}$$

The EM algorithm aims to maximize the conditional expectation of the log-likelihood of the complete data, $\log p(\mathbf{w}(n);\boldsymbol{\theta})$, given the incomplete data $\mathbf{y}(n)$ and the current estimate of the parameters $\widehat{\boldsymbol{\theta}}(n-1)$.

The analysis starts from the prior $p(\mathbf{w}(n);\boldsymbol{\theta})$, which we need to carefully define. Suppose that at time n, we are given the estimate $\widehat{\boldsymbol{\theta}}(n-1)$ of the parameter vector, which has a Gaussian distribution with mean $\mathbf{w}(n-1)$ and covariance $\mathbf{P}(n-1)$. Then, at time n, we assume the following multivariate Gaussian prior on $\mathbf{w}(n)$:

$$p(\mathbf{w}(n);\boldsymbol{\theta}) = \frac{(2\pi)^{-M/2}}{\sqrt{\det \boldsymbol{\Sigma}(n)}} \exp\left\{ -\frac{1}{2} \Big(\mathbf{w}(n) - \boldsymbol{\theta} \Big)^* \boldsymbol{\Sigma}^{-1}(n) \Big(\mathbf{w}(n) - \boldsymbol{\theta} \Big) \right\} \tag{5.6}$$

where $\boldsymbol{\theta}$ is the unknown parameter vector, representing the mean of the channel vector at time n, and

$$\boldsymbol{\Sigma}(n) := \mathbb{E}\left[\big(\mathbf{w}(n) - \boldsymbol{\theta} \big) \big(\mathbf{w}(n) - \boldsymbol{\theta} \big)^* \right] = \operatorname{diag}\big(\mathbf{P}(n-1) + \mathbf{R}(n) \big)$$

is the diagonal covariance matrix of the complete data at time n. Now, we take the conditional expectation of Eq. (5.6) with respect to the observed data $\mathbf{y}(n)$ given the parameter estimate $\widehat{\boldsymbol{\theta}}(n-1)$. This gives us the Expectation step of the EM algorithm:

$$\begin{aligned}
\textbf{E-step :} \quad Q\big(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}(n-1)\big) &= \mathbb{E}_{p(\mathbf{w}(n)|\mathbf{y}(n);\widehat{\boldsymbol{\theta}}(n-1))}\left[\log p(\mathbf{w}(n);\boldsymbol{\theta})\right] && (5.7)\\
&= -(M/2)\log 2\pi + 1/2\boldsymbol{\theta}^*\boldsymbol{\Sigma}^{-1}(n)\mathbb{E}[\mathbf{w}(n)|\mathbf{y}(n);\widehat{\boldsymbol{\theta}}(n-1)] && (5.8)\\
&+ 1/2\mathbb{E}[\mathbf{w}^*(n)|\mathbf{y}(n);\widehat{\boldsymbol{\theta}}(n-1)]\boldsymbol{\Sigma}^{-1}(n)\boldsymbol{\theta} - 1/2\boldsymbol{\theta}^*\boldsymbol{\Sigma}^{-1}(n)\boldsymbol{\theta}\\
&- 1/2\mathrm{Tr}\big[\boldsymbol{\Sigma}^{-1}(n)\mathbb{E}[\mathbf{w}(n)\mathbf{w}^*(n)|\mathbf{y}(n);\widehat{\boldsymbol{\theta}}(n-1)]\big] - 1/2\log(|\boldsymbol{\Sigma}(n)|)\\
&= \text{constant} + \boldsymbol{\theta}^*\boldsymbol{\Sigma}^{-1}(n)\mathbb{E}[\mathbf{w}(n)|\mathbf{y}(n);\widehat{\boldsymbol{\theta}}(n-1)] - \frac{1}{2}\boldsymbol{\theta}^*\boldsymbol{\Sigma}^{-1}(n)\boldsymbol{\theta}
\end{aligned}$$

where the constant incorporates all terms that do not involve $\boldsymbol{\theta}$ and hence do not affect the maximization. The Maximization step of the EM algorithm, described below, calculates the maximum of the penalized Q-function

$$\textbf{M-step :} \quad \widehat{\boldsymbol{\theta}}(n) = \arg\max_{\boldsymbol{\theta}} \Big\{ Q\big(\boldsymbol{\theta}, \widehat{\boldsymbol{\theta}}(n-1)\big) - \gamma\|\boldsymbol{\theta}\|_1 \Big\} \tag{5.9}$$

which in turn leads to the component-wise *soft thresholding* function [55], given by

$$\widehat{\boldsymbol{\theta}}_i(n) = \mathrm{sgn}\left(\boldsymbol{\psi}_i(n)\right)\left[|\boldsymbol{\psi}_i(n)| - \gamma\boldsymbol{\Sigma}_{ii}(n)\right]_+, \tag{5.10}$$

where $\boldsymbol{\psi}(n) := \mathbb{E}[\mathbf{w}(n)|\mathbf{y}(n);\widehat{\boldsymbol{\theta}}(n-1)]$, the signum function $\mathrm{sgn}(z)$ is zero only when $z = 0$ in all other cases $\mathrm{sgn}(z) = z/|z|$ for any $z \in \mathbb{C}$ and $(z)_+ = \max(\Re(z), 0) + j\max(\Im(z), 0)$. This operation shrinks coefficients above the threshold in magnitude value.

The conditional expectation of Eq. (5.7) (essentially the **E-step**), may be obtained recursively using the Kalman filter, if a Gaussian prior is assumed on $\boldsymbol{\theta}$ given the past observations. The Kalman filter then determines the posterior probability density

function for $\boldsymbol{\theta}$ recursively over time. Given the following Gaussian prior:

$$p(\mathbf{w}(n)|\mathbf{y}(n-1);\widehat{\boldsymbol{\theta}}(n-1)) = \mathcal{N}\Big(\widehat{\boldsymbol{\theta}}(n-1), \mathbf{P}(n-1) + \mathbf{R}(n)\Big),$$

the conditional expectation $\boldsymbol{\psi}(n) = \mathbb{E}[\mathbf{w}(n)|\mathbf{y}(n); \widehat{\boldsymbol{\theta}}(n-1)]$ can be obtained by the following recursion [85, p. 60], [108, p. 109] (see also Table 5.1):

$$\begin{aligned} \boldsymbol{\psi}(n) &= \widehat{\boldsymbol{\theta}}(n-1) + \mathbf{k}(n)\varepsilon^*(n), \quad \boldsymbol{\psi}(0) = \overline{\mathbf{w}}_0 \\ \mathbf{P}(n) &= \mathbf{P}(n-1) + \mathbf{R}(n) - \mathbf{k}(n)\mathbf{x}^*(n)\mathbf{P}(n-1) \end{aligned} \tag{5.11}$$

where $\mathbf{k}(n)$ is the Kalman gain defined by

$$\mathbf{k}(n) = \frac{\mathbf{P}(n-1)\mathbf{x}(n)}{\sigma_v^2 + \mathbf{x}^*(n)\mathbf{P}(n-1)\mathbf{x}(n)}$$

and $\varepsilon(n)$ denotes the prediction error given by $\varepsilon(n) = y(n) - \mathbf{x}^*(n)\widehat{\boldsymbol{\theta}}(n-1)$ and $\mathbf{x}^*(n)$ denotes the complex conjugate of the vector $\mathbf{x}(n)$. Note that the i^{th} diagonal entry of the prior covariance $\boldsymbol{\Sigma}(n)$ is

$$\boldsymbol{\Sigma}_{ii}(n) = \mathbf{P}_{ii}(n-1) + \mathbf{R}_{ii}(n).$$

5.3.1 EM-Kalman filter

The method outlined above is named EM-Kalman filter and is summarized in Table 5.1. For notational clarity, we replace $\widehat{\boldsymbol{\theta}}(n)$ by $\widehat{\mathbf{w}}(n)$ in the resulting algorithm. The Kalman filter computes $\boldsymbol{\psi}(n)$ under the assumption that the variances σ_v^2 and $\{\sigma_{q_i}^2(n)\}_i^L$ are known. The noise variances can be estimated in various ways. One method is to use the Maximum Likelihood estimates. These estimates can be obtained

by maximizing the Q-function as is done in [126]

$$\sigma_v^2 = \frac{1}{n}\sum_{t=1}^{n} \|y(t) - \mathbf{x}^*(t)\widehat{\mathbf{w}}(t)\|_2^2 \tag{5.12}$$

$$\sigma_{q_i}^2 = \frac{1}{n-1}\sum_{t=2}^{n} \|\widehat{\mathbf{w}}_i(t) - \widehat{\mathbf{w}}_i(t-1)\|_2^2. \tag{5.13}$$

Alternatively, under the assumption that the state noise covariance satisfies

$$\mathbf{R}_{|\Lambda_0}(n) = r(n)\mathbf{I},$$

both noise disturbances can be estimated adaptively. A smoothed estimate of the state and observation noise can be respectively obtained according to steps 5 and 6 of Table 5.1, where α is a smoothing parameter and $R(x)$ is the ramp function ($R(x) = x$ if $x \geq 0$ and 0 otherwise). These two methods for online estimation of the noise disturbances are due to Jazwinski [74].

5.3.2 EM-RLS filter

The recursive procedure for the determination of the Kalman filter in the special case of the time-varying random walk model Eq. (5.3), resembles the RLS algorithm. In fact, the RLS can be viewed as a special form of Table 5.1 which provides an alternative for the estimation of the noise variances [85, p. 63].

Indeed the RLS results when the weighted prediction error

$$J(\mathbf{w}) = \sum_{i=1}^{n} \beta(n,i)|\varepsilon(i)|^2$$

is minimized. The weighted factor $\beta(n,i)$ (with $0 \leq \beta(n,i) < 1$) is used to discount data and track system changes by allowing recent data to be more heavily weighted. A common choice of $\beta(n,i)$ is the exponentially decaying function $\beta(n,i) = \lambda^{n-i}$.

Table 5.1: The EM-Kalman filter.

EM-Kalman
Initialization : $\mathbf{w}_0 = \bar{\mathbf{w}}_0$, $\mathbf{P}_0 = \delta^{-1}\mathbf{I}$ with δ =const.
For $n := 1, 2, \ldots$ **do**
1: $\mathbf{k}(n) = \dfrac{\mathbf{P}(n-1)\mathbf{x}(n)}{\sigma_v^2 + \mathbf{x}^*(n)\mathbf{P}(n-1)\mathbf{x}(n)}$
2: $\boldsymbol{\psi}(n) = \widehat{\mathbf{w}}(n-1) + \mathbf{k}(n)\varepsilon^*(n)$
3: $\mathbf{P}(n) = \mathbf{P}(n-1) + r(n)\mathbf{I} - \mathbf{k}(n)\mathbf{x}^*(n)\mathbf{P}(n-1)$
4: $\sigma^2_{\varepsilon(n)} = \sigma_v^2 + \mathbf{x}^*(n)\mathbf{P}(n-1)\mathbf{x}(n)$
5: $r(n) = \alpha r(n-1) + (1-\alpha)R\left(\dfrac{\lvert\varepsilon(n)\rvert^2 - \sigma^2_{\varepsilon(n)}}{\mathbf{x}^*(n)\mathbf{x}(n)}\right)$
6: $\sigma_v^2 = \alpha\sigma_v^2 + (1-\alpha)R\big(\lvert\varepsilon(n)\rvert^2 - \mathbf{x}^*(n)\mathbf{P}(n-1)\mathbf{x}(n)\big)$
7: $\widehat{\mathbf{w}}(n) = \operatorname{sgn}\big(\boldsymbol{\psi}(n)\big)\Big[\lvert\boldsymbol{\psi}(n)\rvert - \gamma\operatorname{diag}\big(\mathbf{P}(n-1) + \mathbf{R}(n)\big)\mathbf{1}\Big]_+$
end For

The signum function of a vector is the vector of the signum of its component.

The RLS filter is given by steps 1-3 of Table 5.2 [85, p. 63], with

$$\sigma_v^2 = \lambda, \quad \mathbf{R}(n) = (\lambda^{-1} - 1)\mathbf{P}(n-1). \tag{5.14}$$

The complete algorithm is presented in Table 5.2.

5.3.3 Fast implementations

The RLS algorithm of Table 5.2 admits a fast realization referred to as Fast RLS (FRLS) [32]. FRLS updates the gain with $\mathcal{O}(M)$ operations utilizing the low displacement rank structure of the covariance matrix. More precisely, the Fast RLS reduces the complexity from $\mathcal{O}(M^2)$ to $\mathcal{O}(M)$. This is achieved by exploiting the shift-invariance structure of the covariance matrix [32], [108, Ch. 38]. A similar fast realization of the EM-Kalman algorithm can be developed due to the special

Table 5.2: The EM-RLS filter.

EM-RLS
Initialization : $\mathbf{w}_0 = 0$, $\mathbf{P}_0 = \delta^{-1}\mathbf{I}$ with δ =const.
For $n := 1, 2, \ldots$ **do**
1: $\mathbf{k}(n) = \dfrac{\mathbf{P}(n-1)\mathbf{x}(n)}{\lambda + \mathbf{x}^*(n)\mathbf{P}(n-1)\mathbf{x}(n)}$
2: $\boldsymbol{\psi}_n = \widehat{\mathbf{w}}(n-1) + \mathbf{k}(n)\varepsilon^*(n)$
3: $\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^*(n)\mathbf{P}(n-1)$
4: $\widehat{\mathbf{w}}(n) = \mathrm{sgn}\big(\boldsymbol{\psi}(n)\big)\big[\lvert\boldsymbol{\psi}(n)\rvert - \gamma\lambda^{-1}\mathrm{diag}(\mathbf{P}(n-1))\mathbf{1}\big]_+$
end For

structure of the state covariance matrix. This algorithm is summarized in Table 5.3. Fast implementations tend to suffer from numerical instabilities when implemented in finite precision. A stable version [113] introduces redundancy into the computation of the a priori backward prediction error

$$\tilde{e}_M^b(n) = \lambda E^b(n-1)k^b(n)$$

$$\hat{e}_M^b(n) = x(n-M) + \mathbf{b}_M^*(n)\mathbf{x}(n).$$

The rescue mechanism incorporates the evaluation of $e_M^b(n)$ by a suitable combination of both of these expressions. The difference between $\tilde{e}_M^b(n),\ \hat{e}_M^b(n)$ is fed back, scaled by a certain gain τ

$$e_M^b(n) = \tilde{e}_M^b(n) + \tau[\hat{e}_M^b(n) - \tilde{e}_M^b(n)].$$

The tuning parameter τ is experimentally chosen to ensure numerical stability. In addition, the value of λ, should be within the range $\lambda \in (1 - 1/3M, 1 - 1/10M)$. The fast implementation cannot be used directly for the models given by Eqs. (5.20)-(5.21) below. These models require a multichannel filtering re-formulation [108, p.

Table 5.3: EM-FRLS for sparse adaptive tracking.

Initialization : $\mathbf{w}_0 = 0$, $\mathbf{k}_M(n) = 0$, $\mathbf{a}_M(0) = 0$, $\mathbf{b}_M(0) = 0$, $\bar{a}(0) = 1$, $E^f(0) = \lambda^M E^b(0)$, $E^b(0) = \sigma_x^2$.

For $n := 1, 2, \ldots$ **do**

Forward Predictor

$e_M^f(n) = x(n) + \mathbf{a}_M^*(n-1)\mathbf{x}(n-1)$

$\varepsilon_M^f(n) = e_M^f(n)/\bar{a}(n-1)$

$\mathbf{a}_M(n) = \mathbf{a}_M(n-1) - \mathbf{k}_M(n-1)\varepsilon_M^{f*}(n)$

$E^f(n) = \lambda E^f(n-1) - \varepsilon_M^f(n) e_M^{f*}(n)$

$\bar{a}_{M+1}(n) = \bar{a}(n-1) + \dfrac{|e_M^f(n)|^2}{\lambda E^f(n-1)}$

Kalman Gain vector

$\mathbf{k}_{M+1}(n) = \begin{bmatrix} 0 \\ \mathbf{k}_M(n-1) \end{bmatrix} + \dfrac{e_M^f(n)}{\lambda E^f(n-1)} \begin{bmatrix} 1 \\ \mathbf{a}_M(n-1) \end{bmatrix}$

$\mathbf{k}_{M+1}^*(n) = \begin{bmatrix} \mathbf{k}_M^{\lceil M \rceil}(n) & k^b(n) \end{bmatrix}$

$\mathbf{k}_M(n) = \mathbf{k}_{M+1}^{\lceil M \rceil}(n) - k^b \mathbf{b}_M(n-1)$

Backward Predictor

$e_M^b(n) = \lambda E^b(n-1) k^b(n)$

$\bar{a}(n) = \bar{a}_{M+1}(n) - k^{b*}(n) e_M^b(n)$

$\varepsilon_M^b(n) = e_M^b(n)/\bar{a}(n)$

$\mathbf{b}_M(n) = \mathbf{b}_M(n-1) - \mathbf{k}_M(n)\varepsilon_M^{b*}(n)$

$E^b(n) = \lambda E^b(n-1) - \varepsilon_M^b(n) e_M^{b*}(n)$

Filter update

$e_n = y(n) - \widehat{\mathbf{w}}(n-1)^* \mathbf{x}(n)$

$\varepsilon(n) = e(n)/\bar{a}(n)$

$\boldsymbol{\psi}(n) = \widehat{\mathbf{w}}(n-1) + \mathbf{k}_M(n)\varepsilon^*(n)$

Shrinkage

$\widehat{\mathbf{w}}(n) = \operatorname{sgn}\big(\boldsymbol{\psi}(n)\big)\big[|\boldsymbol{\psi}(n)| - \gamma\lambda^{-1}\operatorname{diag}(\mathbf{P}(n-1))\mathbf{1}\big]_+$

end For

649], [58].

5.3.4 The LMS family

For the purposes of simulations presented in the next section, we discuss the LMS variant developed in [4]. LMS updates a convex function of the prediction error signal

Table 5.4: The EM-LMS family.

EM-LMS
Initialization : $\mathbf{w}_0 = 0$, $0 < \mu < 2\lambda_{\max}^{-1}$
For $n := 1, 2, \ldots$ **do**
1a: $\widehat{\mathbf{w}}(n) = \widehat{\mathbf{w}}(n-1) - \gamma \text{sgn}\widehat{\mathbf{w}}(n-1) + \mu\mathbf{x}(n)\varepsilon^*(n)$
Alternatively step 1a, can be executed as follows
1b: $\widehat{\mathbf{w}}(n) = \widehat{\mathbf{w}}(n-1) - \gamma \dfrac{\text{sgn}\widehat{\mathbf{w}}(n-1)}{1+\epsilon\lvert\widehat{\mathbf{w}}(n-1)\rvert} + \mu\mathbf{x}(n)\varepsilon^*(n)$
end For

$\varepsilon(n)$ plus an ℓ_1-penalty

$$J(\mathbf{w}(n)) = \frac{1}{2}|\varepsilon(n)|^2 + \gamma^{'}\|\mathbf{w}(n)\|_1 = \frac{1}{2}|y(n) - \mathbf{x}^*(n)\mathbf{w}(n)|^2 + \gamma^{'}\|\mathbf{w}(n)\|_1 \qquad (5.15)$$

The regularized cost function of Eq. (5.15) consists of two parts, the instantaneous squared error (of the standard LMS filter) and the ℓ_1 -penalty which has the potential of finding sparse estimates. At each time step the parameter vector is updated using a gradient-descent algorithm that minimizes $J(\mathbf{w}(n))$. The update equation which minimizes Eq. (5.15) is given in step 1a of Table 5.4. The step-size μ controls the rate of convergence of the algorithm to the optimum solution and $\gamma = \mu\gamma^{'}$ [36].

Note the the normalized variant of the above algorithm is a special case of the EM-Kalman filter [85, p. 64] of Table 5.1 by setting:

$$P_0 = \mu\mathbf{I}, \quad \sigma_v^2 = 1, \quad \mathbf{k}(n) = \frac{\mu\mathbf{x}(n)}{1+\mu|\mathbf{x}(n)|}, \quad \mathbf{R}(n) = \mu\mathbf{x}(n)\mathbf{k}^*(n). \qquad (5.16)$$

Motivated by [31], the authors in [36] replace the ℓ_1-norm penalty by the log-sum penalty function. Hence, the new cost function becomes

$$J(\mathbf{w}(n)) = \frac{1}{2}|\varepsilon(n)|^2 + \gamma^{'}\sum_{i=1}^{M}\log(1+|w_i(n)|/\epsilon^{'}). \qquad (5.17)$$

Then, the resulting update equation for this cost function becomes step 1b of Table 5.4, with $\gamma = \mu\gamma^{'}/\epsilon^{'}$ and $\epsilon = 1/\epsilon^{'}$. As a rule of thumb ϵ should be set slightly smaller than the expected non-zero magnitudes of $\mathbf{w}(n)$ [31]. The log-sum penalty function has the potential of being more-sparsity encouraging since it better approximates the non-convex ℓ_0-norm [31]. However computational complexity increases due to the computation of the logarithm at each step.

5.3.5 The re-weighted EM-Kalman filter

The key difference between the ℓ_0 minimization and (P_{ℓ_1}) is that the ℓ_1-norm penalty depends on the magnitudes of the non-zero components of the parameter vector, whereas the ℓ_0-norm penalty does not. As a result, the larger a component of the estimate vector is, the heavier it is penalized by the ℓ_1 penalty. Motivated by the work in [31] to overcome this unfair penalization, we derive a re-weighted version of the EM-Kalman filter, in which at each iteration the following cost function is maximized:

$$\widehat{\boldsymbol{\theta}}(n) = \arg\max_{\boldsymbol{\theta}} \left\{ \mathbb{E}_{p(\mathbf{w}(n)|\mathbf{y}(n-1);\widehat{\boldsymbol{\theta}}(n-1))} \big[\log p(\mathbf{w}(n);\boldsymbol{\theta}) \big] - \gamma \|\mathbf{W}(n)\boldsymbol{\theta}\|_1 \right\} \tag{5.18}$$

where $\mathbf{W}(n)$ is a diagonal weighing matrix with positive diagonal elements $w_1(n)$,. . ., $w_M(n)$. The diagonal elements of $\mathbf{W}(n)$ must be chosen such that they put lower (higher) weight on the smaller (larger) components of the estimate vector. This mechanism yields a better approximation to the ℓ_0 quasi-norm penalization, and thus reduces the estimate bias.

One way to adjust the weighting matrix is to run an adaptation algorithm in parallel to the **E-step** for supplying the required weights [7]. However, this consider-

ably increases the computational complexity of the underlying adaptive algorithm. A widely used choice for the diagonal elements of the matrix is $w_i(n) = 1/(1+\epsilon|\boldsymbol{\psi}_i(n)|)$, for $i = 1, 2, \cdots, M$, where $\epsilon > 0$ is a stability parameter [31]. For this particular choice of the weighting matrix, the re-weighted ℓ_1-norm is identical to the log-sum penalty [31] and thus the **M-step** of Eq. (5.18) leads to the *weighted soft-threholding* function given by

$$\widehat{\boldsymbol{\theta}}_i(n) = \text{sgn}\left(\boldsymbol{\psi}_i(n)\right)\left[|\boldsymbol{\psi}_i(n)| - \frac{\gamma\widehat{\boldsymbol{\Sigma}}_{ii}(n)}{1+\epsilon|\boldsymbol{\psi}_i(n)|}\right]_+. \tag{5.19}$$

5.3.6 Low complexity implementation of the sparse adaptive algorithms

It must be noted that due to the nature of the soft-thresholding step, the estimate $\widehat{\mathbf{w}}(n)$ has many zero entries. This will allow to implement the EM-Kalman, EM-RLS and EM-FRLS algorithms in a low complexity fashion similar to the approach taken in [12]. The idea is to restrict the matrix operations to the instantaneous support of the estimate $\widehat{\mathbf{w}}(n)$.

