A COMPARISON OF SPARSE SIGNAL RECOVERY AND APPROXIMATE BAYESIAN INFERENCE METHODS FOR SPARSE CHANNEL ESTIMATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYLA UÇAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2015

Approval of the thesis:

A COMPARISON OF SPARSE SIGNAL RECOVERY AND APPROXIMATE BAYESIAN INFERENCE METHODS FOR SPARSE CHANNEL ESTIMATION

submitted by **AYLA UÇAR** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. Çağatay Candan
Supervisor, **Electrical and Electronics Eng. Dept., METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Department, METU _____

Assoc. Prof. Dr. Çağatay Candan
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Elif Vural
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Sevinç Figen Öktem
Electrical and Electronics Engineering Department, METU _____

Assoc. Prof. Dr. Ali Cafer Gürbüz
Electrical and Electronics Engineering Department,TOBB ETU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: AYLA UÇAR

Signature :

ABSTRACT

A COMPARISON OF SPARSE SIGNAL RECOVERY AND APPROXIMATE BAYESIAN INFERENCE METHODS FOR SPARSE CHANNEL ESTIMATION

Uçar, Ayla

M.S., Department of Electrical and Electronics Engineering

Supervisor : Assoc. Prof. Dr. Çağatay Candan

September 2015, 76 pages

The concept of sparse representation is one of the central methodologies of modern signal processing and it has had significant impact on numerous application fields such as communications and imaging. Sparsity expresses the idea that the information rate of a continuous time signal may be much smaller than suggested by its bandwidth, or that a discrete time signal depends on a number of degrees of freedom which is comparably much smaller than its (finite) length. With recent advances in sparse signal estimation, some new estimation techniques have emerged yielding more accurate sparse estimates than the traditional methods.

The main goal of this thesis is to analyse the performance of recently proposed sparse signal estimation methods on the problem of sparse channel estimation. In this thesis, a literature survey has been conducted to examine the approaches for estimating the sparse channels, then greedy pursuit algorithms, convex relaxation and an approximate Bayesian inference method, namely expectation propagation method, are comparatively studied.

Keywords: Sparsity, Sparse Channel Estimation, OMP, LASSO, Approximate Bayesian Inference, System Identification.

ÖZ

SEYREK (SPARSE) KANAL KESTİRİM ANALİZİ

Uçar, Ayla
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Çağatay Candan

Eylül 2015 , 76 sayfa

'Seyrek' (sparse) kanal kestirimi sinyal işleme alanının önde gelen araştırma konularından biridir ve haberleşme, biyomedikal görüntüleme gibi pek çok uygulama alanını ilgilendirmektedir. Zamanda sürekli sinyallerin (continuous time signal) taşıdığı bilgi oranının önerilen bant genişliğine göre düşük seviyelerde olması veya ayrık zamanlı sinyallerin (discrete time signal) sıfırdan farklı olan işaret sayısının sinyal uzunluğuna göre oldukça düşük olması durumu 'seyreklik' (sparsity) olarak tanımlanmaktadır. Seyrek kanal kestirimi alanında yapılan güncel çalışmalar sayesinde klasik kestirim yaklaşımlarına göre daha doğru sonuçlar veren bazı yeni kestirim teknikleri geliştirilmiştir.

Bu tezin amacı önerilen seyrek sinyal kestirim metodlarının seyrek kanal kestirim problemi üzerindeki performansını analiz etmektir. Bu tezde, seyrek kanal kestiriminde kullanılan yöntemlere ilişkin literatür taraması yapılmış, ardından greedy algoritması, konveks relaksiyon ve 'beklenti üretimi' (expectation propagation) olarak adlandırılan Bayes kestirim metodunun performansları karşılaştırmalı olarak çalışılmıştır.

Anahtar Kelimeler: Seyreklik, Seyrek Kanal Kestirimi, OMP, LASSO, Bayes Kestirimi, Sistem Tanımlama.

To my family

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Assoc. Prof. Çağatay Candan for his guidance, advice, encouragements and insight throughout my study.

I wish to express my gratitude to ASELSAN A.Ş for providing the opportunity to fulfil my study.

I would also like to thank my parents, Ayşe and Sezai and to my lovely brother, İbrahim for every success in my carrier.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

In conventional systems, the channel estimation is performed with the assistance of known transmitted training sequences. The receiver utilizes the known training bits and the corresponding received samples for the estimation of channel impulse response. The accuracy of the channel estimation is crucial for the receiver performance and it depends on the channel estimation algorithms and the environmental conditions, that is the statistics of the channel. There are several approaches for channel estimation such as the Maximum Likelihood Estimation, Least Squares or Minimum Mean Square Error Method etc.

Consider a communication system which is corrupted by noise as depicted in Figure 1.1. Source bits denoted by $\mathbf{u}$ are transmitted over the channel $\mathbf{x}$ and thermal noise, modelled by additive white Gaussian noise, is added. The received signal $\mathbf{y}$ can be expressed as follows:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}. \tag{1.1}$$

Here, the channel impulse response vector $\mathbf{x}$ is expressed as

$$\mathbf{x} = \begin{bmatrix} x_0 & x_1 & \dots & x_L \end{bmatrix}^T \tag{1.2}$$

and $\mathbf{w}$ denotes the noise samples. Within each transmission burst, the transmitter sends a unique training sequence of length P symbols:

$$\mathbf{u} = \begin{bmatrix} u_0 & u_1 & \dots & u_{P-1} \end{bmatrix}^T. \tag{1.3}$$

The Toeplitz training sequence matrix $\mathbf{A}$ in (1.1) can be explicitly written as

$$\mathbf{A} = \begin{bmatrix} u[0] & 0 & 0 & \dots & 0 \\ u[1] & u[0] & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ u[P-1] & u[P-2] & u[P-3] & \dots & u[P-N] \\ 0 & u[P-1] & u[P-2] & \dots & u[P-N+1] \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & u[P-1] & u[P-2] \\ 0 & 0 & \dots & 0 & u[P-1] \end{bmatrix}. \tag{1.4}$$

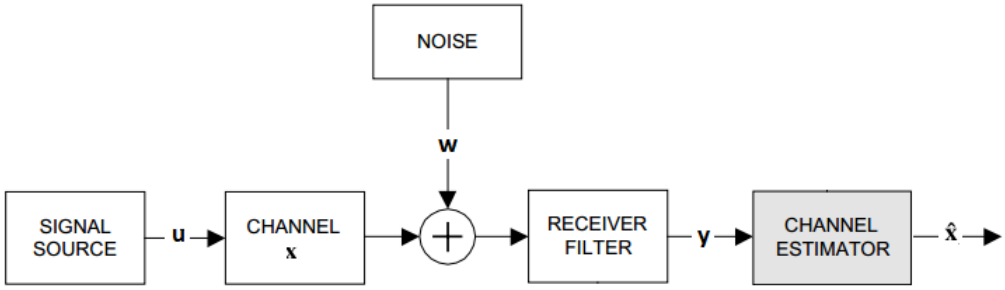

Figure 1.1: Block diagram of a noise-corrupted system

The main concern here is to estimate the channel taps from the received signal $\mathbf{y}$ and transmitted sequence $\mathbf{u}$. In this thesis, we concentrate on the estimation of channels with sparse impulse responses. The exploitation of sparse channel estimation is currently an active research field. Some channel measurements show that sparse or approximate sparse distribution assumption is valid [2], [3].

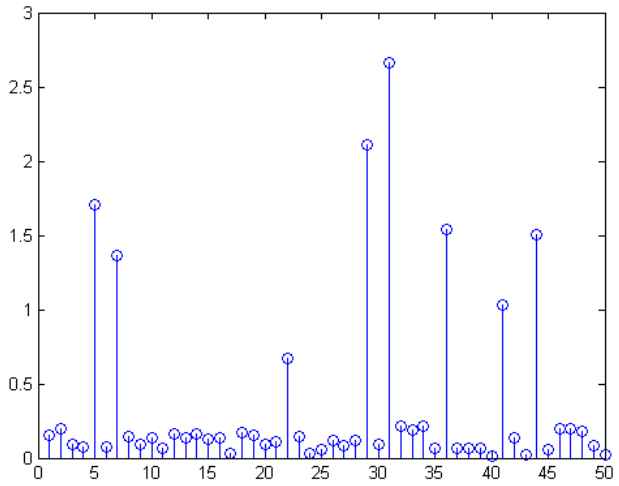

Figure 1.2: A typical example of sparse channel

Sparse channels are frequently encountered in the communication applications such

as Ultra Wide Band (UWB) channels [4], underwater acoustic communications [5] or mobile radio communications [6]. Due to the sparse structure of these channels, the sparse estimation techniques outperform the conventional least squares estimation algorithm which results in over parametrization and thus produces poor performance [7]. There are variety of approaches discussed in the literature for the sparse channel estimation with linear observation model. Among these, the most popular approaches are the greedy algorithms and convex optimizers.

Greedy algorithms iteratively add the most significant taps to the estimation and perform an update only on these selected taps as shown in Figure 1.3. The commonly used two greedy algorithms which differ in their update functions are Matching Pursuit (MP) [8] [9] and Orthogonal Matching Pursuit (OMP). An important aspect of the greedy algorithms is to determine when to stop the iteration. The number of iterations can be limited to some number, say k, or another approach such as limiting the energy of the residual can be considered. In the subsequent chapters, we observe that limiting the iteration has an important effect on the channel recovery performance.