5.4 Application to nonlinear channel estimation

In what follows, power amplifier nonlinear models are incorporated into the study of two important channels: 1) the satellite link, and 2) the multi-path wireless channel. In both cases, the overall communication channel is represented by baseband time-varying Volterra series.

A finite order passband Volterra system has the form

$$y(i) = \sum_{p=1}^{P} \sum_{\tau_1=0}^{M_p} \cdots \sum_{\tau_p=0}^{M_p} h_p(i, \tau_1, \ldots, \tau_p) \prod_{j=1}^{p} x(i - \tau_j) + v(i) \tag{5.20}$$

where $h_p(.)$ is the Volterra kernel of order p. The disturbance $v(i)$ is assumed to be white Gaussian noise. P is the highest order of nonlinearity while M_p is the p^{th} order system memory. The Volterra model becomes linear when $P = 1$, quadratic when $P = 2$ and cubic when $P = 3$. In bandlimited communications only odd products of the input are considered [21, p. 735]. In such cases only spectral components inside the frequency band of interest are maintained and the baseband Volterra model [21, p. 734] is used

$$y(i) = \sum_{p=0}^{\lfloor \frac{P-1}{2} \rfloor} \sum_{\tau_1=0}^{M_{2p+1}} \cdots \sum_{\tau_{2p+1}=0}^{M_{2p+1}} h_{2p+1}(i, \tau_{1:2p+1}) \prod_{j=1}^{p+1} x(i - \tau_j) \prod_{k=p+2}^{2p+1} x^*(i - \tau_k) + v(i) \tag{5.21}$$

where $h_{2p+1}(i, \tau_{1:2p+1})$ denotes the baseband kernel with $\tau_{1:2p+1} = [\tau_1, \ldots, \tau_{2p+1}]$.

The above equations can be written in the linear regression form of Eq. (5.1) using the Kronecker product notation. Indeed consider the baseband case, let

$$\mathbf{x}_{M_{2p+1}}(i) = [x(i), x(i-1), \cdots, x(i - M_{2p+1})]^T$$

then $\mathbf{x}(i)$ is given by the i-fold Kronecker product

$$\mathbf{x}^{(p+1,p)}_{M_{2p+1}}(i) = [\otimes_{j=1}^{p+1} \mathbf{x}_{M_{2p+1}}(i)] \otimes [\otimes_{k=1}^{p} \mathbf{x}^*_{M_{2p+1}}(i)].$$

The Kronecker product contains all $2p + 1$ order products of the input with p conjugated $p + 1$ unconjugated copies. Likewise $\mathbf{w}(i) = \left[\mathbf{w}_1(i, \cdot), \cdots, \mathbf{w}_{2p+1}(i, \cdot)\right]^T$ is obtained by treating the $(2p + 1)$-dimensional kernel as a M^{2p+1}_{2p+1} column vector. The

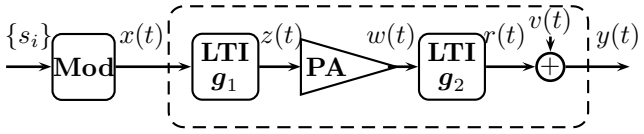


Figure 5.1: Digital satellite link

passband case can be treated in a similar fashion. The regressor vector is now given by

$$\mathbf{x}^T(i) = \left[\mathbf{x}_{M_1}^T(i) \cdots \mathbf{x}_{M_{2p+1}}^{(p+1,p)T}(i)\right]. \tag{5.22}$$

Since the vector $\mathbf{w}(i)$ is sparse, by collecting successive samples , recovery of the unknown Volterra kernels can be accomplished by compressed sensing methods. Recovery of the locations, the magnitudes and the Volterra kernels can be accompanied by the convex program (P_{ℓ_1}).

5.4.1 Volterra representation for satellite links

In satellite digital transmission, both the earth station and the satellite repeater employ power amplifiers. The satellite amplifier operates near saturation due to limited power resources and hence behaves in a nonlinear fashion. The satellite link is composed by a power amplifier sandwiched between two linear filters with impulse responses $\mathbf{g}_1$ and $\mathbf{g}_2$, Fig. 5.1. The LTI filter with impulse response $\mathbf{g}_1$ describes the cascade of all linear operations preceding the power amplifier. Likewise the LTI filter $\mathbf{g}_2$ represents the cascade of all linear devices following the nonlinearity.

An analysis of the above system for static power amplifiers is provided by Benedetto and Biglieri [21, p. 735]. Let us next consider power amplifiers with memory described

by Volterra models. To reduce the computational complexity we shall follow standard practice [71, 80] and confine our study to diagonal Volterra models[2] given by

$$w(t) = \sum_{p=0}^{\lfloor \frac{P-1}{2} \rfloor} \int k_{2p+1}(\mu)|z(t-\mu)|^{2p} z(t-\mu) d\mu \tag{5.23}$$

Note that only odd products of the input appear in (5.23), as a consequence of the bandpass property of the nonlinearity [21, p. 735]. Let us now combine the effects of the linear filters

$$z(t) = \int g_1(\sigma) x(t-\sigma) d\sigma, \quad r(t) = \int g_2(\rho) x(t-\rho) d\rho \tag{5.24}$$

Straightforward calculations lead to the baseband Volterra model

$$r(t) = \sum_{p=0}^{\lfloor \frac{P-1}{2} \rfloor} \int \cdots \int h_{2p+1}(\tau_{1:2p+1}) \times \prod_{i=1}^{p+1} x(t-\tau_i) \prod_{j=p+2}^{2p+1} x^*(t-\tau_j) d\tau_{1:2p+1}. \tag{5.25}$$

The baseband kernel $h_{2p+1}(\tau_{1:2p+1})$ is given by

$$h_{2p+1}(\tau_{1:2p+1}) = \int g_2(\rho) \int k_{2p+1}(\mu) \prod_{i=1}^{p+1} g_1(\tau_i - \mu - \rho) \prod_{j=p+2}^{2p+1} g_1^*(\tau_i - \mu - \rho) d\mu d\rho$$

The Volterra kernels are expressed in the frequency domain as follows:

$$H_{2p+1}(\omega_{1:2p+1}) = G_2 \left(\sum_{l_1=0}^{p} \omega_{2l_1+1} \right) K_{2p+1} \left(\sum_{l_2=0}^{p} \omega_{2l_2+1} \right) \prod_{i=1}^{p+1} G_1(\omega_i) \prod_{j=p+2}^{2p+1} G_1^*(\omega_j)$$

In most cases the filter $\mathbf{g}_1$ performs a specific functionality (for instance pulse shaping) and hence is known. Since we shall deal with channel estimation using known inputs, we may with no loss of generality assume the input signal is the

[2]The analysis of general (non-diagonal) Volterra models is similar.

output of $\mathbf{g}_1$. In this case the Volterra representation from signal $x(t)$ to signal $r(t)$ gets simpler. More precisely we have

$$r(t) = \sum_{p=0}^{\lfloor \frac{P-1}{2} \rfloor} \int h_{2p+1}(\tau_1)|x(t-\tau_1)|^{2p} x(t-\tau_1) d\tau_1. \tag{5.26}$$

where

$$h_{2p+1}(\tau_1) = h_{2p+1}(\tau_{1:2p+1})\delta(\tau_1 - \tau_3)\cdots\delta(\tau_1 - \tau_{2p+1}) = \int g_2(\rho) k_{2p+1}(\tau_1 - \rho) d\rho \tag{5.27}$$

In the frequency domain the Volterra kernels take the form

$$H_{2p+1}(\omega_{2p+1}) = G_2(\omega_{2p+1}) K_{2p+1}(\omega_{2p+1}) \tag{5.28}$$

5.4.2 Multipath channels

In this subsection, fading channels characterized by multipath effects are considered. We shall assume that the modulated signal is amplified by a power amplifier with memory and then transmitted through the wireless medium. The received waveform is the superposition of weighted and delayed versions of the signal resulting from various multipaths plus additive white Gaussian noise.

We shall assume that the different nonzero fading rays arrive at the receiver at different time instances and they vary slowly with time and frequency. Thus the wireless channel becomes a frequency selective channel [16], that is, a Linear Time Invariant system with impulse response

$$g_2(\rho) = \sum_{i=1}^{N_{path}} a_i \delta(\rho - \tau_i) \tag{5.29}$$

where N_{path} is the number of paths, a_i is the attenuation along path i and τ_i is the clustered delay. If we substitute Eq. (5.29) into (5.27) we obtain

$$h_{2p+1}(\tau) = \sum_{i=1}^{N_{path}} a_i k_{2p+1}(\tau - \tau_i) \tag{5.30}$$

The kernels in the frequency domain become

$$H_{2p+1}(\omega_{2p+1}) = K_{2p+1}(\omega_{2p+1}) \sum_{i=1}^{N_{path}} a_i e^{-j\omega_{2p+1}\tau_i}.$$

5.4.3 Sparse Volterra channel representation

The transmission systems described in the previous subsections 5.4.1 and 5.4.2 operate in continuous time. Discrete Volterra forms result when the modulation at the transmitter and the sampling device at the receiver are taken into account. We shall consider memoryless modulation schemes whereby

$$x(t) = \sum_i s_i \delta(t - iT_s). \tag{5.31}$$

The sequence s_i consists of i.i.d (discrete) complex valued random variables and T_s denotes the symbol period. Substituting $x(t)$ from Eq. (5.31) into (5.25) yields a Volterra description of both channels, given by Eq. (5.21). The resulting Volterra kernels are sparse. Indeed, it is well documented in the literature that parsimonious models are highly desirable in the representation of memory PA. In fact it has been experimentally observed [80] that sparse diagonal Volterra models for PA provide enhanced performance in comparison to the full model. The convolutional form of Eq. (5.30) indicates that the sparsity of the $2p+1$ kernel is at most $s_k \times s_m$, where s_k is the sparsity of the PA and s_m is the sparsity of the multipath coefficients. Similar observations hold for the satellite channel.

Table 5.5: Choice of parameters for computer experiments

Exp.	**FRLS** $(\ell_1, w-\ell_1)$	**KF**$(\ell_1, w-\ell_1)$	**LMS** $(\ell_1, \log\ )$	**RLS** (ℓ_1)
5.1[1]	3×10^{-4},2×10^{-3}	–	3×10^{-4},7×10^{-4}	–
5.2[2]	–	–	3×10^{-4},1×10^{-3}	5×10^{-4}
5.3[3]	–	7×10^{-3},2×10^{-3}	4×10^{-4},2×10^{-3}	6×10^{-4}

5.5 Simulations

In this section we evaluate the performance of the proposed algorithms given in section 5.3 through computer simulations. Experiments were conducted on both linear and nonlinear channel setups.

In all experiments the output sequence is disturbed by additive white Gaussian noise, with a Signal to Noise Ratio (SNR) of SNR $= 10dB$. SNR is the ratio of the coefficient vector power to the noise power corrupting the output signal ($\mathbb{E}[\|\hat{\mathbf{w}}\|_2^2]/\sigma_v^2$). The Normalized Mean Square Error, defined as

$$\text{NMSE} = 10\log_{10}\left(\mathbb{E}[\|\hat{\mathbf{w}}-\mathbf{w}\|_2^2]/\mathbb{E}[\|\mathbf{w}\|_2^2]\right),$$

was used to assess performance. The choice of the parameters λ, γ that were used to compare performance of the sparse algorithms for each experiment are summarized in Tables 5.5.

5.5.1 Tracking a Rayleigh fading sparse channel

In this experiment we compare the performance of the three computationally lighter algorithms, *i.e.* the EM-FRLS, the wEM-FRLS and the sparse LMS filter on a linear channel. We consider a wireless channel with three Rayleigh fading rays; all

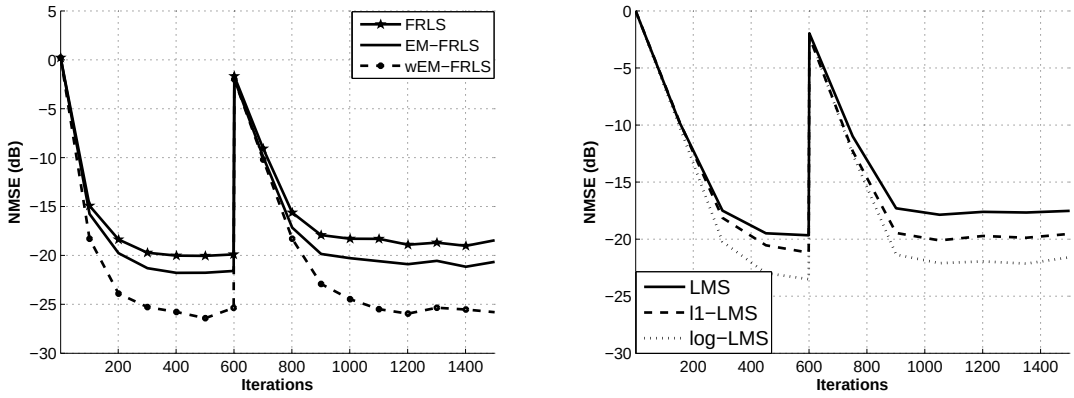
rays are assumed to fade at the same Doppler frequency of $f_D = 10$Hz [108, p. 325-326]. The channel impulse response sequence consists of four non-zero Rayleigh fading rays at positions 4, 10, 16 and 24. In other words, we are assuming a channel length of $M = 30$ taps with only four active Rayleigh fading rays. The auto-correlation function of the non-zero channel taps is modeled as a zeroth-order Bessel function

$$r(\tau) = J_0(2\pi f_D T_s)$$

where T_s is the sampling period and f_D is the maximum Doppler frequency of the Rayleigh fading channel.

All input symbols are drawn from a BPSK constellation. Figs. 5.2(a) and 5.2(b) show the NMSE learning curve for various adaptive filters, which result from averaging over 20 independent computer runs. As it is readily apparent from the figures, EM-FRLS and the re-weighted EM-FRLS (wEM-FRLS) achieve a performance gain of around $4dB$ and $8dB$ respectively, over the conventional FRLS algorithm. Moreover, the convergence rate and the performance of the FRLS family is better than the LMS family.

To demonstrate the support tracking performance of the proposed algorithms, after 600 iterations we set the second and third non-zero fading ray to zero while a new fading ray appears at position 11. We note from Figs. 5.2(a) and 5.2(b), that the conventional algorithms have a slower support tracking behavior than the sparse variants.



(a) FRLS Family (b) LMS Family

Figure 5.2: NMSE between the EM-FRLS, wEM-FRLS and sparse LMS for linear system identification (support change occurs at $n = 600$)

5.5.2 Estimation of a sparse baseband Volterra channel

In the second channel set up a third-order baseband Volterra model is considered. It is given by

$$y(n) = (0.4968 - j0.6707)x(n-3) + (1.3336 + j0.9303)x(n-5)$$
$$+ (1.2745 + j0.2965)x^2(n-1)x^*(n-4) - (0.4794 + j1.3298)x^2(n-5)x^*(n-1).$$

The above channel is excited by an OFDM signal, which can be approximated by a zero-mean, stationary complex normal random process, $\mathcal{CN}(0, 1/4)$. This approximation relies on the Central Limit Theorem and the fact that an OFDM signal consists of a sum of a large number of independent, identically distributed signals [15, p. 70-71]. The memory of the linear and cubic part is $M_1 = M_3 = 7$. The algorithms were run for 10 Monte carlo runs and the performance of the sparse algorithms is illustrated in Fig. 5.3. The EM-RLS algorithms introduces a performance gain of $8dB$ over the conventional RLS. Fig. 5.3 illustrates that the LMS family has very

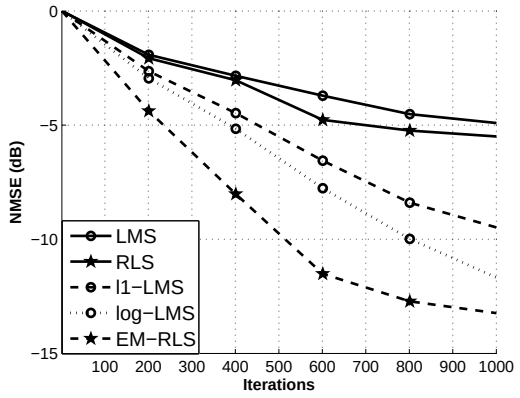


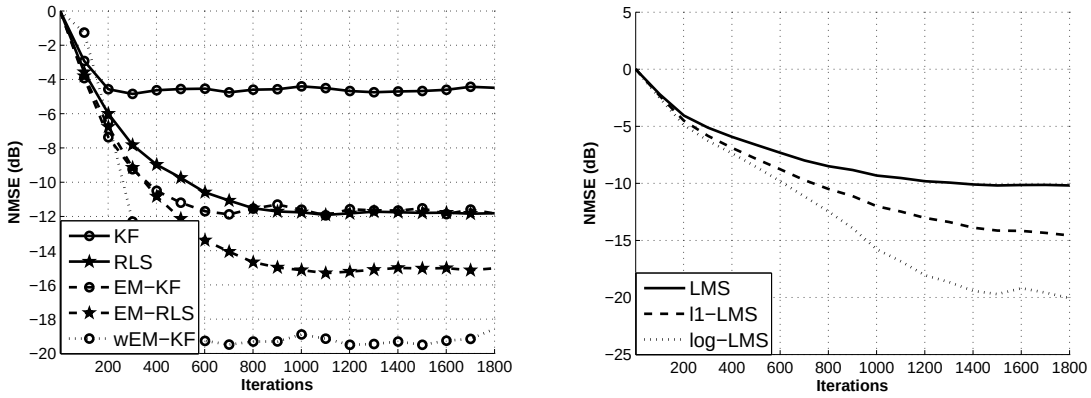
Figure 5.3: NMSE between two sparse algorithms on a cubic baseband Volterra

slow convergence.

5.5.3 Tracking a Rayleigh fading sparse Hammerstein channel

In this experiment the multipath channel setup of Eq. (5.26) is considered. This model is usually encountered in GSM applications. The base station (BSS) communicates with the mobile terminal through the wireless channel. A nonlinear power amplifier at the BSS amplifies the transmitted signal so that it reaches the remote terminal.

The wireless channel taps for the linear and cubic part were generated by sparse Rayleigh fading rays [108, p. 325-326]. All rays are assumed to fade at the same Doppler frequency of $f_D = 80Hz$ with sampling period $T_s = 0.8\mu s$. The linear and the cubic part have equal memory size $M_1 = M_3 = 50$ and the support signal consists of 2 randomly selected elements for each part. The input signal is drawn from a



(a) KF & RLS Family (b) LMS Family

Figure 5.4: NMSE between the three sparse algorithms on a multipath Hammerstein channel.

complex Gaussian distribution $\mathcal{CN}(0, 1/4)$. We observe in Fig. 5.4 that the EM-Kalman, re-weighted EM-Kalman (wEM-Kalman) and EM-RLS algorithms provide gains of $7dB$, $12dB$ and $4dB$ respectively, over the corresponding conventional non-sparse algorithms. Moreover, the proposed algorithms perform almost similar NMSE performance with the LMS family but the convergence speed is significantly faster.

Chapter 6

An Adaptive Greedy Algorithm for Estimation of Sparse Signals

6.1 Introduction

Many signal processing applications [16, 79, 109] require adaptive estimation with minimal complexity and small memory requirements. Existing approaches to sparse adaptive estimation, such as those discussed in Chapters 4 and 5, use the ℓ_1-minimization technique in order to improve the performance of conventional algorithms.

As mentioned in Chapter 1, two major algorithmic approaches to compressive sensing are ℓ_1-minimization (basis pursuit) and greedy algorithms (matching pursuit). Basis pursuit methods solve a convex minimization problem, which replaces the ℓ_0 quasi-norm by the ℓ_1 norm. The convex minimization problem can be solved using linear programming methods, and is thus executed in polynomial time [35]. Greedy algorithms, on the other hand, iteratively compute the support set of the signal and

construct an approximation to it, until a halting condition is met [41,44,50,95,96,100]. Both of the above approaches pose their own advantages and disadvantages. ℓ_1-minimization methods provide theoretical performance guarantees, but they lack the speed of greedy techniques. Recently developed greedy algorithms, such as those developed in [41, 95, 96], deliver some of the same guarantee as ℓ_1-minimization approaches with less computational cost and storage.

In contrast to the algorithms developed in Chapters 4 and 5 on adaptive sparse identification, this chapter focuses on the greedy viewpoint. Greedy algorithms in their ordinary mode of operation, have an inherent batch mode, and hence are not suitable for time-varying environments. We establish a conversion procedure that turns greedy algorithms into adaptive schemes for sparse system identification [92,93]. In particular, a Sparse Adaptive Orthogonal Matching Pursuit (SpAdOMP) algorithm of linear complexity is developed, based on existing greedy algorithms [41, 95], that provide optimal performance guarantees. Also, the steady-state Mean Square Error (MSE) of the SpAdOMP algorithm is studied analytically. The developed algorithm is used to estimate ARMA and Nonlinear ARMA channels. It is shown that channel inversion for these channels, maintains sparsity and that it is equivalent to channel estimation. Computer simulations reveal that the proposed algorithm outperforms most existing adaptive algorithms for sparse channel estimation.

The chapter is structured as follows. The problem formulation and literature review are addressed in section 6.2. Section 6.3 describes the established algorithm, the steady-state error analysis and applications to nonlinear communication channels. Computer simulations are presented in section 6.4.

6.2 Greedy methods and the CoSaMP algorithm

Consider the noisy representation of a vector $\mathbf{y}(n) = [y_1, \cdots, y_n]^T$ in terms of a basis arranged in the columns of a matrix $\mathbf{\Phi}(n)$ at time n

$$\mathbf{y}(n) = \mathbf{\Phi}(n)\mathbf{c} + \boldsymbol{\eta}(n) \tag{6.1}$$

where $\mathbf{c}$ is the parameter vector, $\mathbf{\Phi}(n) = [\boldsymbol{\phi}(1), \ldots, \boldsymbol{\phi}(n)]^T$ and $\boldsymbol{\eta}(n) = [\eta_1, \ldots, \eta_n]^T$ is the additive noise. The measurement matrix $\mathbf{\Phi}(n) \in \mathbb{C}^{n \times N}$ is often referred to as *dictionary* and the parameter vector $\mathbf{c}$ is assumed to be sparse, *i.e.*, $\|\mathbf{c}\|_0 \ll N$, where $\|\cdot\|_0 = |\mathsf{supp}(\cdot)|$ is the ℓ_0 quasi-norm. We will call the parameter vector s-sparse when it contains at most s non-zero entries.

Remark. Recall that in Chapter 4, we used the notation $\mathbf{X}(n)$ and $\mathbf{w}(n)$ for the measurement matrix and the parameter vector, respectively. Here, we adopt the notation of $\mathbf{\Phi}(n)$ and $\mathbf{c}(n)$ instead, to be consistent with the notation used in the CoSaMP and Subspace Pursuit algorithms, for the clarity of presentation.

Recovery of the unknown parameter vector $\mathbf{c}$ can be pursued by finding the sparsest estimate of $\mathbf{c}$ which satisfies the ℓ_2 norm error tolerance δ

$$\min_{\mathbf{c}} \|\mathbf{c}\|_0 \quad \text{subject to} \quad \|\mathbf{y}(n) - \mathbf{\Phi}(n)\mathbf{c}\|_2 \leq \delta. \tag{P_{ℓ_0}}$$

Convex relaxation methods cope with the intractability of the above formulation by approximating the ℓ_0 quasi-norm by the convex ℓ_1 norm. The set of resulting techniques is referred to as ℓ_1-minimization. The ℓ_2 constraint can be interpreted as a noise removal mechanism when $\delta \geq \|\boldsymbol{\eta}(n)\|_2$. The ℓ_1-minimization approach is a convex optimization problem and can be solved by linear programming methods [29, 35], projected gradient methods [56] and iterative thresholding [43].