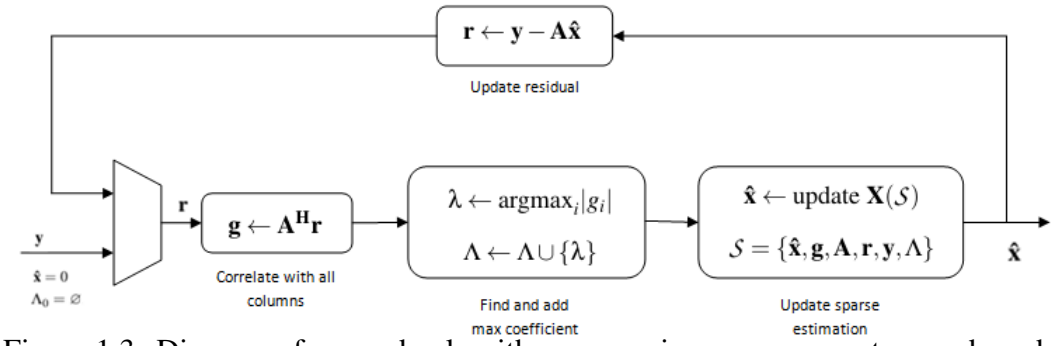

Figure 1.3: Diagram of a greedy algorithm processing measurements $\mathbf{y}$ and producing sparse channel estimates of $\mathbf{x}$ (definition of 'update $\mathbf{X}$' depends on the chosen algorithm)

Greedy algorithms are easy to implement, but have no convergence guarantee. The convex optimizer based on the linear programming resolves the convergence problem of greedy algorithms. The main advantage of the convex programming method is its guaranteed convergence and high estimation accuracy. However, this method is computationally complex and difficult to implement [10].

Bayesian inference is another approach for the solution of sparse channel estimation problem. It is a statistical inference method in which Bayes' rule is used to update the probability of an event. Exact Bayesian inference is not computationally feasible in many problems. For such problems, there are different types of approximate Bayesian inference methods such as Expectation Propagation, Importance Sampling, Iterative Quadrature, Laplace Approximation, Variational Bayes and Markov Chain Monte Carlo.

In this thesis, OMP and Least Absolute Shrinkage and Selection Operator (LASSO, in

some areas known as Basis Pursuit Denoising) methods are investigated. In addition, an efficient approximate Bayesian inference method called expectation propagation method is examined and the performances of the studied algorithms for the case of sparse channel recovery are compared.

The outline of the thesis is as follows: In Chapter 2, an overview of the traditional channel estimation techniques such as minimum variance unbiased estimator, maximum likelihood estimator, least squares and minimum mean square error estimator are provided. In Chapter 3, we present a background information on the basis concept, measurement matrices, atoms, uniqueness and restricted isometry property. Then, we describe types of sparse channel recovery methods and provide a brief summary of these methods. In Chapter 4, the simulation results on the OMP and LASSO methods are presented and the performance comparison of these methods are provided. In Chapter 5, an approximate Bayesian inference algorithm, namely expectation propagation, is investigated and the performance comparison of the approximate Bayesian inference algorithm to the previously studied methods are provided. Finally, in Chapter 6, we summarize the results obtained during our study.

CHAPTER 2

CHANNEL ESTIMATION

In this chapter we provide theoretical background on the channel estimation problem. Then, we introduce well known channel estimation techniques such as the minimum variance unbiased estimation, the maximum likelihood estimation, the least squares estimation and the minimum mean square error methods.

2.1 Overview of Channel Estimation

In telecommunications, the information-bearing transmitted signal and the disturbances introduced by the channel can be modelled statistically. This reflects the fact that receiver knows only some statistical properties of these signals, rather than the signals themselves. From these known statistical properties and the observations of the received signal, the receiver computes an estimate of the transmitted information [14].

The channel estimation is accomplished by transmitting a training sequence which is known by the transmitter-receiver pair. The channel estimation can be repeated in every transmitted burst. The channel estimator generates an estimate on the channel impulse response for each burst by exploiting transmitted bits and corresponding received bits.

Commonly used approaches to estimate parameters from a random sample are the minimum variance unbiased estimation, maximum likelihood estimation, the least squares estimation and the minimum mean square error method.

2.1.1 Minimum Variance Unbiased Estimation Method

The estimate $\hat{x}$ is said to be an unbiased estimate of x if

$$\mathrm{E}\{\hat{x}\} = x \quad \forall\, a < x < b \tag{2.1}$$

where (a, b) denotes the range of possible values of x.

The condition of unbiasedness does not necessarily mean that the estimator is "good". It only guarantees that average of the estimates yields the true value of the unknown parameter.

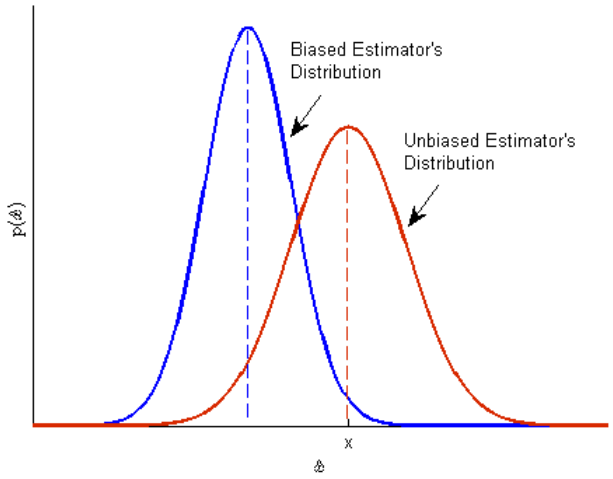

Figure 2.1: Biased and Unbiased Estimator

Even though the estimate is unbiased, sizeable errors are likely to occur. Minimizing the variance of the estimation error has the effect of concentrating the pdf of the estimation error $\epsilon = \hat{x} - x$ about zero. Therefore, we can say that the second measure of the quality of the estimate is to have a small error variance.

The Cramer-Rao Lower Bound (CRLB) allows us to determine that for any unbiased estimator the error variance is greater than or equal to the value indicated by the bound. The Minimum Variance Unbiased (MVU) estimation method relies on the concept of the sufficient statistics and closely related with the CRLB. The estimate $\hat{x}$ is said to be the MVU estimate of x if it has the smallest variance among all unbiased estimates of x. The variance of the MVU estimator does not always need to be equal to the CRLB, however if an estimator exists with variance equal to the CRLB, then it must be the MVU estimator.

For the linear observation model, the MVU estimator is efficient and achieves the CRLB. Consider the linear model given in (2.2)

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{2.2}$$

where $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{C_w})$.

Then, the MVU estimator is given by

$$\hat{\mathbf{x}}_{\mathbf{MVU}} = (\mathbf{A}^T {\mathbf{C_w}}^{-1} \mathbf{A})^{-1} \mathbf{A}^T {\mathbf{C_w}}^{-1} \mathbf{y} \tag{2.3}$$

and the covariance matrix of $\hat{\mathbf{x}}_{\mathbf{MVU}}$ is given by

$$\mathbf{C}_{\hat{\mathbf{x}}_{\mathbf{MVU}}} = (\mathbf{A}^T {\mathbf{C_w}}^{-1} \mathbf{A})^{-1}. \tag{2.4}$$

$\hat{\mathbf{x}}_{\mathbf{MVU}}$ is a linear transformation of a Gaussian vector $\mathbf{y}$, therefore statistical performance of $\hat{\mathbf{x}}_{\mathbf{MVU}}$ is completely specified, i.e.

$$\hat{\mathbf{x}}_{\mathbf{MVU}} \sim \mathcal{N}(\mathbf{x}, (\mathbf{A}^T {\mathbf{C}_\mathbf{w}}^{-1} \mathbf{A})^{-1}). \tag{2.5}$$

If $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$, then the MVU channel estimate in the presence of additive white Gaussian noise becomes

$$\hat{\mathbf{x}}_{\mathbf{MVU}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \tag{2.6}$$

and the covariance matrix of $\hat{\mathbf{x}}_{\mathbf{MVU}}$ becomes

$$\mathbf{C}_{\hat{\mathbf{x}}_{\mathbf{MVU}}} = \sigma_n^2 (\mathbf{A}^T \mathbf{A})^{-1}. \tag{2.7}$$

Note that, $\mathbf{A}^T \mathbf{A}$ is a symmetric Toeplitz autocorrelation matrix. In order to achieve the minimum possible variance of the MVU estimator, $\mathbf{A}$ should be chosen such that $\mathbf{A}^T \mathbf{A}$ becomes a diagonal matrix. Stated differently, under the total energy constraint of $\mathrm{tr}(\mathbf{A}^T \mathbf{A})$, the $\mathrm{tr}(\mathbf{A}^T \mathbf{A})^{-1}$ is minimized when $\mathbf{A}^T \mathbf{A}$ is proportional to the diagonal matrix , i.e., $\mathbf{A}^T \mathbf{A} \propto \mathbf{I}$ [15]. Hence, $\mathbf{A}^T \mathbf{A}$ should be ideally a scaled identity matrix. It can be noted that entries of $\mathbf{A}^T \mathbf{A}$ is related with the deterministic auto-correlation of the training sequence. Hence, ideally the training sequence should have impulsive autocorrelation. If this is the case, $\hat{\mathbf{x}}_{\mathbf{MVU}}$ becomes a Gaussian vector with diagonal covariance matrix.

2.1.2 Maximum Likelihood Estimation Method

There may be situations where MVU estimator does not exist, i.e. no unbiased estimates may exist or none of the unbiased estimates may have uniformly minimum variance, or MVU estimator cannot be found even if it exists. For these cases, we can find an approximately optimal estimator, termed the Maximum Likelihood (ML) estimator. The ML estimation is a well-defined method and optimal for large data records.