Table 6.1: SpAdOMP Algorithm

Algorithm description			Complexity
$\mathbf{c}(0) = 0, \mathbf{w}(0) = 0, \mathbf{p}(0) = 0$		{Initiliazation}	
$v(0) = y(0)$		{Initial residual}	
$0 < \lambda \leq 1$		{Forgetting factor}	
$0 < \mu < 2\lambda_{\max}^{-1}$		{Step size}	
For $n := 1, 2, \ldots$ **do**			
1:	$\mathbf{p}(n) = \lambda \mathbf{p}(n-1) + \boldsymbol{\phi}^*(n-1)v(n-1)$	{Form signal proxy}	N
2:	$\Omega = \mathsf{supp}(\mathbf{p}_{2s}(n))$	{Identify large components}	N
3:	$\Lambda = \Omega \cup \mathsf{supp}(\mathbf{c}(n-1))$	{Merge supports}	s
4:	$\varepsilon(n) = y(n) - \boldsymbol{\phi}_{\vert\Lambda}^T(n)\mathbf{w}_{\vert\Lambda}(n-1)$	{Prediction error}	s
5:	$\mathbf{w}_{\vert\Lambda}(n) = \mathbf{w}_{\vert\Lambda}(n-1) + \mu\boldsymbol{\phi}_{\vert\Lambda}^*(n)\varepsilon(n)$	{LMS iteration}	s
6:	$\Lambda_s = \max(\vert\mathbf{w}_{\vert\Lambda}(n)\vert, s)$	{Obtain the pruned support}	s
7:	$\mathbf{c}_{\vert\Lambda_s}(n) = \mathbf{w}_{\vert\Lambda_s}(n), \quad \mathbf{c}_{\vert\Lambda_s^c}(n) = \mathbf{0}$	{Prune the LMS estimates}	
8:	$v(n) = y(n) - \boldsymbol{\phi}^T(n)\mathbf{c}(n)$	{Update error residual}	s
end For			$\mathcal{O}(N)$

The exact conditions for retrieving the sparse vector rely either on the coherence of the measurement matrix [118] or on the Restricted Isometry Property (RIP) [29]. A measurement matrix $\boldsymbol{\Phi}(n)$ satisfies the Restricted Isometry Property for $\delta_s(n) \in (0, 1)$ if we have

$$\big(1 - \delta_s(n)\big)\|\mathbf{c}\|_2^2 \leq \|\boldsymbol{\Phi}(n)\mathbf{c}\|_2^2 \leq \big(1 + \delta_s(n)\big)\|\mathbf{c}\|_2^2 \tag{6.2}$$

for all s-sparse $\mathbf{c}$. When $\delta_s(n)$ is small, the restricted isometry property implies that the set of columns of $\boldsymbol{\Phi}(n)$ approximately form an orthonormal system.

6.2.1 The CoSaMP greedy algorithm

Greedy algorithms provide an alternative approach to ℓ_1-minimization. For the recovery of a sparse signal in the presence of noise, greedy algorithms iteratively improve the current estimate for the parameter vector $\mathbf{c}$ by modifying one or more parameters until a halting condition is met. The basic principle behind greedy algorithms is to iteratively find the support set of the sparse vector and reconstruct the signal using the restricted support Least Squares (LS) estimate. The computational complexity of these algorithms depends on the number of iterations required to find the correct support set. One of the earliest algorithms proposed for sparse signal recovery is the Orthogonal Matching Pursuit (OMP) [44,100,118]. At each iteration, OMP finds the column of $\mathbf{\Phi}(n)$ most correlated with the residual, $\mathbf{v}(n) = \mathbf{y}(n) - \mathbf{\Phi}(n)\hat{\mathbf{c}}$, using the proxy signal $\mathbf{p}(n) = \mathbf{\Phi}^*(n)\mathbf{v}(n)$ (where $\mathbf{A}^*(n)$ denotes the conjugate transpose of the matrix $\mathbf{A}(n) \in \mathbb{C}^{n\times N}$), and adds it to the support set. Then, it solves the following least squares problem:

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{z}} \|\mathbf{y}(n) - \mathbf{\Phi}(n)\mathbf{z}\|_2$$

and finally updates the residual by removing the contribution of the latter column. By repeating this procedure a total of s times, the support set of $\mathbf{c}$ is recovered. Although OMP is quite fast, it is unknown whether it succeeds on noisy measurements.

An alternative algorithm, called Stagewise OMP (StOMP), was proposed in [50]. Unlike OMP, it selects all components of the proxy signal whose values are above a certain threshold. Due to the multiple selection step, StOMP achieves better runtime than OMP. Parameter tuning in StOMP might be difficult and there are rigorous asymptotic results available.

A more sophisticated algorithm has been recently developed by Needell and Vershynin, and it is known as Regularized OMP (ROMP) [96]. ROMP chooses the s largest components of the proxy signal, followed by a regularization step, to ensure that not too many incorrect components are selected. For a measurement matrix $\mathbf{\Phi}(n)$ with RIP constant $\delta_{2s} = 0.03/\sqrt{\log s}$, ROMP provides uniform and stable recovery results. The recovery bounds obtained in [96] are optimal up to a logarithmic factor. Tighter recovery bounds which avoid the presence of the logarithmic factor are obtained by Needell and Tropp via the Compressed Sampling Matching Pursuit algorithm (CoSaMP) [95]. CoSaMP provides tighter recovery bounds than ROMP optimal up to a constant factor (which is a function of the RIP constants). An algorithm similar to the CoSaMP, was presented by Dai and Milenkovic and is known as Subspace Pursuit (SP) [41].

As with most greedy algorithms, CoSaMP takes advantage of the measurement matrix $\mathbf{\Phi}(n)$ which is approximately orthonormal ($\mathbf{\Phi}^*(n)\mathbf{\Phi}(n)$ is close to the identity). Hence, the largest components of the signal proxy $\mathbf{p}(n) = \mathbf{\Phi}^*(n)\mathbf{\Phi}(n)\mathbf{c}$ is most likely to correspond to the non-zero entries of $\mathbf{c}$. Next, the algorithm adds the largest components of the signal proxy to the running support set and performs least squares to get an estimate for the signal. Finally, it prunes the least square estimation and updates the error residual. The main ingredients of the CoSaMP algorithm are outlined below:

1. *Identification* of the largest $2s$ components of the proxy signal

2. *Support Merger*, which forms the union of the set of newly identified components with the set of indices corresponding to the s largest components of the least

square estimate obtained in the previous iteration

3. *Estimation* via least squares on the merged set of components

4. *Pruning*, which restricts the LS estimate to its s largest components

5. *Sample update*, which updates the error residual.

The above steps are repeated until a halting criterion is met. The main difference between CoSaMP and SP is in the identification step where the SP algorithm chooses the s largest components.

In a time-varying environment, the estimates must be updated adaptively to take into consideration system variations. In such cases, the use of existing greedy algorithms on a measurement block requires that the system remain constant within that block. Moreover, the cost of repetitively applying a greedy algorithm after a new block arrives becomes enormous. Adaptive algorithms, on the other hand, allow online operation. Therefore, our primary goal is to convert existing greedy algorithms into an adaptive mode, while maintaining their superior performance gains. We demonstrate below that the conversion is feasible with linear complexity. We focus our analysis on CoSaMP/SP due to their superior performance, but similar ideas are applicable to other greedy algorithms as well.

6.3 Sparse Adaptive Orthogonal Matching Pursuit Algorithm

This section starts by converting CoSaMP and SP algorithms [41, 95] into an adaptive scheme. The derived algorithm is then used to estimate sparse Nonlinear ARMA channels.

The proposed algorithm relies on three modifications to the CoSaMP/SP structure: the proxy identification, estimation and error residual update. The error residual is now evaluated by

$$v(n) = y(n) - \boldsymbol{\phi}^T(n)\mathbf{c}(n). \tag{6.3}$$

The above formula involves the current sample only, in contrast to the CoSaMP/SP scheme which requires all the previous samples. Eq. (6.3) requires s complex multiplications, whereas the cost of the sample update in the CoSaMP/SP is sn multiplications. A new proxy signal that is more suitable for the adaptive mode, can be defined as:

$$\mathbf{p}(n) = \sum_{i=1}^{n-1} \lambda^{n-1-i} \boldsymbol{\phi}^*(i) v(i)$$

and is updated by

$$\mathbf{p}(n) = \lambda \mathbf{p}(n-1) + \boldsymbol{\phi}^*(n-1) v(n-1)$$

where the forgetting factor $\lambda \in (0, 1]$ is incorporated in order to give less weight in the past and more weight to recent data. This way the derived algorithm is capable of capturing variations on the support set of the parameter vector $\mathbf{c}$. In the case of

a time-invariant environment, λ should be set to 1. The addition of the forgetting factor mechanism requires redefining the Restricted Isometry Property as follows:

Definition 6.3.1. *A measurement matrix $\mathbf{\Phi}(n)$ satisfies the Exponentially-weighted Restricted Isometry Property (ERIP) for $\lambda \in (0,1]$ and $\delta_s(\lambda, n) \in (0,1)$, if we have*

$$\big(1-\delta_s(\lambda,n)\big)\|\mathbf{c}\|_2^2 \leq \|\mathbf{D}^{1/2}(n)\mathbf{\Phi}(n)\mathbf{c}\|_2^2 \leq \big(1+\delta_s(\lambda,n)\big)\|\mathbf{c}\|_2^2 \tag{6.4}$$

where $\mathbf{D}(n) := diag(1, \lambda, \cdots, \lambda^{n-1})$.

The last modification attacks the estimation step. The estimate $\mathbf{w}(n)$ is updated by standard adaptive algorithms such as the LMS and RLS [65]. LMS is one of the most widely used algorithm in adaptive filtering due to its simplicity, robustness and low complexity. On the other hand, the RLS algorithm is an order of magnitude costlier but significantly improves the convergence speed of LMS. The LMS algorithm replaces the exact signal statistics by approximations, whereas RLS updates the inverse covariance matrix. The update rule for RLS cannot be directly restricted to the index support set Λ. Hence, a more sophisticated mechanism is required like the one proposed in [12]. For reasons of simplicity and complexity we focus on the LMS algorithm. At each iteration the current regressor $\boldsymbol{\phi}(n)$ and the previous estimate $\mathbf{w}(n-1)$ are restricted to the instantaneous support originated from the support merging step.

The resulting algorithm, namely the Sparse Adaptive Orthogonal Matching Pursuit (SpAdOMP), is presented in Table 6.1. Note that $\boldsymbol{\phi}_{|\Lambda}$ and $\mathbf{w}_{|\Lambda}$ denote the subvectors corresponding to the index set Λ, $\max(|a|, s)$ returns s indices of the largest elements of a and Λ^c represents the complement of set Λ. An important point to note

about step 5 of Table 6.1 is that, although it is simple to implement, it is difficult to choose the step-size parameter μ which assures convergence. The Normalized LMS (NLMS) update addresses this issue by scaling with the input power

$$\mathbf{w}_{|\Lambda}(n) = \mathbf{w}_{|\Lambda}(n-1) + \frac{\mu}{\epsilon + \|\boldsymbol{\phi}_{|\Lambda}(n)\|^2}\boldsymbol{\phi}^*_{|\Lambda}(n)\varepsilon(n)$$

where $0 < \mu < 2$ and ϵ is a small positive constant (to avoid division by small numbers for stability purposes). NLMS may be viewed as an LMS with time-varying step-size. This observation justifies the superior tracking ability of NLMS with respect to LMS in non-stationary environments.

6.3.1 Compressed Sensing Matrices satisfying the ERIP

As discussed earlier in Chapter 3, one can use the methodology of [63] to establish the ERIP for measurement matrices with Toeplitz structure. For instance, consider the i.i.d. Gaussian input sequences, where $\phi_i(n)$ is distributed according to $\mathcal{N}(0, \nu^2)$. Then, the matrix $\boldsymbol{\Phi}(n)$ has a random Toeplitz structure. As shown in Chapter 3, if $n_\lambda \nu^2 = 1$, with $n_\lambda = \frac{1-\lambda^{n+1}}{1-\lambda}$, and n large enough so that $n_\lambda \approx \frac{1}{1-\lambda}$, the eigen-values of $\boldsymbol{\Psi}_\Lambda(n) := \boldsymbol{\Phi}^{*T}_\Lambda(n)\mathbf{D}(n)\boldsymbol{\Phi}_\Lambda(n)$ with $|\Lambda| = L$, lie in the interval $[1-\delta, 1+\delta]$ with probability exceeding

$$1 - 3N^2 \exp\left(-\frac{n_\lambda \delta^2}{54 s^2}\right). \tag{6.5}$$

In particular, for any $c < \frac{\delta^2}{54}$, by choosing

$$\lambda \geq 1 - \frac{\delta^2 - 54c}{162 s^2 \log N}$$

the exponent in the above expression goes to 0 at a polynomial rate. This analysis can be extended to other classes on input sequences, such as bounded i.i.d. random

sequences [63]. Studying the singular value distribution of exponentially weighted random matrices is in general a hard problem. One such result is discussed in Chapter 3: if the $n \times N$ matrix $\mathbf{\Phi}(n)$ has i.i.d. elements from a random Gaussian vector process distributed according to $\mathcal{N}(0, \nu^2)$ in the limit of $N \to \infty$ and $\lambda \to 1$, with $\beta := s/N < 1$ and $Q := 1/(s(1-\lambda))$ fixed, and $n \to \infty$, the eigenvalues of the matrix $(1-\lambda)\mathbf{\Psi}_\Lambda(n)$ (denoted as x) are distributed according to the density

$$\rho(x) = \frac{Qv}{\pi} \tag{6.6}$$

where v is the solution to the non-algebraic equation

$$\frac{x}{\sigma_\phi^2} - \frac{vx}{\tan(vx)} + \log(v\sigma_\phi^2) - \log\sin(vx) - \frac{1}{Q} = 0. \tag{6.7}$$

Numerical results in [98] show that the density $\rho(x)$ is very similar to the Marchenko-Pastur law, corresponding to the eigen-value distribution of Wishart ensembles. The above example reveals that there is a close connection between the RIP and ERIP conditions (by interpreting $1/\log(1/\lambda)$ as the effective row dimension [98]). The RIP constant of Gaussian measurement matrices has been extensively studied by Blanchard et al. [22]. The above parallelism suggests that one might be able to extend such results regarding the RIP of random measurement matrices to those satisfying ERIP. However, study of the non-asymptotic properties of the eigen-value distribution of exponentially weighted matrices seems to offer more difficulty than their non-weighted counterparts.

6.3.2 Steady-State MSE of SpAdOMP

The following Theorem establishes the steady-state MSE performance of the SpAdOMP algorithm:

Theorem 6.3.2. *(SpAdOMP). Suppose that the input sequence $\boldsymbol{\phi}(n)$ is stationary, i.e., its covariance matrix $\mathbf{R}(n) := \mathbb{E}\{\boldsymbol{\phi}(n)\boldsymbol{\phi}^{*T}(n)\} = \mathbf{R}$ is independent of n. Moreover, assume that $\mathbf{R}$ is non-singular. Finally, suppose that for n large enough, the ERIP constants $\delta_s(\lambda, n)$, $\delta_{2s}(\lambda, n)$, $\cdots$, $\delta_{4s}(\lambda, n)$ exist. Then, the SpAdOMP algorithm, for large n, produces a s-sparse approximation $\mathbf{c}(n)$ to the parameter vector $\mathbf{c}$ that satisfies the following steady-state bound:*

$$\begin{aligned} \epsilon_1(n) &:= \|\mathbf{c} - \mathbf{c}(n)\|_2 \\ &\lesssim C_1(n)\left\|\mathbf{D}^{1/2}(n)\boldsymbol{\eta}(n)\right\|_2 + C_2(n)\left\|\boldsymbol{\phi}_{|\Lambda(n)}(n)\right\|_2 |e_o(n)| \end{aligned} \tag{6.8}$$

where $e_o(n)$ is the estimation error of the optimum Wiener filter, and $C_1(n)$ and $C_2(n)$ are constants independent of $\mathbf{c}$ (which are explicitly given in the Appendix) and are functions of $\lambda_M > 0$ (the minimum eigenvalue of $\mathbf{R}$), the ERIP constants $\delta_s(\lambda, n)$, $\delta_{2s}(\lambda, n)$, $\cdots$, $\delta_{4s}(\lambda, n)$ and the step size μ. The approximation in the above inequality is in the sense of the Direct-averaging technique [65] employed in simplifying the LMS iteration.

The proof is supplied in the Appendix. The above bound can be further simplified if one considers the normalization $\|\boldsymbol{\phi}(n)\|_2^2 = 1$ for all n. Such a normalization is implicitly assumed for the above example on the i.i.d. Gaussian measurement matrix as $n, N \to \infty$ with $\sigma_\phi^2 = 1/N$. In this case, $\left\|\boldsymbol{\phi}_{|\Lambda(n)}(n)\right\|_2 \leq 1$ and thus the second term of the error bound simplifies to $C_2(n)|e_o(n)|$. Note that for large values of n,

the isometry constants can be controlled. As shown in the example above, for a suitably random input sequence (*e.g.*, i.i.d. Gaussian input) and for n large enough, the restricted isometry constants can be sufficiently small. For example, if for n large enough, $\delta_{4s}(\lambda, n) \leq 0.01$ and $\mu\lambda_M = 0.75$, then $C_1(n) \approx 38.6$ and $C_2(n) \approx 7.7$. The corresponding coefficient for the CoSaMP algorithm will be $C_1(n) \approx 6.1$. Hence, the parameters $C_1(n)$ and $C_2(n)$ can be well controlled by feeding enough number of measurements to the SpAdOMP algorithm.

The first term on the right hand side of the Eq. (6.8) is analogous to the steady-state error of the CoSaMP/SP algorithm, corresponding to a batch of data of size n. The second term is the steady-state error induced by performing a single LMS iteration, instead of using the LS estimate. This error term does not exist in the error expression of the CoSaMP/SP algorithm. However, this excess MSE error can be compromised by the significant complexity reduction incurred by removing the LS estimate stage. Note that the promising support tracking behavior of the CoSaMP/SP algorithm is inherited by the LMS iteration, where only the sub-vector of $\boldsymbol{\phi}(n)$ corresponding to $\Lambda(n)$ and $\mathbf{w}_{|\Lambda(n)}$ participate in the LMS iteration, and hence the error term. In other words, the SpAdOMP enjoys the low complexity virtue of LMS, as well as the support detection superiority of the CoSaMP/SP. Indeed, this observation is evident in the simulation results, where the MSE curve of SpAdOMP is shifted from that of LMS towards that of the genie-aided LS estimate (See Section IV).

6.3.3 Sparse NARMA identification

The nonlinear model that we will be concerned with, constitutes a generalization of the class of linear ARMA models [76] and is known as Nonlinear AutoRegressive Moving Average (NARMA) [34]. The output of NARMA models depends on past and present values of the input as well as the output

$$y_i = f(y_{i-1}, \ldots, y_{i-M_y}, x_i, \ldots, x_{i-M_x}) + \eta_i \tag{6.9}$$

where y_i, x_i and η_i are the system output, input and noise, respectively; M_y, M_x denote the output and input memory orders; η_i is Gaussian and independent of x_i; and $f(\cdot)$ is a sparse polynomial function in several variables with degree of nonlinearity p. Known linearization criteria [76] provide sufficient conditions for the Bounded Input Bounded Output stability of (6.9).

Using Kronecker products, we write Eq. (6.9) as a linear regression model

$$y_i = \boldsymbol{\phi}^T(i)\mathbf{c} + \eta_i \tag{6.10}$$

where

$$\boldsymbol{\phi}^T(i) = \left[\begin{array}{ccc} \boldsymbol{\phi}_y^T(i) & \boldsymbol{\phi}_x^T(i) & \boldsymbol{\phi}_{yx}^T(i) \end{array} \right]$$

$\mathbf{y}_i = \left[y_{i-1}, \cdots, y_{i-M_y}\right]^T$ and $\mathbf{x}_i = \left[x_i, \cdots, x_{i-M_x}\right]^T$. Consider the pth order Kronecker powers $\mathbf{y}_i^{(p)} = \mathbf{y}_i^{\otimes p}$ and $\mathbf{x}_i^{(p)} = \mathbf{x}_i^{\otimes p}$. Then, the output and input regressor vectors are respectively given by $\boldsymbol{\phi}_y^T(i) = [\mathbf{y}_i^{(1)}, \cdots, \mathbf{y}_i^{(p)}]$ and $\boldsymbol{\phi}_x^T(i) = [\mathbf{x}_i^{(1)}, \cdots, \mathbf{x}_i^{(p)}]$. $\boldsymbol{\phi}_{yx}^T(i)$ denotes all possible Kronecker product combinations of $\mathbf{y}_i$ and $\mathbf{x}_i$ of degree up to p. The components of $\mathbf{c} = [\ \mathbf{c}_y^T \quad \mathbf{c}_x^T \quad \mathbf{c}_{yx}^T]^T$ correspond to the coefficients of the

polynomial $f(\cdot)$. Hence, if we collect n successive observations, recovery of the sparsest parameter vector can be accomplished by solving the mathematical program (P_{ℓ_0}).

It must be noted that in NARMA models, the input sequence is non-linearly related to the measurement matrix $\mathbf{\Phi}(n)$ through the multi-fold Kronecker product procedure. Thus, the effective measurement matrix generated by an i.i.d. input sequence, will not necessarily maintain the i.i.d. structure. Nevertheless, in case of linear models, by invoking the frequently adopted *independence assumption* [65], the i.i.d. property of the input sequence is carried over to the corresponding measurement matrix, and thus one might be able to guarantee analytically-provable controlled ERIP constants for the measurement matrix (as in the example of Section 6.3.1). Although we have not mathematically established any results regarding the isometry of such structured matrices, simulation results reveal that input sequences which give rise to measurement matrices satisfying the ERIP in linear models (e.g., i.i.d. Gaussian), also perform well in conjunction with non-linear models (See Section IV). Nevertheless, the problem of designing input sequences, with mathematical guarantees on the ERIP of the corresponding measurement matrices in the non-linear models, is of interest and remains open.

6.3.4 Equalization/Predistortion in nonlinear communication channels

Nonlinearities in communication channels are caused by Power Amplifiers (PA) operating near saturation [21] and are addressed by channel inversion. Right inverses are called *predistorters* and are placed at the transmitter side; left inverses are termed

equalizers and are part of the receiver. Predistorters are the preferred solution in single transceiver multiple receiver systems, such as a base station and multiple GSM receivers.

Channel inversion is conveniently effected when Eq. (6.9) is restricted to

$$y_i = b_0 x_i + f(y_{i-1}, \ldots, y_{i-M_y}, x_{i-1}, \ldots, x_{i-M_x}) + \eta_i. \tag{6.11}$$

In the above equation the present input sample enters linearly. If x_i entered polynomially, inversion would require finding the roots of a polynomial which does not always result in a unique solution and is computationally expensive. The inverse of Eq. (6.11) is given by

$$\begin{aligned} x_i = b_0^{-1} \left[y_i - f(y_{i-1}, \ldots, y_{i-M_y}, x_{i-1}, \ldots, x_{i-M_x}) - \eta_i \right], \\ \text{iff } b_0 \neq 0. \end{aligned} \tag{6.12}$$

Note that modulo the scaling by b_0 correction, the system and its inverse are generated by the same function. Hence, estimation of the direct process is equivalent to the estimation of the reverse process.