The ML function indicates how likely the observed sample $\mathbf{y}$ is as a function of possible parameter values $\mathbf{x}$. The ML estimate of $\mathbf{x}$ is mathematically defined as

$$\hat{\mathbf{x}}_{\mathbf{ML}} = \max_{\mathbf{x}} \ \ln \ \mathrm{p}(\mathbf{y}|\mathbf{x}). \tag{2.8}$$

Therefore, maximizing the log likelihood function determines the parameters that are most likely to produce the observed data.

Consider the linear model given in (2.9)

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{2.9}$$

where $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_\mathbf{w})$.

Under these conditions, $\mathrm{p}(\mathbf{y}|\mathbf{x})$ is given by

$$\mathrm{p}(\mathbf{y}|\mathbf{x}) = \frac{1}{((2\pi)^m|\mathbf{C_w}|)^{1/2}}\, \mathrm{e}^{-\frac{1}{2}(\mathbf{y}-\mathbf{A}\mathbf{x})^T\mathbf{C_w}^{-1}(\mathbf{y}-\mathbf{A}\mathbf{x})}. \tag{2.10}$$

Hence, for the above mentioned case it is required to minimize the cost function $\mathrm{J}(\mathbf{x})$ where

$$\mathrm{J}(\mathbf{x}) = (\mathbf{y}-\mathbf{A}\mathbf{x})^T\mathbf{C_w}^{-1}(\mathbf{y}-\mathbf{A}\mathbf{x}). \tag{2.11}$$

Solving for $\hat{\mathbf{x}}_{\mathbf{ML}}$ produces

$$\hat{\mathbf{x}}_{\mathbf{ML}} = (\mathbf{A}^T\mathbf{C_w}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{C_w}^{-1}\mathbf{y}. \tag{2.12}$$

The ML estimate is in general asymptotically unbiased and asymptotically achieves the CRLB, it is therefore asymptotically efficient. Besides this, for the linear observation model examined here, $\hat{\mathbf{x}}$ has a Gaussian pdf [16] and it is given by

$$\hat{\mathbf{x}}_{\mathbf{ML}} \sim \mathcal{N}(\mathbf{x}, (\mathbf{A}^T\mathbf{C_w}^{-1}\mathbf{A})^{-1}). \tag{2.13}$$

It can be noted that ML estimate coincides with MVU estimate. In general, when a closed form expression cannot be found for the ML estimate, it is possible to use numerical approaches such as brute-force search, Newton-Raphson or Expectation Maximization (EM) method by which an approximation to the ML estimate can be found.

2.1.3 Least Squares Estimation Method

The ML estimation uses the statistical knowledge on noise, that is its covariance $\mathbf{C_w}$ to find a "good" estimator that is asymptotically unbiased and efficient. However, in the Least Squares (LS) estimation no probabilistic assumptions are made. The only criterion is to minimize the l_2 norm of error. The LS estimation is widely used in practice, however no claims about optimality can be made in general [16]. The most important aspect of LS is the data fitting. The best fit in the LS sense minimizes the l_2 norm of error.

Consider the linear model given in (2.14)

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{2.14}$$

where $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{C_w})$.

Then, the LS estimate of $\mathbf{x}$ is given by the minimizer of $\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2$ with respect to $\mathbf{x}$. Hence, we need to minimize

$$\mathrm{J}(\mathbf{x}) = (\mathbf{y}-\mathbf{A}\mathbf{x})^T(\mathbf{y}-\mathbf{A}\mathbf{x}). \tag{2.15}$$

Taking the first order partial derivative of $\mathrm{J}(\mathbf{x})$ with respect to $\mathbf{x}$ and setting it equal to zero yield the LS estimate of $\mathbf{x}$ which is

$$\hat{\mathbf{x}}_{\mathbf{LS}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}. \tag{2.16}$$

Here, $\mathbf{A}$ is assumed to be full rank to guarantee the inversion of $(\mathbf{A}^T\mathbf{A})^{-1}$. Thus, the estimate is unique and minimizes $\mathrm{J}(\mathbf{x})$.

From a geometrical perspective, the linear LS estimation takes the orthogonal projection of $\mathbf{y}$ onto the subspace spanned by the columns of $\mathbf{A}$. The aim is to make the error vector $\epsilon = \mathbf{y} - \mathbf{A}\mathbf{x}$ orthogonal to the columns of $\mathbf{A}$. This is the well-known orthogonality principle.

Then, minimum cost function $\mathrm{J}_{\min}$ can be calculated as

$$\begin{aligned} \mathrm{J}_{\min} &= (\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{\mathbf{LS}})^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{\mathbf{LS}}) \\ &= \mathbf{y}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_{\mathbf{LS}}) - \hat{\mathbf{x}}_{\mathbf{LS}}^T\mathbf{A}^T\epsilon \\ &= \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{A}\hat{\mathbf{x}}_{\mathbf{LS}} \\ &= \mathbf{y}^T(\mathbf{I} - \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T)\mathbf{y}. \end{aligned} \tag{2.17}$$

For the examined problem, the linear LS estimator is a special case of the ML method. We see that the ML estimate is equivalent to the linear LS estimate for the linear models with white Gaussian noise.