6.4 Experimental results

In this section we compare through computer simulations the performance of existing algorithms and the proposed algorithm. Experiments were conducted on both linear and nonlinear channel setups. In all experiments the output sequence is disturbed by additive white Gaussian noise for various SNR levels ranging from 5 to $26dB$. SNR is the ratio of the noiseless channel output power to the noise power

corrupting the output signal (σ_y^2/σ_η^2). The Normalized Mean Square Error, defined as

$$\mathbb{E}[\|\mathbf{c}(n) - \mathbf{c}\|_2^2]/\mathbb{E}[\|\mathbf{c}\|_2^2]$$

is used to assess performance.

6.4.1 Sparse ARMA channel identification

In the first experiment sparse ARMA channel estimation is considered. The channel memory is $M_y = M_x = 50$ and the channel to be estimated is given by

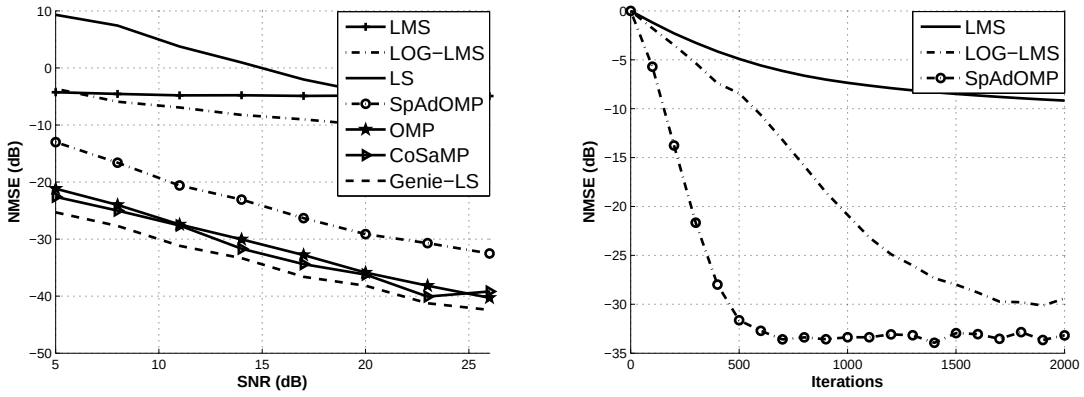
$$y_n = a_1 y_{n-6} + a_2 y_{n-48} + x_n + b_1 x_{n-13} + b_2 x_{n-34}$$

where $[a_1, a_2] = [-0.5167 - j0.2828, 0.1801 + j0.1347]$ and $[b_1, b_2] = [-0.5368 - j0.9198, 1.0719 + j0.0318]$. The system is stable as the roots of the AR part are inside the unit circle.

The input sequence is drawn from a complex Gaussian distribution, $\mathcal{CN}(0, 1/5)$. To reduce the realization dependency, the parameter estimates were averaged over 30 Monte Carlo runs. Program (P_{ℓ_0}) is solved by the CoSaMP [95], OMP [44, 100], log-LMS [36] and SpAdOMP. Moreover, two conventional methods were used, namely, the Least Squares (LS) and the LMS algorithm. The number of samples processed was 500. The sparsity tuning parameter required by the log-LMS is summarized in Table 6.2. The step size for the conventional LMS and the SpAdOMP was set to $\mu_{\text{LMS}} = 1 \times 10^{-2}$ and $\mu_{\text{SpAdOMP}} = 7 \times 10^{-2}$. Note that the choice of the step size μ is made near the instability point of each algorithm to provide the maximum possible convergence speed.

Table 6.2: Choice of sparse parameters for log-LMS

SNR	5-8	11-17	20-26
γ^1	7×10^{-4}	8×10^{-4}	9×10^{-4}



(a) ARMA channel

(b) Learning curve for SNR=23dB

Figure 6.1: NMSE of the channel estimates versus SNR on a linear channel

Fig. 6.1(a) shows the excellent performance match between the Genie LS, CoSaMP and OMP, all of which have quadratic complexity. The LMS, log-LMS and SpAdOMP have an order of magnitude less computational complexity, but only SpAdOMP achieves a performance gain close to Genie LS ($9dB$ less). If we repeat this experiment for a fixed SNR level of $23dB$ and process 2000 samples, then as shown in Fig. 6.1(b), log-LMS improves by $20dB$; however, it achieves $4dB$ less performance gain than SpAdOMP.

To demonstrate the support tracking ability of SpAdOMP, we run this experiment and after 300 iterations we set a_1 to zero. This time, since we have a support varying environment, λ is set to $\lambda = 0.8$ in SpAdOMP. Fig. 6.2 illustrates the time evolution of the estimates of a_1. We note from Fig. 6.2, that the conventional LMS does not take into account sparsity and hence the estimates are nonzero; while log-LMS and SpAdOMP succeed in estimating the zero entries. However, SpAdOMP has a much faster support tracking behavior for the estimation of the zero entries in comparison to log-LMS.

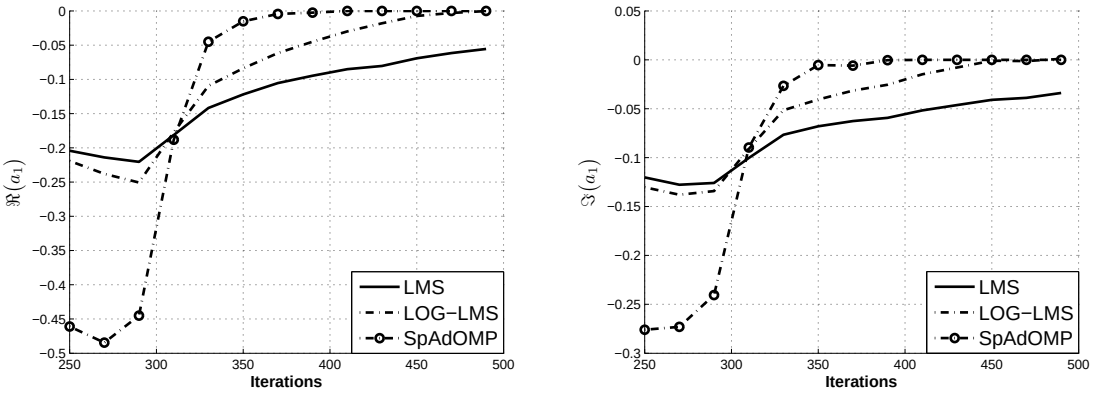
6.4.2 Sparse NARMA channel identification

In the second experiment the following NARMA channel is considered

$$y_n = a_1 y_{n-50} + a_2 y_{n-9}^2 + b_1 x_{n-8} + b_2 |x_{n-21}|^2 x_{n-21}$$

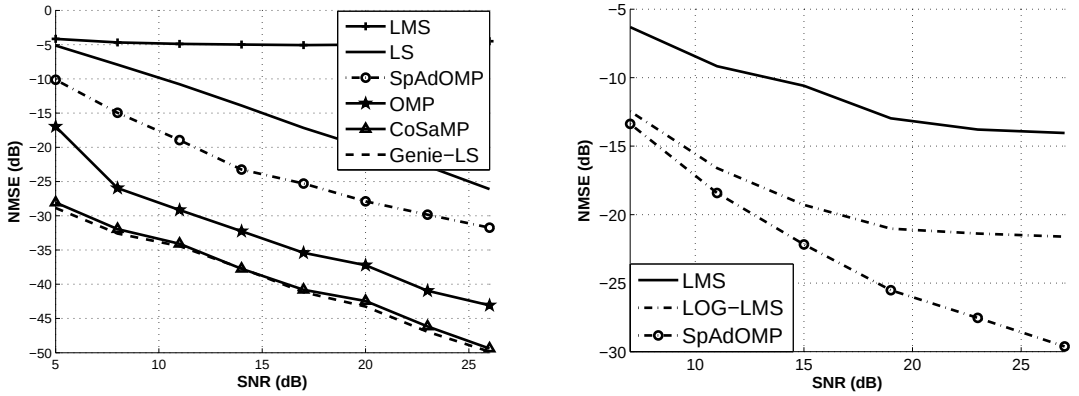
where $[a_1, a_2] = [-0.1586 - j0.7064, -0.1428 - j0.0478]$ and $[b_1, b_2] = [-0.8082 - j0.5221, -0.5177 + j0.7131]$ and the channel memory is $M_y = M_x = 50$.

The experiment is based on 30 Monte Carlo runs and the input sequence is generated from a complex Gaussian distribution, $\mathcal{CN}(0, 1/4)$, consisting of 1000 samples.



(a) Real part (b) Imaginary part

Figure 6.2: Time evolution of a_1 signal entry on a linear ARMA channel



(a) NARMA channel (b) Nonlinear multipath channel

Figure 6.3: NMSE of the channel estimates versus SNR on nonlinear channels

This time, the methods used are CoSaMP, OMP and SpAdOMP, along with the standard LMS algorithm and least squares. The step size parameters $\mu_{LMS} = 6\times10^{-3}$ and $\mu_{SpAdOMP} = 0.3$ are used for the conventional LMS and SpAdOMP. As Fig. 6.3(a) reveals, OMP and SpAdOMP lag behind Genie LS by $5dB$ and $12dB$ respectively

in performance. It is worth pointing out that SpAdOMP obtains an average gain of nearly $19dB$ over the conventional LMS. Note that this significant NMSE gain is the product of both the denoising mechanism and the compressed sampling virtue of the CoSaMP algorithm, which are lacking in conventional adaptive algorithms such as LMS.

6.4.3 Sparse nonlinear multipath channel identification

In this channel setup, a cubic baseband Hammerstein wireless channel with four Rayleigh fading rays (two on the liner and two on the cubic part) is employed; all rays fade at the same Doppler frequency of $f_D = 80Hz$ with sampling period $T_s = 0.8\mu s$. The channel memory length is equal to $M_1 = M_3 = 50$ (for both the linear and cubic parts) and the position of the fading rays is randomly chosen. In this experiment, 2000 samples from a complex Gaussian distribution $\mathcal{CN}(0, 1/4)$ were processed. Fig. 6.3(b) illustrates that SpAdOMP provides an average gain of $11dB$, over the conventional LMS and $5dB$ over the log-LMS developed in [36].

6.5 Proof of Theorem 6.3.2

Note that, unlike CoSaMP, the iterations of SpAdOMP are not applied to a fixed batch of measurements. Hence, we need to revisit the error analysis of CoSaMP taking into account the time variations. Recall that the LMS update for $\mathbf{w}_{|\Lambda(n)}(n)$ is given by

$$\mathbf{w}_{|\Lambda(n)}(n) = \mathbf{w}_{|\Lambda(n)}(n-1) + \mu\boldsymbol{\phi}^*_{|\Lambda(n)}(n)\big(y(n) - \boldsymbol{\phi}^T_{|\Lambda(n)}(n)\mathbf{w}_{|\Lambda(n)}(n-1)\big). \qquad (6.13)$$

Suppose that the estimate at time n is given by $\mathbf{c}(n)$. Let

$$\epsilon_1(n) := \|\mathbf{c} - \mathbf{c}(n)\|_2, \quad \epsilon_2(n) := \|\mathbf{w}_{|\Lambda(n)}(n) - \mathbf{w}_{o|\Lambda(n)}\|_2 \tag{6.14}$$

where $\mathbf{w}_{o|\Lambda(n)}$ is the *optimum Wiener solution* restricted to the set $\Lambda(n)$, given by

$$\mathbf{w}_{o|\Lambda(n)} := \mathbf{R}_{|\Lambda(n)}^{-1}\mathbf{r} \tag{6.15}$$

with $\mathbf{R} := \mathbb{E}\{\boldsymbol{\phi}(n)\boldsymbol{\phi}^{*T}(n)\}$ and $\mathbf{r} := \mathbb{E}\{\boldsymbol{\phi}^*(n)y(n)\}$. One can write

$$\begin{aligned}\mathbf{w}_{|\Lambda(n)}(n) - \mathbf{w}_{o|\Lambda(n)} = &\Big(\mathbf{I}_{|\Lambda(n)} - \mu\boldsymbol{\phi}_{|\Lambda(n)}(n)\boldsymbol{\phi}_{|\Lambda(n)}^{*T}(n)\Big) \\ &\times \Big\{\big(\mathbf{w}_{|\Lambda(n-1)}(n-1) - \mathbf{w}_{o|\Lambda(n-1)}\big) \\ &\quad + \big(\mathbf{w}_{|\Lambda(n)}(n-1) - \mathbf{w}_{|\Lambda(n-1)}(n-1)\big) \\ &\quad + \big(\mathbf{w}_{o|\Lambda(n-1)} - \mathbf{w}_{o|\Lambda(n)}\big)\Big\} + \mu\boldsymbol{\phi}_{|\Lambda}^*(n)e_o(n),\end{aligned} \tag{6.16}$$

where $e_o(n)$ is the estimation error of the optimum Wiener filter, given by $e_o(n) := (y(n) - \boldsymbol{\phi}_{|\Lambda}^T(n)\mathbf{w}_{o|\Lambda})$. Invoking the *Direct-Averaging* approximation [65], one can substitute $\boldsymbol{\phi}_{|\Lambda(n)}(n)\boldsymbol{\phi}_{|\Lambda(n)}^{*T}(n)$ with $\mathbf{R}_{|\Lambda(n)}$. Hence,

$$\begin{aligned}\epsilon_2(n) \le &(1 - \mu\lambda_M)\epsilon_2(n-1) + \mu\|\boldsymbol{\phi}_{|\Lambda}(n)\|_2|e_o(n)| \\ &+ (1 - \mu\lambda_M)\Big\{\big\|\mathbf{w}_{|\Lambda(n)}(n-1) - \mathbf{w}_{|\Lambda(n-1)}(n-1)\big\|_2 + \big\|\mathbf{w}_{o|\Lambda(n-1)} - \mathbf{w}_{o|\Lambda(n)}\big\|_2\Big\},\end{aligned} \tag{6.17}$$

where λ_M is the minimum eigenvalue of $\mathbf{R}$. Here we assume that the covariance matrix $\mathbf{R}$ is non-singular, *i.e.*, $\lambda_M > 0$. Note that the direct-averaging method yields a reasonable approximation particularly when $\mu \ll 1$ [81]. A more direct and rigorous convergence analysis of the LMS algorithm is possible, which is much more complicated [52]. Hence, for the sake of simplicity and clarity of the analysis, we proceed with the direct-averaging approach.

In order to obtain a closed set of difference equations for $\epsilon_1(n)$ and $\epsilon_2(n)$, we need to express the third and fourth terms of Eq. (7.17) in terms of $\epsilon_1(n)$ and $\epsilon_2(n)$ (and time-shifts thereof). First, we consider the third term. Let

$$\boldsymbol{\delta}(n-1) := \mathbf{w}(n-1) - \mathbf{c}. \tag{6.18}$$

Note that $\mathbf{w}(n-1)$ is supported on the index set $\Lambda(n-1)$. Hence,

$$\begin{aligned} \left\|\mathbf{w}_{|\Lambda(n)}(n-1) - \mathbf{w}_{|\Lambda(n-1)}(n-1)\right\|_2 &= \left\|\mathbf{c}_{\Lambda(n)\Delta\Lambda(n-1)} + \boldsymbol{\delta}_{\Lambda(n)\Delta\Lambda(n-1)}(n-1)\right\|_2 \\ &\leq \left\|\mathbf{c}_{\Lambda(n)\Delta\Lambda(n-1)}\right\|_2 + \left\|\boldsymbol{\delta}(n-1)\right\|_2 \end{aligned} \tag{6.19}$$

where Δ represents the symmetric difference of $\Lambda(n)$ and $\Lambda(n-1)$. The key here is the fact that the support estimates $\Lambda(n-1)$ and $\Lambda(n)$ contain most of the energy of the true vector $\mathbf{c}$, due to the restricted isometry of the measurement matrix and the construction of the proxy signal. Consider the squared form of the first term in the above equation:

$$\begin{aligned} \left\|\mathbf{c}_{\Lambda(n)\Delta\Lambda(n-1)}\right\|_2^2 &= \left\|\mathbf{c}_{\Lambda(n)\cap\Lambda^c(n-1)}\right\|_2^2 + \left\|\mathbf{c}_{\Lambda(n-1)\cap\Lambda^c(n)}\right\|_2^2 \\ &\leq \left\|\mathbf{c}_{\Lambda^c(n-1)}\right\|_2^2 + \left\|\mathbf{c}_{\Lambda^c(n)}\right\|_2^2 \end{aligned} \tag{6.20}$$

Hence,

$$\left\|\mathbf{c}_{\Lambda(n)\Delta\Lambda(n-1)}\right\|_2 \leq \sqrt{2}\max\left\{\left\|\mathbf{c}_{\Lambda^c(n-1)}\right\|_2, \left\|\mathbf{c}_{\Lambda^c(n)}\right\|_2\right\} \tag{6.21}$$

Lemmas 4.2 and 4.3 of [95] provide the following bound on $\|\mathbf{c}_{\Lambda^c(n)}\|_2$:

$$\left\|\mathbf{c}_{\Lambda^c(n)}\right\|_2 \leq \gamma(n)\epsilon_1(n-1) + \xi(n)\left\|\boldsymbol{\eta}'(n)\right\|_2 \tag{6.22}$$

where

$$\gamma(n) := \frac{\delta_{2s}(\lambda,n) + \delta_{4s}(\lambda,n)}{1-\delta_{2s}(\lambda,n)}, \quad \xi(n) := \frac{2\sqrt{1+\delta_{2s}(\lambda,n)}}{1-\delta_{2s}(\lambda,n)}, \tag{6.23}$$

and

$$\boldsymbol{\eta}'(n) := \mathbf{D}^{1/2}(n)\boldsymbol{\eta}(n) + \mathrm{Diag}\left(\boldsymbol{\Phi}(n)\boldsymbol{\Theta}(n)\right) \tag{6.24}$$

with $\boldsymbol{\Theta}_{ij}(n) := \lambda^{n-j-1}\big(\mathbf{c}_i(n) - \mathbf{c}_i(j)\big)$, for $i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, n-1$. The *effective* noise vector $\boldsymbol{\eta}'(n)$ consists of two parts: the first term is the exponentially-weighted additive noise vector, and the second term is the excess error due to the adaptive update of the proxy signal (in contrast to the batch construction used in the CoSaMP algorithm). Note that the isometry constants $\delta_s(\lambda, n), \cdots, \delta_{4s}(\lambda, n)$ are all functions of n, since the matrix $\boldsymbol{\Phi}(n)$ depends on n. If the input sequence is generated by a stationary source, for n large enough, one can approximate the covariance matrix $\mathbf{R}$ by the exponentially weighted sample covariance $\boldsymbol{\Phi}^{*T}(n)\mathbf{D}(n)\boldsymbol{\Phi}(n)$. Similarly, one can approximate $\mathbf{r}$ by $\boldsymbol{\Phi}^{*T}(n)\mathbf{D}(n)\mathbf{r}(n)$. In this case, we have $\mathbf{w}_{o|\Lambda(n)} \approx \mathbf{b}(n)$, where $\mathbf{b}(n)$ is the exponentially-weighted least squares solution restricted to the index set $\Lambda(n)$, given by

$$\mathbf{b}(n) := \begin{cases} \big(\mathbf{D}^{1/2}(n)\boldsymbol{\Phi}(n)\big)^{\dagger}_{|\Lambda(n)}\mathbf{D}^{1/2}(n)\mathbf{r}(n), & \text{on } \Lambda(n) \\ 0, & \text{elsewhere} \end{cases} \tag{6.25}$$

Using this approximation, the ℓ_2-norm of $\boldsymbol{\delta}(n-1)$ can be bounded as follows:

$$\begin{aligned} \big\|\boldsymbol{\delta}(n-1)\big\|_2 &\leq \big\|\mathbf{w}(n-1) - \mathbf{b}(n-1)\big\|_2 + \big\|\mathbf{b}(n-1) - \mathbf{c}\big\|_2 \\ &\leq \epsilon_2(n-1) + \big\|\mathbf{b}(n-1) - \mathbf{c}\big\|_2 \end{aligned} \tag{6.26}$$

Moreover, using Lemmas 4.2, 4.3, and 4.4 of [95], one can express $\|\mathbf{c} - \mathbf{b}(n)\|_2$ in terms of $\epsilon_1(n)$ and $\boldsymbol{\eta}'(n)$ as follows:

$$\|\mathbf{c} - \mathbf{b}(n)\|_2 \leq \frac{1}{2}\alpha(n)\epsilon_1(n-1) + \frac{1}{2}\beta(n)\big\|\boldsymbol{\eta}'(n)\big\|_2, \tag{6.27}$$

where

$$\alpha(n) := 2\Big(1 + \frac{\delta_{4s}(\lambda, n)}{1 - \delta_{3s}(\lambda, n)}\Big)\gamma(n), \tag{6.28}$$

$$\beta(n) := \frac{2}{\sqrt{1 - \delta_{3s}(\lambda, n)}} + 2\Big(1 + \frac{\delta_{4s}(\lambda, n)}{1 - \delta_{3s}(\lambda, n)}\Big)\xi(n). \tag{6.29}$$

Denoting $\|\mathbf{c} - \mathbf{b}(n)\|_2$ by $\epsilon_3(n)$ and using Eqs. (6.21), (6.26), and (6.27), one can obtain the following recurrence relation for $\epsilon_2(n)$:

$$\begin{aligned}
\epsilon_2(n) &\leq (1 - \mu\lambda_M)\epsilon_2(n-1) + \mu\|\boldsymbol{\phi}_{|\Lambda}(n)\|_2|e_o(n)| \\
&\quad + (1 - \mu\lambda_M)\Big\{\big\|\mathbf{w}_{o|\Lambda(n-1)} - \mathbf{c}\big\|_2 + \big\|\mathbf{w}_{o|\Lambda(n)} - \mathbf{c}\big\|_2 \\
&\qquad\qquad + \big\|\mathbf{w}_{|\Lambda(n)}(n-1) - \mathbf{w}_{|\Lambda(n-1)}(n-1)\big\|_2\Big\} \\
&\leq (1 - \mu\lambda_M)\epsilon_2(n-1) + \mu\big\|\boldsymbol{\phi}_{|\Lambda}(n)\big\|_2|e_o(n)| \\
&\quad + (1 - \mu\lambda_M)\Big\{\epsilon_3(n) + 2\epsilon_3(n-1) + \epsilon_2(n-1)\Big\} \\
&\quad + \sqrt{2}(1 - \mu\lambda_M)\max\Big\{\gamma(n)\epsilon_1(n-1) + \xi(n)\|\boldsymbol{\eta}'(n)\|_2, \\
&\qquad\qquad \gamma(n-1)\epsilon_1(n-2) + \xi(n-1)\|\boldsymbol{\eta}'(n-1)\|_2\Big\}
\end{aligned} \tag{6.30}$$

From Lemma 4.5 of Needell et al. [95], one can write

$$\begin{aligned}
\epsilon_1(n) &:= \|\mathbf{c} - \mathbf{c}(n)\|_2 \\
&\leq \|\mathbf{c} - \mathbf{b}_s(n)\|_2 + \|\mathbf{b}_s(n) - \mathbf{c}(n)\|_2 \\
&\leq 2\|\mathbf{c} - \mathbf{b}(n)\|_2 + 4\|\mathbf{b}(n) - \mathbf{w}(n)\|_2 \\
&\leq 2\|\mathbf{c} - \mathbf{b}(n)\|_2 + 4\|\mathbf{w}_{o|\Lambda(n)} - \mathbf{w}(n)\|_2 + 4\|\mathbf{w}_{o|\Lambda(n)} - \mathbf{b}(n)\|_2,
\end{aligned} \tag{6.31}$$

where the last line of Eq. (6.31) is obtained from the second line by adding and subtracting $\mathbf{w}_{o|\Lambda(n)}$ from $\mathbf{b}(n) - \mathbf{w}(n)$, and using the triangle inequality. The last term on the right hand side of Eq. (6.31) denotes the difference between the optimum Wiener

solution and the LS solution, both restricted to the index set $\Lambda(n)$. As mentioned earlier, one can approximate the covariance matrix $\mathbf{R}$ by the exponentially weighted sample covariance $\mathbf{\Phi}^{*T}(n)\mathbf{D}(n)\mathbf{\Phi}(n)$, and the correlation vector $\mathbf{r}$ by $\mathbf{\Phi}^{*T}(n)\mathbf{D}(n)\mathbf{r}(n)$. In this case, we have $\mathbf{w}_{o|\Lambda(n)} \approx \mathbf{b}(n)$, and hence the contribution of the last term on the right hand side of Eq. (6.31) to the steady-state error becomes negligible. Also, by construction, the estimate $\mathbf{w}(n)$ is supported on the index set $\Lambda(n)$. Hence, the second term of Eq. (6.31) can be identified as $4\|\mathbf{w}_{o|\Lambda(n)} - \mathbf{w}_{|\Lambda(n)}(n)\|_2 = 4\epsilon_2(n)$. With the above-mentioned simplifications, one can arrive at the following set of non-linearly coupled difference equations for $\epsilon_1(n)$, $\epsilon_2(n)$ and $\epsilon_3(n)$:

$$\begin{cases} \epsilon_1(n) \leq 2\epsilon_3(n) + 4\epsilon_2(n), \\ \epsilon_2(n) \leq (1-\mu\lambda_M)\Big\{2\epsilon_2(n-1) + 2\epsilon_3(n-1) + \epsilon_3(n)\Big\} \\ \qquad +\sqrt{2}(1-\mu\lambda_M)\max\Big\{\gamma(n)\epsilon_1(n-1) + \xi(n)\|\boldsymbol{\eta}'(n)\|_2, \\ \qquad\qquad \gamma(n-1)\epsilon_1(n-2) + \xi(n-1)\|\boldsymbol{\eta}'(n-1)\|_2\Big\} \\ \qquad +\mu\big\|\boldsymbol{\phi}_{|\Lambda(n)}(n)\big\|_2|e_o(n)|, \\ \epsilon_3(n) \leq \frac{1}{2}\alpha(n)\epsilon_1(n-1) + \frac{1}{2}\beta(n)\big\|\boldsymbol{\eta}'(n)\big\|_2. \end{cases} \tag{6.32}$$

Although the above set of difference equations is sufficient to obtain the error measures $\epsilon_1(n)$, $\epsilon_2(n)$, and $\epsilon_3(n)$ for all n, the solution is non-trivial for general n due to its high non-linearity. However, for large n, it is possible to obtain the steady-state solution. It is easy to substitute $\epsilon_3(n)$ in terms of $\epsilon_1(n)$. Also, for large enough n, the arguments of the $\max\{\cdot,\cdot\}$ operator do not vary significantly with n. Hence, we can substitute the maximum with the second argument. Hence, the steady-state values of $\epsilon_1(n)$ and $\epsilon_2(n)$ can be obtained from the following equation:

$$\begin{pmatrix} 1-\alpha(n) & -4 \\ -(1-\mu\lambda_M)\big(\frac{3}{2}\alpha(n)+\sqrt{2}\gamma(n)\big) & 1-2(1-\mu\lambda_M) \end{pmatrix} \begin{pmatrix} \epsilon_1(n) \\ \epsilon_2(n) \end{pmatrix}$$
$$\leq \big\|\mathbf{D}^{1/2}(n)\boldsymbol{\eta}(n)\big\|_2 \begin{pmatrix} \beta(n) \\ (1-\mu\lambda_M)\big(\frac{3}{2}\beta(n)+\sqrt{2}\xi(n)\big) \end{pmatrix} + \mu\big\|\boldsymbol{\phi}_{|\Lambda(n)}(n)\big\|_2 |e_o(n)| \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (6.33)$$

Note that, the contribution of proxy error term in $\boldsymbol{\eta}'(n)$ becomes negligible for large n, due to the effect of forgetting factor, and the fact that the estimates $\mathbf{c}(n)$ do not vary much with n. Hence, we can approximate $\boldsymbol{\eta}'(n)$ by $\mathbf{D}^{1/2}(n)\boldsymbol{\eta}(n)$ for large n. In particular, the asymptotic solution to $\epsilon_1(n)$ is given by:

$$\epsilon_1(n) \lesssim C_1(n)\big\|\mathbf{D}^{1/2}(n)\boldsymbol{\eta}(n)\big\|_2 + C_2(n)\big\|\boldsymbol{\phi}_{|\Lambda(n)}(n)\big\|_2 |e_o(n)| \quad (6.34)$$

where,

$$C_1(n) := \frac{4(1-\mu\lambda_M)\big(\frac{3}{2}\beta(n)+\sqrt{2}\xi(n)\big)}{\Delta(n)} + \frac{\big(1-2(1-\mu\lambda_M)\big)\beta(n)}{\Delta(n)}, \quad (6.35)$$

$$C_2(n) := \frac{4\mu}{\Delta(n)}, \quad (6.36)$$

and

$$\Delta(n) := (2\mu\lambda_M - 1) - (5-4\mu\lambda_M)\alpha(n) - 4\sqrt{2}(1-\mu\lambda_M)\gamma(n). \quad (6.37)$$

Note that a sufficient condition for the above bound to hold is $\Delta(n) > 0$.

Chapter 7

Construction of Pseudo-Random Matrices for Compressed Sensing

7.1 Introduction

As mentioned in the introduction, several researchers have tried to simplify the recovery process of compressed sensing by introducing structure into the measurement matrix. The structure of the measurement matrix can potentially reduce both the computational and storage costs.

In particular, several structured measurement matrices based on error correcting codes have been proposed. Jafarpour et al. have proposed methods based on encoding and decoding of codes from expander graphs in [72]. In [1, 2], Akgakaya et al. have introduced the low density frames inspired by LDPC codes, which allow fast recovery using belief propagation techniques. Howard et al. [69] have proposed measurement matrices based on Reed-Muller codes along with a fast recovery algo-

rithm. Calderbank et al. [24] have constructed measurement matrices from the cosets of second order Reed-Muller codes, such as Delsarte-Goethals and Kerdock codes. It has been shown that these matrices satisfy a statistical version of the RIP, namely UStRIP, which guarantees their near isometry property on all but a small fraction of the underlying sparse signals.

In this chapter, we study the spectral properties of random matrices from binary block codes, and show that under certain conditions these matrices resemble the spectral behavior of i.i.d. random matrices [13,14]. It is well-known that several i.i.d. random matrices satisfy the RIP (See, for example, [29] and [17]), which makes them very appealing for compressed sensing. Therefore, random matrices from binary block codes can be viewed as pseudo-random counterparts to i.i.d. random matrices, with similarly appealing spectral behaviors.

Apart from the potential utility of random matrices from binary block codes in compressed sensing, studying their spectral behavior is independently interesting from the viewpoint of random matrix theory. It is well-known that the spectrum of certain random matrices converges to deterministic distributions as the dimensions grow to infinity. Canonical examples are the Wishart distribution for the sample covariance matrix of a multivariate normal distribution [129], the Wigner semicircle law for the asymptotic empirical spectral distribution of real symmetric i.i.d. random matrices [128] and the Marchenko-Pastur law for the empirical spectral distribution of the Gram matrix of real i.i.d. random matrices [90]. But, to the best of our knowledge, very little is known about the spectral behavior of random matrices from binary combinatorial structures.

Let $\mathcal{C}$ be an $[n, k, d]$ binary linear block code of length n, dimension k and minimum Hamming distance d over $\mathrm{GF}(2)^n$. The dual code of $\mathcal{C}$, denoted by $\mathcal{C}^\perp$ is an $[n, n-k, d^\perp]$ binary linear block code over $\mathrm{GF}(2)^n$ such that all the codewords of $\mathcal{C}^\perp$ are orthogonal to those of $\mathcal{C}$ with the inner product defined over $\mathrm{GF}(2)^n$. Let $\varepsilon : \mathrm{GF}(2)^n \mapsto \{-1, 1\}^n$ be the component-wise mapping $\varepsilon(v_i) := (-1)^{v_i}$, for $\mathbf{v} = (v_1, v_2, \cdots, v_n) \in \mathrm{GF}(2)^n$. Finally, for $p < n$, let $\mathbf{\Phi}_\mathcal{C}$ be a $p \times n$ random matrix whose rows are obtained by mapping a uniformly drawn set of size p of the codewords $\mathcal{C}$ under ε.

We study the empirical spectral distribution of the Gram matrix of $\frac{1}{\sqrt{n}}\mathbf{\Phi}_\mathcal{C}$ and show that for $d^\perp$ sufficiently large, the asymptotic empirical spectral distribution is very similar to that of random i.i.d. Rademacher matrices, which is given by the Marchenko-Pastur distribution. We uniformly bound the distance of the asymptotic empirical spectral distribution of the Gram matrix of $\frac{1}{\sqrt{n}}\mathbf{\Phi}_\mathcal{C}$ to the Marchenko-Pastur distribution as a function of $y := p/n$ and $d^\perp$. Numerical experiments on low-rate BCH codes confirm the theoretical results. To the best of our knowledge, this is the first result relating the randomness of a matrix from a binary vector space to the algebraic properties of the underlying dual space. Not only this result is interesting from the random matrix theory viewpoint, but also it introduces a new criterion for the joint randomness of block codes or sequences.

The outline of the chapter follows next. In Section 7.2, we introduce the notation and state the Main Result of this chapter followed by a discussion of the Main Result, as well as numerical simulations. The detailed proof of the Main Result is presented in Section 7.3. Finally, we study the group randomness properties of shortened first

order Reed-Muller codes and Gold sequences in Section 7.4.

7.2 Main Result

7.2.1 Definitions and the Main Theorem

Before presenting the Main Result, we introduce the notation and state some preliminary definitions:

A $(n, \mathcal{M}, d)$ binary code $\mathcal{C}$ is defined as a set of $\mathcal{M}$ binary n-tuples such that any two such n-tuples differ in at least d places, with d being the largest number with this property. The Hamming weight of an n-tuple $\mathbf{u} \in \mathrm{GF}(2)^n$, denoted by $\mathrm{wt}(\mathbf{u})$, is defined as the number of non-zero elements of $\mathbf{u}$.

Consider the group algebra over $\mathrm{GF}(2)^n$, in which the code $\mathcal{C}$ is represented by the element

$$C := \sum_{\mathbf{u} \in \mathrm{GF}(2^n)} c_{\mathbf{u}} z^{\mathbf{u}}, \tag{7.1}$$

where $z^{\mathbf{u}} := z_1^{\mathbf{u}_1} z_2^{\mathbf{u}_2} \cdots z_n^{\mathbf{u}_n}$, and

$$c_{\mathbf{u}} := \begin{cases} 1 & \text{if } \mathbf{u} \in \mathcal{C} \\ 0 & \text{otherwise.} \end{cases} \tag{7.2}$$

For a binary n-tuple $\mathbf{u} \in \mathrm{GF}(2)^n$, let $\chi_{\mathbf{u}}$ be the character mapping

$$\chi_{\mathbf{u}}(z^{\mathbf{v}}) = (-1)^{\mathbf{u} \cdot \mathbf{v}} \tag{7.3}$$

with $\mathbf{u} \cdot \mathbf{v} := \sum_i u_i v_i \mod 2$, for all $\mathbf{v} \in \mathrm{GF}(2)^n$. Suppose that for a code $\mathcal{C}$, corre-

sponding to the element of the group algebra given by Eq. (7.1), we have

$$\mathcal{M} = \sum_{\mathbf{u} \in \mathrm{GF}(2)^n} c_{\mathbf{u}} \neq 0. \tag{7.4}$$

Now, consider the normalized linear span of C

$$D := \frac{1}{\mathcal{M}} C^2 = \sum_{\mathbf{u} \in \mathrm{GF}(2)^n} d_{\mathbf{u}} z^{\mathbf{u}}. \tag{7.5}$$

The *distance distribution* of the code $\mathcal{C}$ is defined as the set $\{w_0, w_1, \cdots, w_n\}$, where

$$w_i := \sum_{\mathrm{wt}(\mathbf{u})=i} d_{\mathbf{u}}. \tag{7.6}$$

The *transformed* distance distribution of the code $\mathcal{C}$ under the character mapping is given by the set $\{w'_0, \cdots, w'_n\}$, where

$$w'_j := \frac{1}{\mathcal{M}} \sum_{\mathrm{wt}(\mathbf{u})=j} \chi_{\mathbf{u}}(D). \tag{7.7}$$

Finally, the *dual distance* of the code $\mathcal{C}$ is defined as $d^\perp$ such that $w'_i = 0$ for $1 \leq i \leq d^\perp - 1$ and $w'_{d^\perp} \neq 0$. Note that when the code $\mathcal{C}$ is linear, we refer to the distance distribution and the transformed distance distribution as the weight distribution and the dual weight distribution, respectively. The *dual* of an $[n, k, d]$ binary linear block code $\mathcal{C}$, is an $[n, n-k, d^\perp]$ binary linear block code, denoted by $\mathcal{C}^\perp$, with the property that all the codewords of $\mathcal{C}^\perp$ are orthogonal to those of $\mathcal{C}$.

We say that a $p \times n$ random matrix $\mathbf{\Phi}$ is *based on* a binary linear block code $\mathcal{C}$, if the rows of $\mathbf{\Phi}$ are obtained by mapping a uniformly drawn set of size p of the codewords $\mathcal{C}$ under ε. In other words, let $\mathbf{c}_i := (c_{i1}, c_{i2}, \cdots, c_{in})$ be a codeword of the code $\mathcal{C}$. Then, for a randomly drawn set of p codewords $\mathcal{C}_p := \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_p\}$, we have $\mathbf{\Phi}_{ij} = (-1)^{c_{ij}}$ for $i = 1, 2, \cdots, p$ and $j = 1, 2, \cdots, n$. The *Gram matrix* of the $p \times n$ matrix $\mathbf{\Phi}$ is defined as $\mathcal{G} := \mathbf{\Phi}\mathbf{\Phi}^T$.

Let $\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ denote the set of eigen-values of an $n \times n$ matrix $\mathbf{A}$. Then, the *spectral measure* of $\mathbf{A}$ is defined by

$$\mu_{\mathbf{A}} := \frac{1}{n} \sum_{i=1}^{n} \delta_{\lambda_i} \tag{7.8}$$

where δ_z is the Dirac measure. The *empirical spectral distribution* of $\mathbf{A}$ is defined as

$$M_{\mathbf{A}}(z) := \int_{-\infty}^{z} \mu_{\mathbf{A}}(dz) \tag{7.9}$$

In particular, we denote by $M_{MP}(z)$ the distribution corresponding to the Marchenko-Pastur measure μ_{MP} whose density is given by

$$\frac{d\mu_{MP}}{dz} := \frac{1}{2\pi zy} \sqrt{(b-z)(z-a)} 1_{(a \le z \le b)} \tag{7.10}$$

with $a = (1 - \sqrt{y})^2$ and $b = (1 + \sqrt{y})^2$.

The main result of this chapter can be summarized as the following theorem:

Theorem 7.2.1 (Main Theorem)**.** *Consider a sequence of $[n, k_n, d_n]$ binary linear block codes $\{\mathcal{C}_n\}_{n=1}^{\infty}$. Let $\mathbf{\Phi}_{\mathcal{C}_n}$ be a $p \times n$ random matrix based on $\mathcal{C}_n$, $\mathcal{G}_{\mathcal{C}_n}$ denote the Gram matrix of the matrix $\frac{1}{\sqrt{n}}\mathbf{\Phi}_{\mathcal{C}_n}$, and $M_{\mathcal{C}_n}(z)$ denote the empirical spectral distribution of $\mathcal{G}_{\mathcal{C}_n}$. Finally, let r_n be the greatest even integer less than or equal to $[(d_n^{\perp} - 1)/2]$, and let $r := \liminf_n r_n$. Then, as $n \to \infty$ with $y := p/n \in (0, 1)$ fixed, we have*

$$\limsup_n \left| M_{\mathcal{C}_n}(z) - M_{MP}(z) \right| \le c(y, r) \left(\frac{1}{r} + \frac{1}{r^2} \right) \tag{7.11}$$

almost surely for all z, where $c(y, r)$ is a bounded function of r (explicitly given in this chapter).

7.2.2 Discussion of the Main Theorem

The Main Theorem states that the empirical spectral distribution of the Gram matrix of the random matrix $\frac{1}{\sqrt{n}}\mathbf{\Phi}_{\mathcal{C}}$ based on a binary linear block code $\mathcal{C}$ is very close to the universal empirical spectral distribution of the Gram matrix of random i.i.d. generated matrices (e.g., i.i.d. Gaussian or Rademacher matrices) as $n \to \infty$, if the dual distance of the code $\mathcal{C}$ is large enough. In particular, an i.i.d. generated Rademacher matrix can be viewed as a random matrix based on the $[n, n, 1]$ binary linear block code. This interpretation is reminiscent of the celebrated result by Sidel'nikov [112]: for an $[n, k, d]$ binary linear block code $\mathcal{C}$ with $d^{\perp} \geq 3$, we have:

$$\left|A(z) - \Phi(z)\right| \leq \frac{9}{\sqrt{d^{\perp}}} \tag{7.12}$$

as $n \to \infty$, where $A(z)$ is the cumulative weight distribution function of the code $\mathcal{C}$ and

$$\Phi(z) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2} dt. \tag{7.13}$$

In this case, $\Phi(z)$ is the limit cumulative weight distribution of the $[n, n, 1]$ code as $n \to \infty$ (See Ch. 9 of [86] for details). Note that the dual distance of the code $\mathcal{C}$ gives a distortion metric for comparing the asymptotic empirical distribution of a random matrix based on $\mathcal{C}$ to the Marchenko-Pastur distribution. Moreover, the main result implies that for a sequence of codes $\mathcal{C}_n$ for which the dual distance remains finite as $n \to \infty$ (e.g., r-th order Reed-Muller codes, $\mathscr{R}(r, m)$ [86] with r fixed as $n \to \infty$), the distortion can be quite small if the asymptotic dual distance is large enough. However, the converse is not necessarily true.

The Main Theorem can be interpreted as a joint randomness test for codes or sequences. For instance, consider pseudo-noise sequences which are widely used in

several communications applications such as range-finding, synchronization, modulation and scrambling [86]. A class of such sequences corresponds to $[2^m - 1, m, 2^{m-1}]$ shortened first-order Reed-Muller codes $\mathscr{S}_m$ [86]. Although shortened first-order Reed-Muller codes have very appealing randomness properties along individual codewords [86], it is straightforward to show that the spectral properties of random matrices from these codes dramatically differ from those of random i.i.d. generated matrices. Let $\mathbf{\Phi}_{\mathscr{S}}$ be a full-rank $p \times n$ random matrix based on $\mathscr{S}_m$, with $n := 2^m - 1$ and $y := p/n < 1$. Then, it can be easily verified that the eigen-values of the Gram matrix of $\frac{1}{\sqrt{n}}\mathbf{\Phi}_{\mathscr{S}}$ are equal to $1-y+1/n$ and $1+1/n$ with multiplicities 1 and $p-1$, respectively [9]. Figure 7.1 shows the empirical spectral distribution of a random matrix based on $\mathscr{S}_8$ versus the Marchenko-Pastur distribution for $y = 63/255$. As it is implied from the figure, the two distributions are indeed different. Hence, pseudo-noise sequences do not possess the joint randomness property of random i.i.d. generated sequences. In contrast, the empirical spectral distribution of random matrices from codes with larger dual distances (e.g., BCH codes) resembles that of random i.i.d. generated matrices, as predicted by the Main Theorem.

Furthermore, the Main Theorem sheds light on the observations reported in the literature of compressed sensing regarding the similarity of the behavior of the empirical spectral distribution of sub-matrices of structured frames to those of random frames (See, for example, [8] and [24]). In particular, Calderbank et al. [24] present numerical experiments which reveal the similarity of the behavior of the condition number of sub-matrices of frames from Reed-Muller codes with that of random Gaussian matrices.

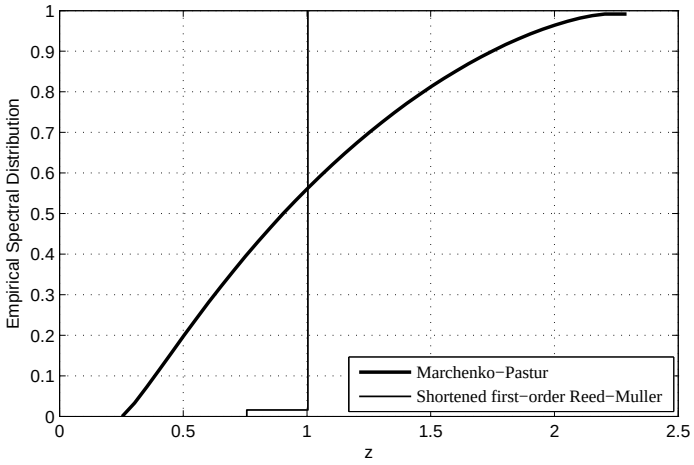


Figure 7.1: Empirical spectral distribution for a random matrix based on the $[255, 8, 128]$ shortened first-order Reed-Muller code vs. Marchenko-Pastur distribution, for $y = 63/255$.

Also, note that the main result, which is valid for $p < n$, can not be readily generalized to the case of $p > n$ due to a fundamental loss of symmetry: as opposed to the i.i.d. random matrix case, one can not interchange the roles of p and n, since the columns of the resulting random matrix do not necessarily correspond to the codewords of a code with the same dual distance. So is the case of random matrices whose columns are i.i.d. selected, but whose rows have statistical correlations (See, for example, [111]).

Finally, note that $\liminf_n d_n^\perp$ must be sufficiently large for the upper bound to be meaningful. For example, for $y = 0.5$, we need $\liminf_n d_n^\perp \geq 52$ for the bound to be less than 1. Moreover, although the result is asymptotic, numerical experiments on low-rate BCH codes reveal the significant similarity of the empirical distributions to the Marchenko-Pastur distribution for dimensions (and consequently, dual distances) as small as $n = 63$. Figures 7.2-7.5 show the empirical spectral distribution and density for matrices based on [63,30,6], [127,15,27], [255,21,55] and [511, 28, 111]

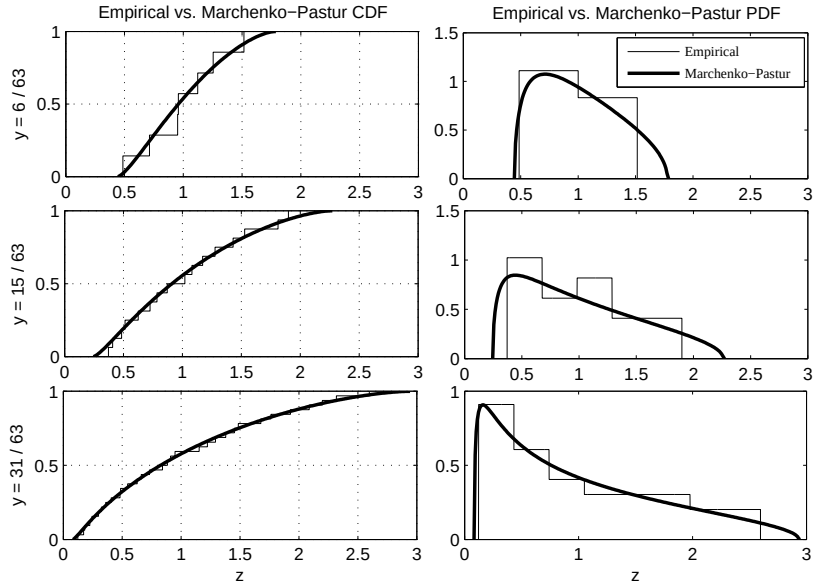


Figure 7.2: Empirical spectral distribution and density for random matrices based on a [63,30,6] BCH code for $y = 6/63, 15/63$ and $31/63$.