Some other adaptation rules like Recursive Least Squares (RLS) [17] could be considered in order to track time-varying signals.

2.1.4 Minimum Mean Square Error Method

The LS estimate is commonly used because of its simplicity. However; if the channel statistics of $\mathbf{x}$ is known a-priori, then we can do better than the LS estimate. The a-priori information can be exploited to decrease the estimation error. For instance, if it is known that $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \mathbf{C}_{\mathbf{x}})$ a-priori, then the estimate of $\mathbf{x}$ which minimizes the mean square error (MSE, $\mathrm{E}\{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2\}$) and its error covariance matrix is given by

$$\hat{\mathbf{x}}_{\mathbf{MMSE}} = \mathrm{E}\{\mathbf{x}|\mathbf{y}\} = \boldsymbol{\mu}_{\mathbf{x}} + \mathbf{C}_{\mathbf{x}}\mathbf{A}^T(\mathbf{A}\mathbf{C}_{\mathbf{x}}\mathbf{A}^T + \mathbf{C}_{\mathbf{w}})^{-1}(\mathbf{y} - \mathbf{A}\mu_{\mathbf{x}}) \tag{2.18}$$

and

$$\mathbf{C}_{\mathbf{e}} = \mathbf{C}_{\hat{\mathbf{x}}_{\mathbf{MMSE}}} = \mathbf{C}\{\mathbf{x}|\mathbf{y}\} = \mathbf{C}_{\mathbf{x}} - \mathbf{C}_{\mathbf{x}}\mathbf{A}^T(\mathbf{A}\mathbf{C}_{\mathbf{x}}\mathbf{A}^T + \mathbf{C}_{\mathbf{w}})^{-1}\mathbf{A}\mathbf{C}_{\mathbf{x}}. \tag{2.19}$$

The alternative form of MMSE estimate and its error covariance matrix is provided below.

$$\hat{\mathbf{x}}_{\mathbf{MMSE}} = \mathrm{E}\{\mathbf{x}|\mathbf{y}\} = \boldsymbol{\mu}_{\mathbf{x}} + (\mathbf{A}^T{\mathbf{C}_{\mathbf{w}}}^{-1}\mathbf{A} + {\mathbf{C}_{\mathbf{x}}}^{-1})^{-1}\mathbf{A}^T{\mathbf{C}_{\mathbf{w}}}^{-1}(\mathbf{y} - \mathbf{A}\mu_{\mathbf{x}}) \tag{2.20}$$

and

$$\mathbf{C_e} = \mathbf{C}_{\hat{\mathbf{x}}_{\mathbf{MMSE}}} = \mathbf{C}\{\mathbf{x}|\mathbf{y}\} = (\mathbf{A}^T {\mathbf{C_w}}^{-1}\mathbf{A} + {\mathbf{C_x}}^{-1})^{-1}. \tag{2.21}$$

The equations (2.18) and (2.19) require $m \times m$ matrix inversion whereas (2.20) and (2.21) requires $n \times n$ matrix inversion. So that, it is recommended to use the second form for the applications where the number of observations is significantly larger than the number of unknowns, that is $m >> n$.

When $\mathbf{C_x} \to \infty$ (meaning not having a-priori knowledge of $\mathbf{x}$), then MMSE and LS estimates coincide. Generally MMSE estimate is better (in the MSE sense) than the LS estimate, owing to the a-priori knowledge of $\mathbf{x}$.

CHAPTER 3

AN OVERVIEW OF SPARSE CHANNEL RECOVERY METHODS AND PROBLEM STATEMENT

Wireless channels in particular propagation environments are characterized as sparse or sparse clustered. The sparsity of the channel impulse response is illustrated in Figure 3.1, showing only a few peaks and many zeros in between.

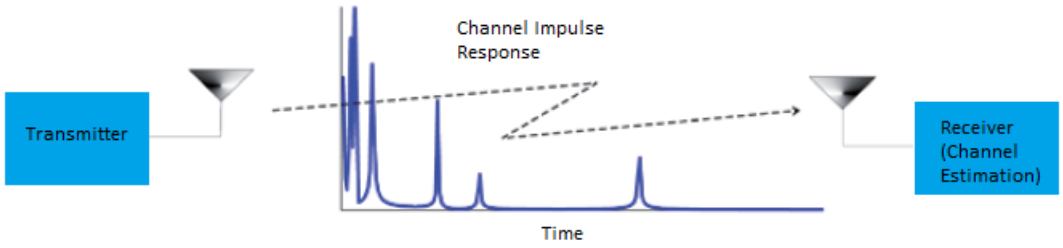

Figure 3.1: Channel with Sparse Impulse Response

There are many advantages of working with the sparse vectors. For instance, calculations involving multiplying a vector by a matrix take less time if the vector is sparse. In addition, the vectors are often used to represent large amount of data which can be difficult to store or transmit and by using a sparse approximation, the amount of space needed to store the vector would be reduced to a fraction of what was originally needed when they are stored only as the position and the value of the non-zero entries.