BCH codes for $y := p/n = 0.1, 0.25$ and 0.5, respectively. As it can be observed from these figures, the empirical distributions and densities of these matrices are very similar to those of Marchenko-Pastur. In fact, for the $[511, 28, 111]$ case, the distributions are almost indistinguishable, even for y as small as 0.1.

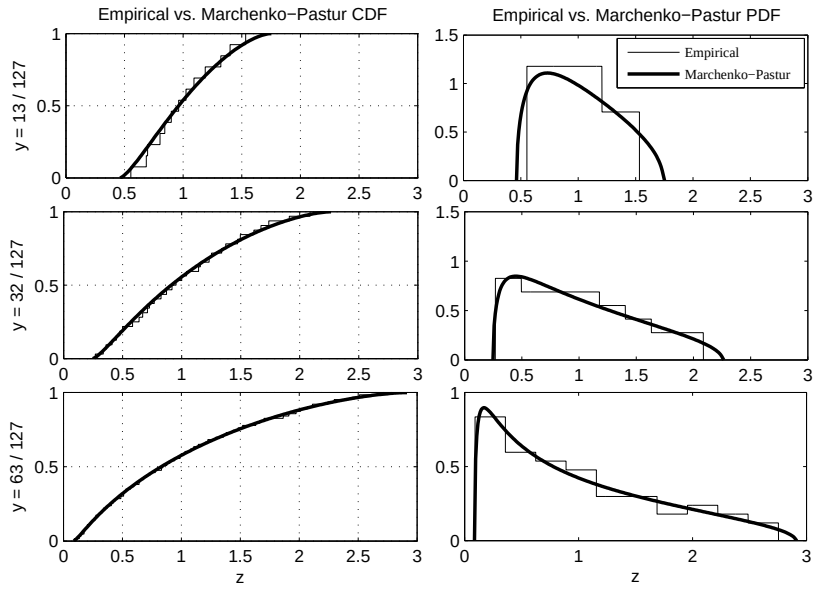


Figure 7.3: Empirical spectral distribution and density for random matrices based on a [127,15,27] BCH code for $y = 13/127, 32/127$ and $63/127$.

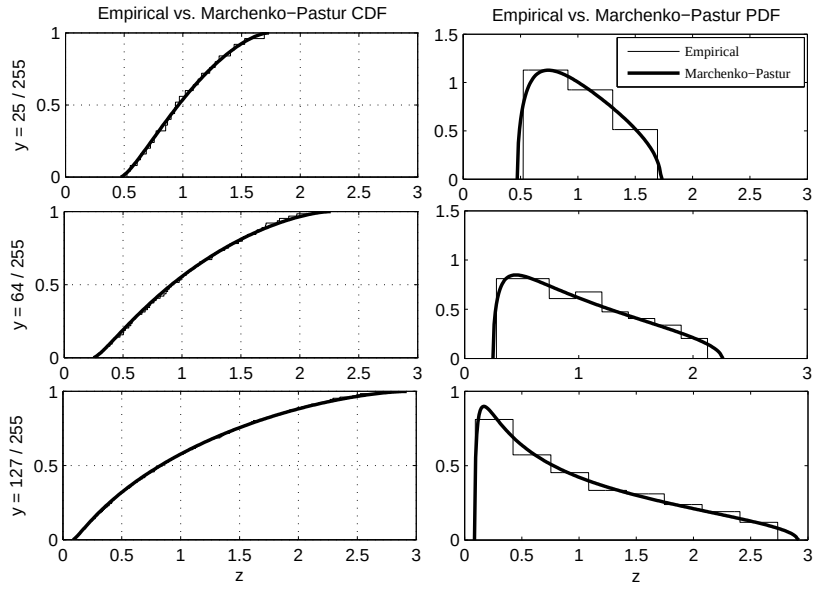


Figure 7.4: Empirical spectral distribution and density for random matrices based on a [255,21,55] BCH code for $y = 25/255, 64/255$ and $127/255$.

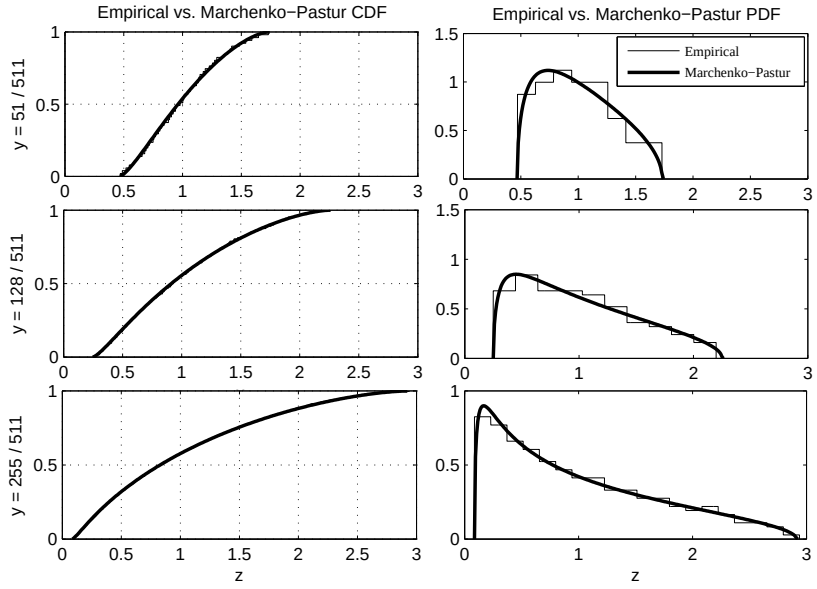


Figure 7.5: Empirical spectral distribution and density for random matrices based on a [511,28,111] BCH code for $y = 51/511, 128/511$ and $255/511$.

7.3 Proof of the Main Theorem

Before presenting the proof of the Main Theorem, we need to establish a number of lemmas. First, we state a lemma from probability theory, which is discussed in detail in [54, Ch. XVI-3]:

Lemma 7.3.1. *Let F be a probability distribution with vanishing expectation and characteristic function ϕ. Suppose that $F - G$ vanishes at $\pm\infty$ and that G has a derivative g such that $|g| \leq m$. Finally, suppose that g has a continuously differentiable Fourier transform γ such that $\gamma(0) = 1$ and $\gamma'(0) = 0$. Then, for all z and $T > 0$ we have*

$$|F(z) - G(z)| \leq \frac{1}{\pi} \int_{-T}^{T} \left| \frac{\phi(t) - \gamma(t)}{t} \right| dt + \frac{24m}{\pi T} \tag{7.14}$$

Proof. The proof uses smoothing techniques to cope with the fact that the underlying density of F may not enjoy sufficient smoothness properties. The detailed proof can be found in [54] and is thus omitted for brevity. □

The next lemma establishes a basic fact regarding the minimum distance of the dual code of a binary block code:

Lemma 7.3.2. *Let $\mathcal{C}$ be an $(n, \mathcal{M}, d)$ binary block code with dual distance $d^\perp$. Then, any set of $r \leq d^\perp - 1$ coordinates of the codewords of $\mathcal{C}$ contains each binary r-tuple exactly $\mathcal{M}/2^r$ times, and $d^\perp$ is the largest number with this property.*

Proof. By the definition of $d^\perp$ and the properties of the character mapping, we have $\chi_{\mathbf{u}}(C) = 0$ for all $\mathbf{u}$ with $\mathrm{wt}(\mathbf{u}) = 1, 2, \cdots, d^\perp - 1$. For $\mathrm{wt}(\mathbf{u}) = 1$, this implies that each component of the codewords takes the values of 0 and 1 a total of $\mathcal{M}/2$ times each. For $\mathrm{wt}(\mathbf{u}) = 2$, this implies that any set of two components of the codewords take the combinations $00, 01, 10, 11$ a total of $\mathcal{M}/4$ times each, etc. Hence, any set of $d^\perp - 1$ components of the codewords take all the possible $(d^\perp - 1)$-tuples a total of $\mathcal{M}/2^{(d^\perp - 1)}$ times each. Since $w'_{d'} \neq 0$, there must be a codeword $\mathbf{u}$ with $\mathrm{wt}(\mathbf{u}) = d^\perp$ such that $\chi_{\mathbf{u}}(C) \neq 0$. Thus, the statement of the lemma follows. □

The following lemma establishes the almost sure convergence of the first $[(d^\perp - 1)/2]$ moments of $\mu_{\mathcal{C}_n}$ to those of μ_{MP}, as $n \to \infty$:

Lemma 7.3.3. *Let $\mu_{\mathcal{C}_n}$ denote the spectral measure of the Gram matrix $\mathcal{G}_{\mathcal{C}_n}$ corresponding to a matrix $\mathbf{\Phi}_{\mathcal{C}_n}$ based on a binary linear block code $\mathcal{C}_n$. Let*

$$m_{\mathcal{C}_n}^{(\ell)} := \int z^\ell \mu_{\mathcal{C}_n}(dz), \qquad m_{MP}^{(\ell)} := \int z^\ell \mu_{MP}(dz) \tag{7.15}$$

be the ℓth moment of the spectral measures $\mu_{\mathcal{C}}$ and μ_{MP}, respectively. Then, for $\ell = 1, 2, \cdots, [(d_n^{\perp} - 1)/2]$, as $n \to \infty$ we have

$$m_{\mathcal{C}_n}^{(\ell)} \to m_{MP}^{(\ell)} = \sum_{i=0}^{\ell-1} \frac{y^i}{i+1} \binom{\ell}{i} \binom{\ell-1}{i} \tag{7.16}$$

almost surely, where $y := p/n$.

Proof. Let $P_{\mathcal{C}_n}$ be the probability measure induced by the i.i.d. selection of p codewords from the $[n, k_n, d_n]$ binary linear block code $\mathcal{C}_n$. From the Borel-Cantelli Lemma (See, for example, [5] or [120]), it is enough to show:

$$\mathbb{E}_{P_{\mathcal{C}_n}} \left\{ m_{\mathcal{C}_n}^{(\ell)} \right\} \to m_{MP}^{(\ell)} = \sum_{i=0}^{\ell-1} \frac{y^i}{i+1} \binom{\ell}{i} \binom{\ell-1}{i} \tag{7.17}$$

and

$$\mathbb{E}_{P_{\mathcal{C}_n}} \left\{ \left(m_{\mathcal{C}_n}^{(\ell)} \right)^2 - \left(\mathbb{E}_{P_{\mathcal{C}_n}} \left\{ m_{\mathcal{C}_n}^{(\ell)} \right\} \right)^2 \right\} = \mathcal{O}\left(\frac{1}{n^2} \right) \tag{7.18}$$

where $\mathbb{E}_{P_{\mathcal{C}_n}}$ is the expectation with respect to $P_{\mathcal{C}_n}$.

Thus we need to prove that the average of the first $d_n^{\perp}$ moments of the measure $\mu_{\mathcal{C}_n}$ coincide with those of the Marchenko-Pastur density almost surely and that the variance of these moments of $\mu_{\mathcal{C}_n}$ drops as $1/n^2$. In what follows, we drop the subscript n for notational convenience. The ℓth moment of $\mu_{\mathcal{C}}$ can be written as

$$\begin{aligned} \mathbb{E}_{P_{\mathcal{C}}} \left\{ m_{\mathcal{C}}^{(\ell)} \right\} &= \mathbb{E}_{P_{\mathcal{C}}} \left\{ \int z^{\ell} \mu_{\mathcal{C}}(dz) \right\} = \mathbb{E}_{P_{\mathcal{C}}} \left\{ \frac{1}{p} \sum_{i=1}^{p} \lambda_i^{\ell} \right\} \\ &= \frac{1}{pn^{\ell}} \mathbb{E}_{P_{\mathcal{C}}} \left\{ \mathrm{Tr} \left\{ (\mathbf{\Phi}\mathbf{\Phi}^T)^{\ell} \right\} \right\} \\ &= \frac{1}{pn^{\ell}} \sum_{\mathcal{I},\mathcal{J}} \mathbb{E}_{P_{\mathcal{C}}} \left\{ (-1)^{s_{\mathcal{I},\mathcal{J}}} \right\}, \end{aligned} \tag{7.19}$$

where

$$\mathcal{I} := \{i_t\}_{t=1}^{\ell} \in \{1, 2, \cdots, p\}^{\ell}, \qquad \mathcal{J} := \{j_t\}_{t=1}^{\ell} \in \{1, 2, \cdots, n\}^{\ell}, \tag{7.20}$$

$$s_{\mathcal{I},\mathcal{J}} := c_{i_1 j_1} \oplus c_{i_2 j_1} \oplus c_{i_2 j_2} \oplus c_{i_3 j_2} \oplus \cdots \oplus c_{i_\ell j_\ell} \oplus c_{i_1 j_\ell}, \tag{7.21}$$

with $\oplus$ denoting the binary addition and the summation running over all $\mathcal{I} \in \{1, 2, \cdots, p\}^\ell$ and $\mathcal{J} \in \{1, 2, \cdots, n\}^\ell$. Note that $s_{\mathcal{I},\mathcal{J}}$ corresponds to a directed cycle of length 2ℓ on a complete bipartite graph $G = (X \cup Y, E)$ with $X := \{1, 2, \cdots, p\}$ and $Y := \{1, 2, \cdots, n\}$, where c_{ij} corresponds to an edge from node $i \in \mathcal{I}$ to node $j \in \mathcal{J}$.

First, mimicking the proofs of Wigner [128] and Marchencko and Pastur [90], we argue that the only cycles which contribute to the above summation are those in which every edge appears at least twice. To observe this, consider two index sets $\mathcal{I}$ and $\mathcal{J}$. Pick $i_1 \in \mathcal{I}$ (without loss of generality) and let $\{c_{i_1 j_t}\}_{t=1}^m$ denote the components of $\mathbf{c}_{i_1}$ appearing exactly once in the sum $s_{\mathcal{I},\mathcal{J}} := c_{i_1 j_1} \oplus c_{i_2 j_1} \oplus c_{i_2 j_2} \oplus c_{i_3 j_2} \oplus \cdots \oplus c_{i_\ell j_\ell} \oplus c_{i_1 j_\ell}$. Clearly, $m \leq [(d^\perp - 1)/2] < d^\perp$. From Lemma 7.3.2 we conclude that the sum $\oplus_{t=1}^m c_{i_1 j_t}$ takes the values of 1 and 0 equally many times, with respect to the probability measure $P_{\mathcal{C}}$. Therefore, the average of $(-1)^{s_{\mathcal{I},\mathcal{J}}}$ will be zero.

Thus, the only cycles that contribute to the expectation are those in which every edge appears at least twice. Moreover, if $s := |\mathcal{I}| + |\mathcal{J}| < \ell$, then the contribution of such cycles drops at least as fast as $1/pn^{\ell - s}$, since there are at most n^s such cycles. Hence, such cycles can be ignored as $n \to \infty$. Hence, the only contributing cycles are those whose skeleton is a tree and whose number is given by the right hand side of Eq. (7.17) [5], [90].

Proving the second statement is very similar to the case of Wigner [5]. The

variance of the moments can be written as:

$$\mathbb{E}_{P_{\mathcal{C}}}\left\{\left(m_{\mathcal{C}}^{(\ell)}\right)^2-\left(\mathbb{E}_{P_{\mathcal{C}}}\left\{m_{\mathcal{C}}^{(\ell)}\right\}\right)^2\right\}$$
$$=\frac{1}{p^2n^{2\ell}}\sum_{\mathcal{I},\mathcal{I}',\mathcal{J},\mathcal{J}'}\mathbb{E}_{P_{\mathcal{C}}}\left\{(-1)^{s_{\mathcal{I},\mathcal{J}}\oplus s_{\mathcal{I}',\mathcal{J}'}}\right\}-\mathbb{E}_{P_{\mathcal{C}}}\left\{(-1)^{s_{\mathcal{I},\mathcal{J}}}\right\}\mathbb{E}_{P_{\mathcal{C}}}\left\{(-1)^{s_{\mathcal{I}',\mathcal{J}'}}\right\} \quad (7.22)$$

If $\mathcal{I}\cap\mathcal{I}'=\mathcal{J}\cap\mathcal{J}'=\emptyset$, there is no contribution to the variance. Also, there can be at most $m\leq 2[(d^{\perp}-1)/2]<d^{\perp}$ elements of a codeword, say $\mathbf{c}_1$, appearing exactly once in the sum $s_{\mathcal{I},\mathcal{J}}\oplus s_{\mathcal{I}',\mathcal{J}'}$. Similar to the previous case, the contribution of such terms vanishes with respect to the probability distribution $P_{\mathcal{C}}$. Finally, it can be argued that ([5], [120]) the only loops that contribute have $|\mathcal{I}\cup\mathcal{I}'|+|\mathcal{J}\cup\mathcal{J}'|\leq 2\ell$, and the contribution of these terms in Eq. (7.22) drops as $\mathcal{O}(1/n^2)$, since there at at most $n^{2\ell}$ such loops. This completes the proof of the Lemma. □

Finally, the following lemma gives an upper bound on the moments of the Marchenko-Pastur density:

Lemma 7.3.4. *For the ℓth moment of the Marchenko-Pastur density $m_{MP}^{(\ell)}$, we have:*

$$m_{MP}^{(\ell)}\leq\frac{1}{\alpha_\ell\sqrt{2\pi\alpha_\ell(1-\alpha_\ell)\ell}}\left(y^{\alpha_\ell}4^{H(\alpha_\ell)}\right)^{\ell} \quad (7.23)$$

where

$$\alpha_\ell:=\frac{1}{\ell}\left\lfloor\frac{\sqrt{(2\ell y+y+1)^2+4\ell(\ell+1)y(1-y)}-(2\ell y+y+1)}{2(1-y)}\right\rfloor$$

and $H(z):=-z\log_2 z-(1-z)\log_2(1-z)$ is the binary entropy function.

Proof. We know that

$$m_{MP}^{(\ell)}=\sum_{i=0}^{\ell-1}\frac{y^i}{i+1}\binom{\ell}{i}\binom{\ell-1}{i} \quad (7.24)$$

for $y \in (0,1)$. The ratio of the ith and $(i-1)$th terms in the above summation is the following:

$$y\frac{(\ell-i+1)(\ell-i)}{i(i+1)} \tag{7.25}$$

The largest integer i for which this ratio stays greater or equal to 1, namely i^*, corresponds to the maximum summand in the above summation. Solving for i^* yields:

$$i^* = \left\lfloor \frac{\sqrt{(2\ell y+y+1)^2+4\ell(\ell+1)y(1-y)}-(2\ell y+y+1)}{2(1-y)} \right\rfloor \tag{7.26}$$

Hence, the ℓth moment can be upper bounded as

$$m_{MP}^{(\ell)} \leq \frac{\ell y^{i^*}}{i^*+1}\binom{\ell}{i^*}\binom{\ell-1}{i^*} \tag{7.27}$$

We can use the following version of the Stirling's bounds on $n!$,

$$\left(\frac{n}{e}\right)^n\sqrt{2\pi n}\exp\left(\frac{1}{12n+1}\right) \leq n! \leq \left(\frac{n}{e}\right)^n\sqrt{2\pi n}\exp\left(\frac{1}{12n}\right) \tag{7.28}$$

in order to simplify the upper bound on $m_{MP}^{(\ell)}$. Letting $\alpha_\ell := i^*/\ell$, we get

$$m_{MP}^{(\ell)} \leq \frac{1}{\alpha_\ell\sqrt{2\pi\alpha_\ell(1-\alpha_\ell)\ell}}\left(y^{\alpha_\ell}4^{H(\alpha_\ell)}\right)^\ell \exp\left(\frac{1}{12\ell}-\frac{1}{12(1-\alpha_\ell)\ell+1}-\frac{1}{12\alpha_\ell\ell+1}\right) \tag{7.29}$$

Noting that the argument of the exponential is always negative, the statement of the lemma follows. □

We now have all the ingredients for the proof of the Main Theorem:

Proof of the Main Theorem: We present a proof along the lines of Sidel'nikov's proof [112]. First of all, note that $\mathrm{Tr}(\mathcal{G}_{\mathcal{C}_n}) = p$. Therefore, the first moment of the spectral measure $\mu_{\mathcal{C}}$ is equal to 1. Hence, the spectral measure of $\mathcal{G}_{\mathcal{C}_n} - \mathbb{I}_{p\times p}$ has

vanishing first moment. Clearly, the same statement holds for the Marchenko-Pastur measure μ_{MP}, i.e., the density

$$g(z) = \frac{1}{2\pi(z+1)y}\sqrt{(b-z-1)(z+1-a)}1_{(a-1\leq z\leq b-1)} \tag{7.30}$$

has vanishing first moment. Now, let F and G in Lemma 7.3.1 be the distributions corresponding to the spectral measure of $\mathcal{G}_{\mathcal{C}_n} - \mathbb{I}_{p\times p}$ and $g(z)$, respectively. Moreover, it is easy to see that the density $g(z)$ achieves its maximum for $z = -y > (1-\sqrt{y})^2 - 1$ and hence $g(z)$ is upper bounded by

$$g(z) \leq g(-y) = \frac{1}{\pi\sqrt{2y(1-y)}}. \tag{7.31}$$

Let ϕ and γ denote the characteristic functions of the spectral measure of $\mathcal{G}_{\mathcal{C}_n}$ and μ_{MP}, respectively. More explicitly,

$$\phi(t) := \frac{1}{p}\sum_{k=1}^{p}\exp\left(it\lambda_k\right) \tag{7.32}$$

We now invoke Lemma 7.3.1 to bound the difference of the distributions $M_{\mathcal{C}_n}(z)$ and $M_{MP}(z)$ as follows:

$$\left|M_{\mathcal{C}_n}(z) - M_{MP}(z)\right| \leq \frac{1}{\pi}\int_{-T}^{T}\left|\frac{\phi(t)e^{-it} - \gamma(t)e^{-it}}{t}\right|dt + \frac{24}{\pi^2\sqrt{2y(1-y)}T} \tag{7.33}$$

for some fixed T. Let r_n be the greatest even integer less than or equal to $[(d_n^{\perp}-1)/2]$. Let r be a fixed number so that $r \leq r_n$ for all $n > N$, for some fixed N. Using the inequality

$$\left|\exp(it) - 1 - \frac{it}{1!} - \cdots - \frac{(it)^{r-1}}{(r-1)!}\right| \leq \frac{|t|^r}{r!} \tag{7.34}$$

which can easily be verified by induction on r, we have the following bound on the tail of the characteristic function $\phi(t)$:

$$\left|\phi(t) - \sum_{\ell=0}^{r-1} m_{\mathcal{C}_n}^{(\ell)}\frac{(it)^\ell}{\ell!}\right| \leq m_{\mathcal{C}_n}^{(r)}\frac{|t|^r}{r!} \tag{7.35}$$

where $m_{\mathcal{C}_n}^{(\ell)}$ is the ℓth moment of the spectral measure of $\mathcal{G}_{\mathcal{C}_n}$ given by

$$m_{\mathcal{C}_n}^{(\ell)} := \frac{1}{p}\sum_{k=1}^{p}\lambda_k^{\ell} \tag{7.36}$$

From Lemma 7.3.3 we know that $m_{\mathcal{C}_n}^{(\ell)} \to m_{MP}^{(\ell)}$ almost surely as $n \to \infty$, for $\ell = 1, 2, \cdots, r$. Hence, Eq. (7.35) can be written as

$$\limsup_n \left|\phi(t) - \sum_{\ell=0}^{r-1} m_{\mathcal{C}_n}^{(\ell)}\frac{(it)^{\ell}}{\ell!}\right| \leq m_{MP}^{(r)}\frac{|t|^r}{r!}. \tag{7.37}$$

Using the above equation, the integral in Eq. (7.33) can be upper bounded as

$$\begin{aligned}
&\limsup_n \frac{1}{\pi}\int_{-T}^{T}\left|\frac{\phi(t)-\gamma(t)}{t}\right|dt \\
&\quad\leq \limsup_n \frac{1}{\pi}\int_{-T}^{T}\frac{1}{|t|}\left|\sum_{\ell=0}^{r-1} m_{\mathcal{C}_n}^{(\ell)}\frac{(it)^{\ell}}{\ell!} + \sum_{\ell=r}^{\infty} m_{MP}^{(\ell)}\frac{(it)^{(\ell)}}{\ell!} - \gamma(t)\right|dt \\
&\qquad + \frac{2}{\pi}\int_{-T}^{T} m_{MP}^{(r)}\frac{|t|^{r-1}}{r!}dt.
\end{aligned} \tag{7.38}$$

Now, consider the first integral on the right hand side of the above inequality. Let

$$\zeta_n(t) := \frac{1}{|t|}\left|\sum_{\ell=0}^{r-1} m_{\mathcal{C}_n}^{(\ell)}\frac{(it)^{\ell}}{\ell!} + \sum_{\ell=r}^{\infty} m_{MP}^{(\ell)}\frac{(it)^{(\ell)}}{\ell!} - \gamma(t)\right| \tag{7.39}$$

Clearly,

$$\zeta_n(t) \leq \xi_n(t) := \sum_{\ell=1}^{r-1}\left|m_{\mathcal{C}_n}^{(\ell)} - m_{MP}^{(\ell)}\right|\frac{|t|^{\ell-1}}{\ell!}. \tag{7.40}$$

We will first show that $\xi_n(t)$ is integrable over $[-T, T]$ for all n. We have:

$$\begin{aligned}
\int_{-T}^{T}\xi_n(t)dt &= 2\sum_{\ell=1}^{r-1}\left|m_{\mathcal{C}_n}^{(\ell)} - m_{MP}^{(\ell)}\right|\frac{T^{\ell}}{\ell\cdot\ell!} \\
&\leq 2\Big(\sum_{\ell=1}^{r-1}\left|m_{\mathcal{C}_n}^{(\ell)} - m_{MP}^{(\ell)}\right|\Big)\max_{1\leq\ell\leq r-1}\frac{T^{\ell}}{\ell\cdot\ell!}.
\end{aligned} \tag{7.41}$$

Using the bound of

$$\frac{T^{\ell}}{\ell\cdot\ell!} < e^T, \tag{7.42}$$

we get:

$$\int_{-T}^{T} \xi_n(t)dt \;\; \leq \;\; 2re^T\Big(\frac{1}{r}\sum_{\ell=1}^{r-1}\Big|m_{\mathcal{C}_n}^{(\ell)} - m_{MP}^{(\ell)}\Big|\Big). \tag{7.43}$$

By the Chebyshev's inequality and Eq. (7.18), it is easy to see that for all $\epsilon > 0$,

$$\mathbb{P}\bigg(\Big|m_{\mathcal{C}_n}^{(\ell)} - m_{MP}^{(\ell)}\Big| > \epsilon\bigg) \leq \mathcal{O}\Big(\frac{1}{n^2}\Big) \tag{7.44}$$

for all ℓ. Hence, the right hand side of Eq. (7.43) is bounded almost surely for some large n, and thus $\xi_n(t)$ is integrable almost surely. Next, by applying the Fatou's Lemma [105, p. 23] to the non-negative sequence $\xi_n(t) - \zeta_n(t)$, we get:

$$\limsup_n \int_{-T}^{T} \zeta_n(t)dt \leq \int_{-T}^{T} \limsup_n \zeta_n(t)dt = 0 \tag{7.45}$$

almost surely. Thus, we only need to consider the second integral on the right hand side of Eq. (7.38), which is equal to

$$\frac{4}{\pi r} m_{MP}^{(r)} \frac{T^r}{r!}. \tag{7.46}$$

Hence, we obtain the following asymptotic almost sure bound:

$$\limsup_n \Big|M_{\mathcal{C}_n}(z) - M_{MP}(z)\Big| \leq \left\{\frac{4}{\pi r} m_{MP}^{(r)} \frac{T^r}{r!} + \frac{24}{\pi^2\sqrt{2y(1-y)}T}\right\} \tag{7.47}$$

Invoking Lemma 7.3.4, we have:

$$m_{MP}^{(r)} \leq \frac{1}{\alpha_r\sqrt{2\pi\alpha_r(1-\alpha_r)r}}\Big(y^{\alpha_r}4^{H(\alpha_r)}\Big)^r \tag{7.48}$$

where α_r is as defined in the statement of Lemma 7.3.4. Thus, the bound on the right hand side of Eq. (7.47) simplifies to

$$\frac{4}{\pi\alpha_r\sqrt{2\pi\alpha_r(1-\alpha_r)r}} \frac{\Big(y^{\alpha_r}4^{H(\alpha_r)}\Big)^r T^r}{r \cdot r!} + \frac{24}{\pi^2\sqrt{2y(1-y)}T} \tag{7.49}$$

Using the Stirling's approximation to lower bound $r!$ and balancing the two terms on the right hand side of the above equation with respect to T yields:

$$T = \left\{ \frac{12\alpha_r \sqrt{\alpha_r(1-\alpha_r)}}{\sqrt{2y(1-y)}} \right\}^{\frac{1}{r+1}} \frac{r}{\left(y^{\alpha_r} 4^{H(\alpha_r)} e\right)^{\frac{r}{r+1}}} \tag{7.50}$$

Finally, substituting the above choice of T in Eq. (7.49) yields:

$$\limsup_n \left| M_{\mathcal{C}_n}(z) - M_{MP}(z) \right| \leq \frac{24\left(\frac{y^{\alpha_r} 4^{H(\alpha_r)} e}{\sqrt{2y(1-y)}} \right)^{\frac{r}{r+1}}}{\pi^2 \left\{ 12\alpha_r \sqrt{\alpha_r(1-\alpha_r)} \right\}^{\frac{1}{r+1}}} \left(\frac{1}{r} + \frac{1}{r^2} \right) \tag{7.51}$$

Note that since the choice of N was arbitrary, the above bound particularly holds for $r = \liminf_n r_n$, as $n \to \infty$. Letting

$$c(y,r) := \frac{24\left(\frac{y^{\alpha_r} 4^{H(\alpha_r)} e}{\sqrt{2y(1-y)}} \right)^{\frac{r}{r+1}}}{\pi^2 \left\{ 12\alpha_r \sqrt{\alpha_r(1-\alpha_r)} \right\}^{\frac{1}{r+1}}}, \tag{7.52}$$

with $r = \liminf_n r_n$, we get:

$$\limsup_n \left| M_{\mathcal{C}_n}(z) - M_{MP}(z) \right| \leq c(y,r) \left(\frac{1}{r} + \frac{1}{r^2} \right), \tag{7.53}$$

which establishes the claim of the Main Theorem.

7.4 Group Randomness of Pseudo-Noise and Gold Sequences

We first briefly overview the definition, construction and pseudo-random properties of the shortened first-order Reed-Muller codes, as well as those of Gold sequences.

7.4.1 Definition and randomness properties of shortened first-order Reed-Muller codes

First-order Reed-Muller codes, denoted by $\mathscr{R}(1,m)$, are a family of $[2^m, m+1, 2^{m-1}]$ binary linear block codes. The dual of $\mathscr{R}(1,m)$ is the $\mathscr{R}(m-2,m)$ Reed-Muller code, which is a $[2^m, 2^m-m-1, 4]$ binary linear block code [86].

Let $n=2^m$, and let $\mathbf{c}=(c_1,c_2,\cdots,c_n)$ be a codeword of $\mathscr{R}(1,m)$. The shortened first-order Reed-Muller code can be obtained by shortening the $\mathscr{R}(1,m)$ code via taking a cross-section [86]. That is, taking all the codewords of $\mathscr{R}(1,m)$ which begin with $c_1=0$ and deleting the c_1 coordinate. Thus, the shortened first-order Reed-Muller code is a $[2^m-1, m, 2^{m-1}]$ code, which we refer to as the Simplex code $\mathscr{S}_m$.

Let $h(x)=x^m+h_{m-1}x^{m-1}+\cdots+h_1x+1$ be a primitive irreducible polynomial of degree m over GF(2). It can be shown that $h(x)$ is the check polynomial of $\mathscr{S}_m$ [86]. Hence, the codewords of $\mathscr{S}_m$ can be generated by the feedback shift register realization of the polynomial $h(x)$.

Next, we briefly review some of the pseudo-randomness properties of the codewords of $\mathscr{S}_m$ (See [86] for a detailed discussion and proof of these properties):

1) In any codeword (except for **1** and **0**), there are 2^{m-1} ones and $2^{m-1}-1$ zeros.

Let a *run* denote the maximal string of consecutive identical symbols in a string. Then, we have:

2) In any codeword (except for **1** and **0**), half of the runs have length 1, one quarter have length 2, one eighth have length 3, and so on, as long as these fractions

give integral numbers of runs. In each case, the number of runs of 0's is equal to the number of runs of 1's.

Let $\rho(\tau)$ be the auto-correlation function of a codeword $\mathbf{c}$ defined by

$$\rho(\tau) := \frac{1}{n}\sum_{j=1}^{n-1}(-1)^{c_j+c_{j+\tau}} \tag{7.54}$$

where $j+\tau$ is interpreted modulo n. Then, we have:

3) The auto-correlation function of any codeword of $\mathscr{S}_m$ (except for $\mathbf{1}$ and $\mathbf{0}$) is given by:

$$\rho(\tau) = \begin{cases} 1 & \text{for } \tau = 0. \\ -\frac{1}{n} & \text{for } 1 \leq \tau \leq 2^m - 2. \end{cases} \tag{7.55}$$

7.4.2 Definition and randomness properties of Gold sequences

Gold sequences are a class of pseudo-random sequences which can be obtained by XOR-ing the shifted versions of two PN sequences generated by two distinct primitive polynomials [59]. Let $h_1(x)$ and $h_2(x)$ be two primitive polynomials of degree m over GF(2), such that $h_1(\alpha) = 0$ and $h_2(\alpha^\delta) = 0$ for some integer δ. Suppose that $m \neq 0 \mod 4$. If $\delta = 2^h+1$ or $\delta = 2^{2h}-2^h+1$, and m/e is odd, with $e := gcd(m,h)$, then the two polynomials $h_1(x)$ and $h_2(x)$ are denoted by the preferred pair of polynomials. Let $\mathbf{u}$ and $\mathbf{v}$ denote two PN sequences of length 2^m-1, corresponding to the preferred pair of polynomials $h_1(x)$ and $h_2(x)$, respectively. Then, the set of Gold sequences $\mathscr{G}(\mathbf{u},\mathbf{v})$ is defined as:

$$\mathscr{G}(\mathbf{u},\mathbf{v}) := \{\Re^a\mathbf{u}, \Re^b\mathbf{v}, \text{ or } \Re^a\mathbf{u} \oplus \Re^b\mathbf{v} | 0 \leq a, b \leq 2^m - 2\}$$

where $\oplus$ and $\Re$ denote the binary XOR and cyclic shift operators, respectively. The set $\mathscr{G}(\mathbf{u}, \mathbf{v})$ consists of $2^{2m} - 1$ binary sequences of length $2^m - 1$, with desirable cross-correlation properties.

Let $\mathbf{g}_1, \mathbf{g}_2 \in \mathscr{G}(\mathbf{u}, \mathbf{v})$. Then, it can be shown that [59] the cross-correlation between $\mathbf{g}_1$ and $\mathbf{g}_2$ is either equal to -1, $-t(m)$ or $t(m) - 2$, where

$$t(m) := \begin{cases} 1 + 2^{(m+1)/2} & m \text{ odd} \\ \\ 1 + 2^{(m+2)/2} & m \text{ even} \end{cases} \tag{7.56}$$

Similarly, the auto-correlation of each Gold sequence is a three-valued function taking the values -1, $-t(m)$ or $t(m) - 2$. It is also known that only $(2^m - 1)(2^{m-1} + 1)$ of the Gold sequences are balanced (for m odd), *i.e.*, contain 2^{m-1} ones and $2^{m-1} - 1$ zeros [59]. Note that $\mathscr{G}(\mathbf{u}, \mathbf{v}) \cup \{\mathbf{0}\}$ is a $[2^m - 1, 2m]$ cyclic code [59], which we refer to as $\mathscr{G}_m$.

7.4.3 Spectral distribution of random matrices from shortened first-order Reed-Muller codes

We consider random matrices from shortened first-order Reed-Muller (Simplex) codes $\mathscr{S}_m$. It is easy to verify that the dual code of the Simplex code $\mathscr{S}_m$ is the $[2^m - 1, 2^m - m - 1, 3]$ Hamming code. Hence, the dual distance is not necessarily large enough for the Main Theorem of [14] to give a meaningful bound. Indeed, the following proposition establishes that the spectral distribution of a random matrix from $\mathscr{S}_m$ is very different from the Marchenko-Pastur distribution [9]:

Proposition 7.4.1. *Let $\mathbf{\Phi}_{\mathscr{S}}$ be a $p \times n$ random matrix based on the shortened first-order Reed-Muller code $\mathscr{S}_m$ (excluding the $\mathbf{1}$ and $\mathbf{0}$ codewords), where $n := 2^m - 1$ and $p < n$. Suppose that all the rows of $\mathbf{\Phi}_{\mathscr{S}}$ are distinct, i.e., $\mathbf{\Phi}_{\mathscr{S}}$ is full-rank. Finally, let $y := p/n$. Then, the eigen-values of $\frac{1}{n}\mathbf{\Phi}_{\mathscr{S}}\mathbf{\Phi}_{\mathscr{S}}^T$ are given by*

$$1 - y + \frac{1}{n}, \underbrace{1 + \frac{1}{n}, 1 + \frac{1}{n}, \cdots, 1 + \frac{1}{n}}_{p-1}.$$

Proof. Let $\mathcal{G} := \frac{1}{n}\mathbf{\Phi}_{\mathscr{S}}\mathbf{\Phi}_{\mathscr{S}}^T$. Then, the elements of $\mathcal{G}$ are given by:

$$\mathcal{G}_{ij} = \begin{cases} 1 & i = j \\ -\frac{1}{n} & i \neq j \end{cases} \tag{7.57}$$

To see this, we first note that for $i = j$, $\mathcal{G}_{ii}$ is clearly equal to 1 doe to the normalization by $1/n$. Next, for $i \neq j$, we have $\mathcal{G}_{ij} = \frac{1}{n}\sum_{k=1}^{n}(-1)^{c_{ik}+c_{jk}}$, where $\mathbf{c}_i$ and $\mathbf{c}_j$ are the ith and jth rows of $\mathbf{\Phi}_{\mathscr{S}}$. It is easy to see that

$$\sum_{k=1}^{n}(-1)^{c_{ik}+c_{jk}} = n - 2d_H(\mathbf{c}_i, \mathbf{c}_j) \tag{7.58}$$

where $d_H(\cdot, \cdot)$ denotes the Hamming distance. Also, for the simplex code $\mathscr{S}_m$, we have $d_H(\mathbf{c}_i, \mathbf{c}_j) = 2^{m-1}$. Substituting the later in Eq. (7.58) gives Eq. (7.57). Next, we write $\mathcal{G}$ as

$$\mathcal{G} = \frac{n+1}{n}\mathbb{I}_{p\times p} - \frac{1}{n}\mathbf{1}_{p\times p} \tag{7.59}$$

where $\mathbf{1}_{p\times p}$ is the all ones $p \times p$ matrix. Since the matrix $\mathbf{1}_{p\times p}$ has only one non-zero eigen-value equal to p, the statement of the proposition follows from Eq. (7.59). □

Figures 7.6 and 7.7 show the empirical spectral distribution and density for a random matrix based on the $[511, 9, 256]$ simplex code $\mathscr{S}_9$ vs. Marchenko-Pastur law.

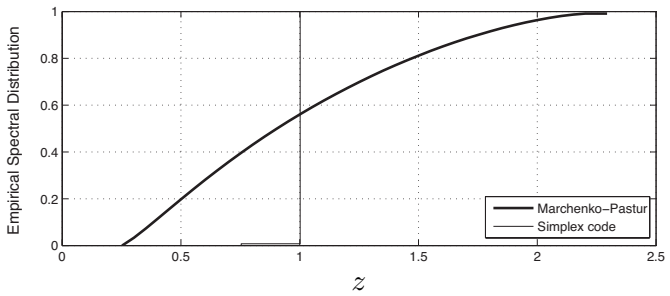


Figure 7.6: Empirical spectral distribution for a random matrix based on the $[511, 9, 256]$ Simplex code for $y = 127/511$.

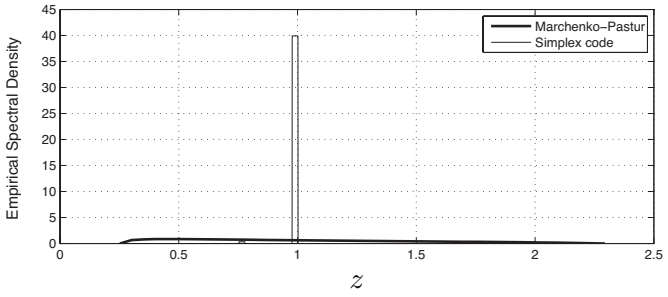


Figure 7.7: Empirical spectral density for a random matrix based on the $[511, 9, 256]$ Simplex code for $y = 127/511$.

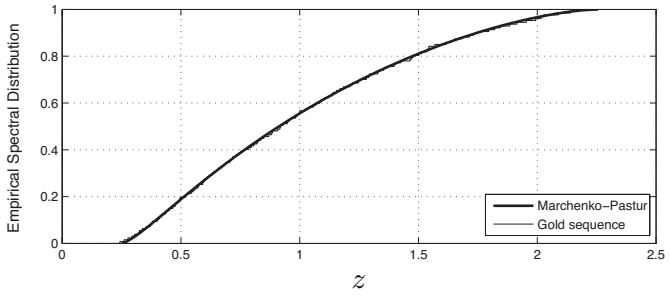


Figure 7.8: Empirical spectral distribution for a random matrix based on Gold sequences of length 511 generated by the preferred pair of polynomials $h_1(x) = x^9+x^4+1$ and $h_2(x) = x^9 + x^6 + x^4 + x^3 + 1$, for $y = 127/511$.

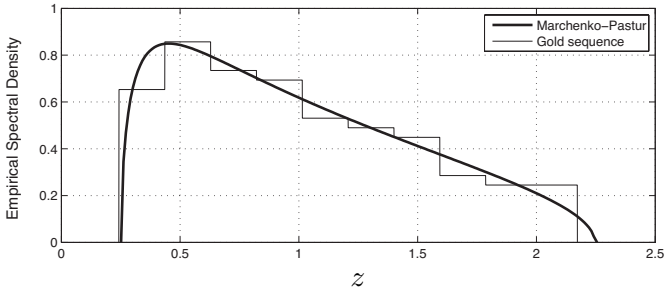


Figure 7.9: Empirical spectral density for a random matrix based on Gold sequences of length 511 generated by the preferred pair of polynomials $h_1(x) = x^9 + x^4 + 1$ and $h_2(x) = x^9 + x^6 + x^4 + x^3 + 1$, for $y = 127/511$.

As it can be observed from the figures, the distribution and density are dramatically different from the Marchenko-Pastur law. In fact, the Kolmogorov distance of the two distributions is about 0.55.

Figures 7.8 and 7.9 show the empirical spectral distribution and density for a random matrix based on Gold sequences of length 511, generated by the preferred pair of polynomials $h_1(x) = x^9+x^4+1$ and $h_2(x) = x^9+x^6+x^4+x^3+1$ vs. Marchenko-Pastur law. As it can be observed from the figures, the distribution and density are very similar to the Marchenko-Pastur law. By an application of the MacWilliams

identity [86], it can be shown that the dual distance of the Gold sequence of length 511 is 5. Though, the Gold sequences surprisingly behave like random i.i.d. ± 1 sequences, in the group randomness sense.

7.4.4 Kolmogorov complexity of shortened first-order Reed-Muller codes and Gold sequences

As another measure of randomness, one can consider the Kolmogorov complexity of pseudo-random sequences. Recall that the Kolmogorov complexity of an object is the minimum length of a program that generates that object and halts, with respect to a universal computer [40]. It can also be shown that the average Kolmogorov complexity of an i.i.d. binary sequence $\mathbf{x}$ of length n is lower bounded by $nH_0(p)$, where $p := \Pr(x_i = 1)$ and H_0 is the binary entropy function defined as $H_0(x) := -x \log_2 x - (1-x)\log_2(1-x)$.

Now, consider a codeword of the shortened first-order Reed-Muller code $\mathscr{S}_m$, generated by the primitive polynomial $h(x) = x^m + h_{m-1}x^{m-1} + \cdots + h_1 x + 1$. The following program can generate the ith codeword of of $\mathscr{S}_m$:

> Input the m-digit binary representation of i as the seed to the linear feedback shift register given by the characteristic equation $x_n = x_{n-1}h_{m-1} + x_{n-2}h_{n-2} + \cdots + x_{n-m+1}h_1 + x_{n-m}$. Print the output of length $2^m - 1$.

The Kolmogorov complexity of expressing the linear feedback shift register is clearly $m-1+c_1$, since $m-1$ bits are enough to store the coefficients of $h(x)$. Note that c_1 is a constant independent of i. Also, m bits are enough to store the binary representation

of the integer i. Hence, the overall Kolmogorov complexity is upper bounded by

$$\mathscr{K}(\mathbf{s} \in \mathscr{S}_m | n) \leq 2m - 1 + c_2 \tag{7.60}$$

where $c_2 > 0$ is a constant independent of i. Recall that the codewords of $\mathscr{S}_m$ (except for $\mathbf{1}$ and $\mathbf{0}$) have 2^{m-1} ones and $2^{m-1} - 1$ zeros. Therefore, the above upper bound implies that the Kolmogorov complexity of a codeword of $\mathscr{S}_m$ is much smaller than that of a random i.i.d. binary sequence of the same length with $p = \frac{2^{m-1}}{2^m - 1} \approx \frac{1}{2}$, for m large enough.

Similarly, the following program can generate a desired Gold sequence in $\mathscr{G}(\mathbf{u}, \mathbf{v})$, where $\mathbf{u}$ and $\mathbf{v}$ are generated by the preferred pair of polynomials $h(x) := x^m + h_{m-1}x^{m-1} + \cdots + h_1 x + 1$ and $g(x) := x^m + g_{m-1}x^{m-1} + \cdots + g_1 x + 1$:

> Generate $\mathbf{u}$ and $\mathbf{v}$ using the linear feedback shift registers based on $h(x)$ and $g(x)$, respectively. Generate $\Re^a\mathbf{v}$ and $\Re^b\mathbf{u}$ by cyclicly shifting $\mathbf{v}$ and $\mathbf{u}$ a total of a and b times, respectively. XOR $\Re^a\mathbf{u}$ and $\Re^b\mathbf{v}$. Print the output.

The Kolmogorov complexity of generating each of $\mathbf{v}$ and $\mathbf{u}$ is clearly bounded by $2m - 1 + c_2$, as discussed earlier. Also, $2m$ bits are enough to express a and b. Therefore, the overall Kolmogorov complexity is bounded by:

$$\mathscr{K}(\mathbf{g} \in \mathscr{G}(\mathbf{u}, \mathbf{v}) | n) \leq 6m - 2 + c_3 \tag{7.61}$$

where $c_3 > 0$ is a constant independent of k. Similarly, the Kolmogorov complexity of a Gold sequence is much smaller than that of a random i.i.d. binary sequence, for m large enough.