In this chapter we are mainly interested in sparse channel estimation. MP and OMP are efficient greedy algorithms used for estimating sparse channels. Both of them iteratively build up the sparse signal by selecting the vector that improves the representation at each iteration. In the MP algorithm, iteration optimization is performed over all vectors in the dictionary so that it is possible to re-select a previously selected vector. This process slows down the convergence speed to a sparse solution [7]. In the OMP algorithm the re-selection problem is eliminated by using the stored dictionary at each iteration and more accurate channel estimates can be acquired [18]. On the other hand, the convex optimizer Basis Pursuit (BP) [19] searches for the vector

which minimizes the l_1 norm of the solution. It substitutes l_0 norm by the closest convex norm, which is the l_1 norm and turns the problem into an l_1 norm minimization problem. Since the mentioned algorithms are better suited for the channels we consider, they give more accurate estimates than the traditional methods such as LS [20] and MMSE.

In this chapter, an overview of the sparse signal recovery is presented. In Section 3.1 basis concept is examined. In Section 3.2 and 3.3, we provide definitions for "dictionary", "atom" and "measurement matrix", which are frequently used in the incoming chapters. In Section 3.4, 3.5 and 3.6, we deal with the sparsity concept and in Section 3.7 commonly used sparse signal estimation techniques are briefly summarized.

3.1 Review of Basis

A set $\{\boldsymbol{\phi}_i\}_{i=1}^n$ is called a basis for $\mathbb{R}^n$ if the vectors in the set span $\mathbb{R}^n$ and are linearly independent. This implies that each vector in the space has a unique representation as a linear combination of these basis vectors. Specifically, for any $\mathbf{x} \in \mathbb{R}^n$, there exist unique coefficients $\{c_i\}_{i=1}^n$ such that

$$\mathbf{x} = \sum_{i=1}^{n} c_i \boldsymbol{\phi}_i. \tag{3.1}$$

Let $\boldsymbol{\phi}$ denote the $n \times n$ matrix with columns given by $\boldsymbol{\phi}_i$, and let $\mathbf{c}$ denote the length-n vector with entries c_i, then this relation can be expressed as

$$\mathbf{x} = \boldsymbol{\phi}\mathbf{c}. \tag{3.2}$$

An important special case of a basis is an orthonormal basis, defined as a set of vectors $\{\boldsymbol{\phi}_i\}_{i=1}^n$ satisfying the inner product relation

$$\langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}. \tag{3.3}$$

An orthonormal basis has the advantage that the coefficients of $\mathbf{c}$ can be easily calculated as

$$c_i = \langle \mathbf{x}, \boldsymbol{\phi}_i \rangle \tag{3.4}$$

or

$$\mathbf{c} = \boldsymbol{\phi}^T \mathbf{x}. \tag{3.5}$$

3.2 Dictionaries and Atoms

Mallat [21] introduced the terminology of dictionary. A dictionary $\mathbf{D}$ is a collection of parametrized waveforms such that $\mathbf{D} = \{\boldsymbol{\phi}_\gamma : \gamma \in \Gamma\}$. The waveforms $\boldsymbol{\phi}_\gamma$ are

discrete signals of length m, called atoms. Depending on the dictionary, the parameter γ can have the interpretation of indexing frequency in which case the dictionary is a frequency or Fourier dictionary, of indexing time-scale in which case the dictionary is a time-scale dictionary or of indexing time-frequency jointly in which case the dictionary is a time-frequency dictionary [19].

3.3 Measurement Matrices

Consider a model such that

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{3.6}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{y} \in \mathbb{R}^m$. The matrix $\mathbf{A}$ is called sensing matrix or measurement matrix.

If $m = n$ and the dictionary furnishes a basis, then $\mathbf{A}$ becomes an $n \times n$ non-singular matrix. In this case, we can say that there exists a unique representation of $\mathbf{x}$ such that $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$. When the atoms are, in addition, mutually orthonormal then $\mathbf{A}^{-1} = \mathbf{A}^T$ and the reconstruction formula becomes simpler.

If m is smaller than n, then the matrix $\mathbf{A}$ represents a dimensionality reduction, i.e., it maps $\mathbb{R}^n$ into $\mathbb{R}^m$. In the over complete case ($m \ll n$), $\mathbf{A}$ is not invertible. There are then many representations of $\mathbf{x}$, i.e., there is no unique $\mathbf{x}$.

We are motivated by the aim of achieving the sparsest possible representation and non-uniqueness gives us the possibility of adaptation. It allows us to choose from among many representations the one that is most suited to our purpose.

3.4 Sparse Models

Signals can often be well-approximated as a linear combination of just a few elements from a known basis or dictionary. When this representation is exact it is said that the signal is sparse. Sparse signal models provide a mathematical framework for capturing the fact that in many cases high-dimensional signals contain relatively little information compared to their ambient dimension [22].