Chapter 8

Summary and Direction for Future Research

In this chapter, we summarize the contributions of this thesis, and discuss possible future research directions. The main contributions of this thesis are the following:

- We have considered the problem of estimating an L-sparse vector $\mathbf{x} \in \mathbb{C}^M$ from N noisy observations. We have constructed the joint typicality decoder, which is asymptotically unbiased and achieves the Cramér-Rao bound on the mean squared estimation error of the Genie-aided estimator without any knowledge of the locations of nonzero elements of $\mathbf{x}$, as $M \to \infty$ for $\alpha = L/N$ and $\beta = M/L$ fixed and $\sqrt[4]{L}(\min_i |x_i|)/\sqrt[4]{\log L} \to \infty$ as $M \to \infty$. This is a surprising result, since locations of the non-zero elements of the sparse vector $\mathbf{x}$ (asymptotically represented by $\log M$ information bits) are not known to the estimator, yet it achieves the performance of the Genie-aided estimator.

- We have obtained a universal sufficiency conditions for asymptotically reliable sparse recovery, which guarantees that for *any* sequence of L-sparse vectors $\{\mathbf{x} \in \mathbb{C}^M\}_M$ such that $\min_i |x_i| \geq \mu_0$ for some constant μ_0, and *any* sequence of $N \times M$ i.i.d. zero-mean Gaussian measurement matrices, it is possible to achieve asymptotically reliable sparse recovery with overwhelming probability as $M, N, L \to \infty$ with L/N and M/L fixed, if $N > CL$, for some constant C.

- We have developed a Recursive ℓ_1-Regularized Least Squares (SPARLS) algorithm for the estimation of a sparse tap-weight vector in the adaptive filtering setting. We have presented analytical results regarding the convergence, steady state error and parameter adjustments of the SPARLS algorithm. Simulation studies, in the context of multi-path wireless channel estimation, show that the SPARLS algorithm has significant improvement over the conventional widely-used Recursive Least Squares (RLS) algorithm in terms of mean squared error (MSE). Moreover, these simulation results suggest that the SPARLS algorithm has a lower computational complexity than the RLS algorithm, when the underlying tap-weight vector has a fixed support.

- We have developed a class of adaptive algorithms for sparse system identification. These adaptive algorithms combine the Expectation-Maximization framework and Kalman filtering. Simulation results on various linear and non-linear channels reveal significant performance gains in comparison to the conventional non-sparse methods.

- We have proposed an adaptive algorithm for sparse approximations (SpAdOMP)

with linear complexity, employing the underlying principles of existing batch-greedy algorithms. Analytical bounds on the steady-state MSE are obtained, which highlight the superior performance of the SpAdOMP algorithm. The proposed algorithm was applied to sparse NARMA identification and in particular to NARMA channel equalization/predistortion. Simulation results validated the superior performance of the SpAdOMP algorithm.

- We have studied the spectral distribution of random matrices from binary block codes. We have proved that if the underlying block code has a large dual distance, the empirical spectral distribution resembles the Marchenko-Pastur distribution corresponding to the spectral distribution of i.i.d. ± 1 random matrices. Not only this result is interesting from the viewpoint of random matrix theory, but also sheds light on the group randomness of pseudo-random sequences. Moreover. we have studied the group randomness and Kolmogorov complexity of two classes of pseudo-random sequences, namely, shortened first-order Reed-Muller codes and Gold sequences.

We next provide future research directions that are aimed to build on our previous work:

- The problem of asymptotically reliable support recovery of sparse signals seems to pose certain power constraints on the underlying signal. That is, asymptotically reliable sparse recovery is achieved when the signal to noise ratio (SNR) goes to infinity. An interesting research problem is to obtain sufficient conditions which possibly relax this requirement to constant SNR.

- It has been shown that the Cramér-Rao bound on the MSE of the Genie-aided estimator can be obtained by the joint-typicality decoder, which has exponential complexity in the dimension of the signal. Whether a polynomial time algorithm can provably achieve the Cramér-Rao bound is an open problem.

- The existing adaptive algorithms for sparse estimation have provable performance results for underlying signals with fixed support. Developing adaptive algorithms with provable support tracking abilities is very important and to the best of our knowledge remains open.

- We have proved that a sufficiency condition for the resemblance of the spectral distribution of random matrices from a binary block code to the Marchenko-Pastur law is having a large dual distance. However, simulation studies reveal that this is not a necessary condition. An interesting research problem is to strengthen this sufficient condition, and possibly obtain necessary conditions thereof.

- Gold sequences have very small dual distances (4 or 5, independent of their length). However, they manifest group randomness properties very similar to i.i.d. random sequences. We conjecture that the spectral distribution of random matrices from Gold sequences converges weakly to the Marchenko-Pastur law.

Bibliography

[1] M. Akçakaya, J. Park, and V. Tarokh. Compressive sensing using low density frames. arXiv:0903.0650v1 [cs.IT], code available at: http://www.people.fas.harvard.edu/∼akcakaya/suprem.html, March 2009.

[2] M. Akçakaya, J. Park, and V. Tarokh. Low density frames for compressive sensing. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 2010.

[3] M. Akçakaya and V. Tarokh. A frame construction and a universal distortion bound for sparse representations. *IEEE Transactions on Signal Processing*, 56(6):2443–2450, June 2008.

[4] M. Akçakaya and V. Tarokh. Shannon theoretic limits on noisy compressive sampling. *IEEE Transactions on Information Theory*, 56(1):492–504, January 2010.

[5] G. Anderson, A. Guionnet, and O. Zeitouni. *An Introduction to Random Matrices*. Cambridge University Press, 1st edition, Dec. 2009.

[6] D. Andrews and C. Mallows. Scale mixtures of normal distributions. *J. R. Stat. Soc.*, 36:99–102, 1974.

[7] D. Angelosante and G. Giannakis. RLS-weighted LASSO for adaptive estimation of sparse signals. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 2009.

[8] L. Applebaum, S. D. Howard, S. Searle, and A. R. Calderbank. Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery. *Appl. Comput. Harmon. Anal.*, 26:283290, 2009.

[9] B. Babadi, S. Ghassemzadeh, and V. Tarokh. Group randomness properties of pseudo-noise and gold sequences. In *Proc. of the 2011 Canadian Workshop on Information Theory*, 2011.

[10] B. Babadi, N. Kalouptsidis, and V. Tarokh. Comparison of SPARLS and RLS algorithms for adaptive filtering. In *Proceedings of IEEE Sarnoff Symposium*, March.

[11] B. Babadi, N. Kalouptsidis, and V. Tarokh. Asymptotic achievability of the cramér-rao bound for noisy compressive sampling. *IEEE Transactions on Signal Processing*, 57(3):1233 – 1236, March 2009.

[12] B. Babadi, N. Kalouptsidis, and V. Tarokh. SPARLS: The sparse RLS algorithm. *IEEE Transactions on Signal Processing*, 58(8):4013 – 4025, 2010.

[13] B. Babadi and V. Tarokh. Random frames from binary linear block codes.

In *Proceedings of the Annual Conference on Information Sciences and Systems (CISS'10)*, 2011.

[14] B. Babadi and V. Tarokh. Spectral distribution of random matrices from binary linear block codes. *IEEE Transactions on Information Theory*, 57(6):3955–3962, 2011.

[15] A.R.S. Bahai, B.R. Saltzberg, and M. Ergen. *Multi–Carrier Digital Communications –Theory and Applications of OFDM*. Kluwer Academic/Plenum, 2004.

[16] W. Bajwa, J. Haupt, G. Raz, and R. Nowak. Compressed channel sensing. In *Proceedings of the 42nd Annual Conference on Information Sciences and Systems (CISS'08)*, March 2008.

[17] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[18] D. Baron, S. Sarvotham, and R. G. Baraniuk. Bayesian compressive sensing via belief propagation. *IEEE Transactions on Signal Processing*, 58(1):269–280, January 2010.

[19] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183202, 2009.

[20] Z. Ben-Haim, Y. C. Eldar, and M. Elad. Coherence-based performance guar-

antees for estimating a sparse vector under random noise. *IEEE Transactions on Signal Processing*, 58(10):50305043, Oct. 2010.

[21] S. Benedetto and E. Biglieri. *Principles of Digital Transmission: with wireless applications*. Springer, 1999.

[22] J. Blanchard, C. Cartis, and J. Tanner. Compressed sensing: How sharp is the restricted isometry property? *submitted*.

[23] A. Bruckstein, D. Donoho, , and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):3481, 2009.

[24] R. Calderbank, S. Howard, and S. Jafarpour. Construction of a large class of deterministic matrices that satisfy a statistical isometry property. *IEEE Journal of Selected Topics in Signal Processing, special issue on Compressed Sensing*, pages 358 – 374, Apr. 2010.

[25] E. J. Candès and P. A. Randall. Highly robust error correction by convex programming. *IEEE Transactions on Information Theory*, 54(7):2829–2840, July 2008.

[26] E. J. Candès and J. Romberg. Practical signal recovery from random projections. In *Proc. Wavelet Appl. Signal Image Process. XI, SPIE Conf.*, 2005.

[27] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52:489–509, February 2006.

[28] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery for incomplete and inaccurate measurements. *Commun. Pure Appl. Math*, 59:1207–1223, August 2006.

[29] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, December 2005.

[30] E. J. Candès and T. Tao. The Dantzig selector: statistical estimation when p is much larger than n. *Annals of Statistics*, 35:2313–2351, December 2007.

[31] E.J. Candès, M.B. Wakin, and S. Boyd. Enhancing sparsity by reweighted l1 minimization. *J Fourier Anal. Appl.*, 14:877–905, 2008.

[32] G. Carayannis, D. Manolakis, and N. Kalouptsidis. A fast sequential algorithm for least-squares filtering and prediction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 31(6):1394–1402, 1983.

[33] A. Carmi, P. Gurfil, and D. Kanevsky. A simple method for sparse signal recovery from noisy observations using kalman filtering. *IBM Res. Report*, 2008.

[34] S. Chen and S.A. Billings. Representations of non-linear systems: the NARMAX model. *Int. J. Control*, 49(3):1013–1032, 1989.

[35] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput*, 20(1):33–61, 1998.

[36] Y. Chen, Y. Gu, and A. O. Hero III. Sparse LMS for system identification. In

Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, April.

[37] Kwan F Cheung and Robert J Marks. Imaging sampling below the nyquist density without aliasing. *JOSA A*, 7(1):92–105, 1990.

[38] Y. S. Chow and T. L. Lai. Limiting behavior of weighted sums of independent random variables. *Ann. Probab.*, 1(5):810–824, 1973.

[39] J. D. Cohen. Noncentral chi-square: Some observations on recurrence. *The American Statistician*, 42(2):120–122, May 1988.

[40] T. M. Cover and J. M. Thomas. *Elements Of Information Theory*. John Wiley & Sons, second edition, 2006.

[41] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. Inf. Theory*, 55(5):2230–2249, 2009.

[42] S. DasGupta and S. K. Sarkar. On TP_2 and log-concavity. *Lecture Notes-Monograph Series, Vol. 5, Inequalities in Statistics and Probability*, pages 54–58, 1984.

[43] I. Daubechies, M. Defrise, and C.D. Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57(11):1413–1457, 2004.

[44] S. Davis, G.M. Mallat and Z. Zhang. Adaptive time-frequency decompositions. *SPIE J. Opt. Engin.*, 33(7):2183–2191, 1994.

[45] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

[46] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

[47] D. L. Donoho, M. Elad, and V. N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, January 2006.

[48] D. L. Donoho, A. Maleki, and A. Montanari. Message-passing algorithms for compressed sensing. *Proc. of National Academy of Sciences (PNAS)*, 106:18914–18919, 2009.

[49] D. L. Donoho and J. Tanner. Counting faces of randomly projected polytopes when the projection radically lowers dimension. *J. Amer. Math. Soc.*, 22(1):1–53, 2009.

[50] D.L. Donoho, Y. Tsaig, I. Drori, and J.L. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. *Submitted for publication.*

[51] D.L. Duttweiler. Proportionate normalized least-mean-squares adaptation in echo cancellers. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 8(5):508518, 2000.

[52] E. Eweda and O. Macchi. Convergence of an adaptive linear estimation algorithm. *IEEE Trans. Autom. Control*, 29(2):119–127, 1984.

[53] M. Feder. *Statistical Signal Processing Using a Class of Iterative Estimation Algorithms*. PhD thesis, M.I.T., Cambridge, MA, 1987.

[54] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 2. Wiley, 2nd edition, 1991.

[55] M. Figueirado and R. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 12(8):906–916, August 2003.

[56] M. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: Applications to compressed sensing and other inverse problems. *IEEE Journal Of Selected Topics In Signal Processing*, 1(4), Dec. 2007.

[57] A. K. Fletcher, S. Rangan, and V. K. Goyal. Necessary and sufficient conditions on sparsity pattern recovery. *IEEE Transactions on Information Theory*, 55(12):57585772, December 2009.

[58] G.-O.A. Glentis, P. Koukoulas, and N. Kalouptsidis. Efficient algorithms for Volterra system identification. *IEEE Trans. Signal Process.*, 47(11):3042–3057, Nov. 1999.

[59] R. Gold. Maximal recursive sequences with 3-valued recursive cross-correlation functions (corresp.). *IEEE Transactions on Information Theory*, 14(1):154–156, 1968.

[60] S. W. Golomb. *Digital Communications with Space Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1964.

[61] S. Gurevich, R. Hadani, and N. Sochen. The finite harmonic oscillator and its applications to sequences, communication and radar. *IEEE Transactions on Information Theory*, 54(9), 2008.

[62] S. Gurevich, R. Hadani, and N. Sochen. The finite harmonic oscillator and its associated sequences. *Proceedings of the National Academy of Sciences of the United States of America*, 105(29):9869–9873, 2008.

[63] J. Haupt, W. U. Bajwa, G. Raz, and R. Nowak. Toeplitz compressed sensing matrices with applications to sparse channel estimation. *IEEE Transactions on Information Theory*, 56(11):5862–5875, Nov. 2010.

[64] J. Haupt and R. Nowak. Signal reconstruction from noisy random projections. *IEEE Transactions on Information Theory*, 52(9), Sep.

[65] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 3rd edition, 1996.

[66] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[67] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[68] S. D. Howard, A. R. Calderbank, and W. Moran. The finite heisenberg-weyl groups in radar and communications. *EURASIP Jounrnal on Applied Signal Processing*, 2006:1–12, 2006.

[69] S. D. Howard, A. R. Calderbank, and S. J. Searle. A fast reconstruction algortihm for deterministic compressive sensing using second order Reed-Muller codes. In *Conference on Information Sciences and Systems*, March 2008.

[70] P. Indyk. Explicit constructions for compressed sensing of sparse signals. In *SODA*, 2008.

[71] M. Isaksson, D. Wisell, and D. Ronnow. A comparative analysis of behavioral models for RF power amplifiers. *IEEE Trans. Microw. Theory Tech.*, 54(1):348–359, Jan. 2006.

[72] S. Jafarpour, W. Xu, B. Hassibi, and R. Calderbank. Efficient and robust compressed sensing using optimized expander graphs. *IEEE Transactions on Information Theory*, 55(9):4299–4308, 2009.

[73] W. C. Jakes, editor. *Microwave Mobile Communications*. New York: John Wiley & Sons Inc, 1975.

[74] A. Jazwinski. Adaptive filtering. *Automatica*, 5:475485, 1969.

[75] Robert E Kahn and Bede Liu. Sampling representations and the optimum reconstruction of signals. *Information Theory, IEEE Transactions on*, 11(3):339–347, 1965.

[76] N. Kalouptsidis. *Signal Processing Systems: Theory & Design*. Wiley, 1997.

[77] N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh. Adaptive algorithms for sparse nonlinear channel estimation. In *Proc. IEEE SSP*, pages 221–224, 2009.

[78] N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh. Adaptive algorithms for sparse system identification. *Signal Processing*, 91(8):1910–1919, August 2011.

[79] M. Kocic, D. Brady, and M. Stojanovic. Sparse equalization for real-time digital underwater acoustic communications. In *Proc. IEEE OCEANS*, pages 1417–1422, 1995.

[80] H. Ku and J.S. Kenney. Behavioral modeling of nonlinear RF power amplifiers considering memory effects. *IEEE Trans. Microw. Theory Tech.*, 51(12):2495–2504, Dec. 2003.

[81] H.J. Kushner. *Approximation and Weak Convergence Methods for Random Processes with Applications to Stochastic System Theory*. MIT Press, 1984.

[82] HJ Landau. Necessary density conditions for sampling and interpolation of certain entire functions. *Acta Mathematica*, 117(1):37–52, 1967.

[83] HJ Landau. Sampling, data transmission, and the nyquist rate. *Proceedings of the IEEE*, 55(10):1701–1706, 1967.

[84] B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, 28(5):1303–1338, October 2000.

[85] L. Ljung. *Adaptive system identification and signal processing algorithms, Eds. N. Kalouptsidis and S. Theodoridis*, chapter General Structure of Adaptive Algorithms: Adaptation and Tracking. Prentice Hall, Inc., 1993.

[86] F. J. Macwilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland Mathematical Library, 1988.

[87] A. Maleki and D. L. Donoho. Optimally tuned iterative reconstruction algorithms for compressed sensing. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):330–341, April 2010.

[88] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, 1993.

[89] E. B. Manoukian. *Mathematical Nonparametric Statistics*. CRC Press, 1st edition, 1986.

[90] V. A. Marchenko and L. A. Pastur. The distribution of eigenvalues in certain sets of random matrices. *Math. Sb.*, 72:507536, 1967.

[91] X. L. Meng and D. B. Rubin. On the global and componentwise rates of convergence of the em algorithm. *Linear Algebra Appl.*, 199:413–425, 1997.

[92] G. Mileounis, B. Babadi, N. Kalouptsidis, and V. Tarokh. An adaptive greedy algorithm with application to nonlinear communication. *IEEE Transactions on Signal Processing*, 58(6):2998 – 3007, June 2010.

[93] G. Mileounis, B. Babadi, N. Kalouptsidis, and V. Tarokh. An adaptive greedy algorithm with application to sparse NARMA identification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2010.

[94] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24:227–234, 1995.

[95] D. Needell and J.A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmon. Anal*, 26:301–321, 2009.

[96] D. Needell and R. Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Found. Comput. Math*, 9(3):317–334, 2009.

[97] D. Omidiran and M. J. Wainwright. High-dimensional subset recovery in noise: Sparsified measurements without loss of statistical efficiency. Technical report, UC Berkeley, Deparment of Statistics, May 2008.

[98] S. Pafka, M. Potters, and I. Kondor. Exponential weighting and random-matrix-theory-based filtering of financial covariance matrices for portfolio optimization. *preprint, (available at http://arxiv.org/abs/cond-mat/0402573)*, 2004.

[99] B. W. Parkinson and J. J. Spilker (eds.). *Global Positioning System: Theory & Applications*, volume I and II. AIAA, Washington, DC, 1996.

[100] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *27th Asilomar Conf.on Signals, Systems and Comput.*, pages 40–44, 1993.

[101] F. Ping and Y. Bresler. Spectrum-blind minimum-rate sampling and reconstruction of 2-d multiband signals. In *Image Processing, 1996. Proceedings., International Conference on*, volume 1, pages 701–704. IEEE, 1996.

[102] G. Reeves and M. Gastpar. Sampling bounds for sparse support recovery in the

presence of noise. In *IEEE International Symposium on Information Theory*, July 2008.

[103] C. Robert. *The Bayesian Choice: A Decision Theoretic Motivation*. Springer-Verlag, New York, NY, first edition, 1994.

[104] R. T. Rockafellar. *Convex Analysis*. Princeton Univ. Press, 1970.

[105] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, New York, NY, 3rd edition, 1987.

[106] Venkatesh Saligrama and Manqi Zhao. Thresholded basis pursuit: LP algorithm for order-wise optimal support recovery for sparse and approximately sparse signals from noisy random measurements. *Information Theory, IEEE Transactions on*, 57(3):1567–1586, 2011.

[107] S. Sarvotham, D. Baron, and R. Baraniuk. Measurements vs. bits: Compressed sensing meets information theory. In *Proc. Allerton Conference on Communication, Control, and Computing*, September 2006.

[108] A.H. Sayed. *Adaptive Filters*. Wiley-IEEE press, 2008.

[109] W.F. Schreiber. Advanced television systems for terrestrial broadcasting: Some problems and some proposed solutions. 83:958–981, 1995.

[110] P. K. Sen. The mean-median-mode inequality and noncentral chi square distributions. *Sankhya: The Indian Journal of Statistics, Series A*, 51(1):106–114, February 1989.

[111] A. M. Sengupta and P. P. Mitra. Distributions of singular values for some random matrices. *Phys. Rev. E*, 60(3), 1999.

[112] V. M. Sidel'nikov. Weight spectrum of binary bose-chaudhuri-hoquinghem codes. *Problems of Information Transmission*, 7(1):11–17, 1971.

[113] D.T.M. Slock and T. Kailath. Numerically stable fast transversal filters for recursive least squares adaptive filtering. *IEEE Trans. Signal Process.*, 39(1):92–114, Jan. 1991.

[114] R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. R. Statist. Soc. B*, 58(1):267–288, 1996.

[115] H. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I.* John Wiley and Sons Inc., first edition, 2001.

[116] J. Tropp. Just relax: Convex programming methods for identifying sparse signals. *IEEE Transactions on Information Theory*, 51(3):1030–1051, Mar. 2006.

[117] J. A. Tropp and A. C. Gilbert. Signal recovery from partial information via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, December 2007.

[118] J.A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory*, 50(10):2231–2242, 2004.

[119] P. P. Vaidyanathan and V. C. Liu. Efficient reconstruction of band-limited sequences from nonuniformly decimated versions by use of polyphase filter banks.

IEEE Transactions on Acoustics, Speech and Signal Processing, 38(11):1927–1936, 1990.

[120] B. Valkó. lecuture notes on random matrices. *available at http://www.math.wisc.edu/∼valko/courses/833/833.html*, 2009.

[121] E. van den Berg and M. P. Friedlander. Necessary and sufficient conditions on sparsity pattern recovery. *SIAM J. Sci. Comput.*, 31(2):890–912, 2008.

[122] N. Vaswani. Kalman filtered compressed sensing. In *Proc. IEEE ICIP*, pages 893–896, 2008.

[123] M. J. Wainwright. Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. *IEEE Transactions on Information Theory*, 55(12):5728–5741, December 2009.

[124] M. J. Wainwright. Sharp thresholds for noisy and high-dimensional recovery of sparsity using ℓ_1-constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, May 2009.

[125] W. Wang, M. J. Wainwright, and K. Ramchandran. Information-theoretic limits on sparse signal recovery: Dense versus sparse measurement matrices. *IEEE Transactions on Information Theory*, 56(6):2967–2979, June 2010.

[126] E. Weinstein, A.V. Oppenheim, M. Feder, and J.R. Buck. Iterative and sequential algorithms for multisensor signal enhancement. *IEEE Trans. Signal Process.*, 42(4):846–859, Apr. 1994.

[127] R. L. Wheeden and A. Zygmund. *Measure and Integral: An Introduction to Real Analysis*. New York: Dekker, 1977.

[128] E. Wigner. Characteristic vectors of bordered matrices with infinite dimensions. *The Annals of Mathematics, Second Series*, 62(3):548–564, Nov. 1955.

[129] J. Wishart. The generalised product moment distribution in samples from a normal multivariate population. *Biometrika*, 20A (1-2):3252, 1928.

[130] J. Wolfowitz. *Coding Theorems of Information Theory*. Springer-Verlag, third edition, 1978.