Sparsity is measured by the l_0 norm. Mathematically, a signal $\mathbf{x} \in \mathbb{R}^n$ is called k-sparse, if it has at most k non-zeros, i.e., $\|\mathbf{x}\|_0 \leq k$. The set of all k-sparse signals can be denoted by $\sum_k$ such that

$$\sum_k = \{\mathbf{x} : \|\mathbf{x}\|_0 \leq k\}. \tag{3.7}$$

In the following subsections, when $\mathbf{x}$ is referred to be as k-sparse, it should be understood that either $\mathbf{x}$ itself is sparse or $\mathbf{x}$ can be expressed as $\mathbf{x} = \boldsymbol{\phi}\mathbf{c}$ where $\|\mathbf{c}\|_0 \leq k$.

3.5 Sparse Signal Recovery

To state the problem mathematically precisely, let $\mathbf{x} \in \mathbb{R}^n$ be the signal of interest. As a prior information, it is assumed that $\mathbf{x}$ is sparse, i.e., it has very few non-zero coefficients or there exists a frame $\boldsymbol{\phi}$ such that $\mathbf{x} = \boldsymbol{\phi}\mathbf{c}$ with $\mathbf{c}$ being sparse. Further, consider $\mathbf{A}$ to be a full rank $m \times n$ matrix with $m < n$ and define the under-determined linear system of equation $\mathbf{y} = \mathbf{A}\mathbf{x}$.

In that case, it is clear that

- there are more variables than equations,
- $\mathbf{x}$ is underspecified, i.e., many choices of $\mathbf{x}$ lead to the same $\mathbf{y}$.

In order to narrow down the choice to one particular solution, there is a need for an additional criteria such as introducing a cost function $\mathrm{J}(\mathbf{x})$. Given measurements $\mathbf{y}$ and the knowledge that original signal $\mathbf{x}$ is sparse, the recovery of $\mathbf{x}$ can be achieved by solving

$$\hat{\mathbf{x}} = \min_{\mathbf{x}} \mathrm{J}(\mathbf{x}) \tag{3.8}$$

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

Selecting a convex function $\mathrm{J}(.)$ guarantees a unique solution. If we choose $\mathrm{J}(\mathbf{x})$ to be the squared Euclidean norm $\|\mathbf{x}\|_2^2$ then the unique solution $\hat{\mathbf{x}}$, so-called minimum norm solution, is given by

$$\hat{\mathbf{x}} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{y}. \tag{3.9}$$

$\mathbf{A}\mathbf{A}^T$ is invertible since $\mathbf{A}$ is full rank and at this point it is worth saying that $\hat{\mathbf{x}} \perp \mathcal{N}(\mathbf{A})$ where $\mathcal{N}(\mathbf{A}) = \{\mathbf{x} | \mathbf{A}\mathbf{x} = \mathbf{0}\}$.

When we define $\mathrm{J}(.)$ to be the l_0 norm of $\mathbf{x}$ such that $\mathrm{J}(\mathbf{x}) = \mathrm{J}_0(\mathbf{x}) = \|\mathbf{x}\|_0$, the recovery of $\mathbf{x}$ can be achieved by solving

$$(\mathrm{P}_0) : \hat{\mathbf{x}} = \mathrm{argmin}_{\mathbf{z}} \|\mathbf{z}\|_0 \quad \text{subject to } \mathbf{z} \in \mathrm{B}(\mathbf{y}) \tag{3.10}$$

where $\mathrm{B}(\mathbf{y}) = \{\mathbf{z} : \mathbf{A}\mathbf{z} = \mathbf{y}\}$ for the case of noise-free measurements. When the measurements have been corrupted with a small amount of bounded noise, one could consider $\mathrm{B}(\mathbf{y}) = \{\mathbf{z} : \|\mathbf{A}\mathbf{z} - \mathbf{y}\|_2 \leq \epsilon\}$. In both cases, (P_0) finds the sparsest $\mathbf{x}$ that is consistent with the measurement $\mathbf{y}$. It should be noted that l_0 norm, which is the number of non-zero components of a vector $\mathbf{x}$, is a quasi norm, since it does not satisfy homogeneity property, that is $\|\alpha\mathbf{x}\|_0 \neq |\alpha|\|\mathbf{x}\|_0$.

Unfortunately, this problem requires an exhaustive search and, in general, it is not a feasible problem. One way to convert this problem to a more tractable one is to replace l_0 norm with a convex approximation, i.e., l_1 norm. Chen, Donoho and Saunders have shown that l_1 minimization promotes sparsity, [22]. This leads us to the

following minimization problem, which is coined as Basis Pursuit:

$$(\mathrm{P}_1) : \hat{\mathbf{x}} = \text{argmin}_{\mathbf{z}} \|\mathbf{z}\|_1 \quad \text{subject to } \mathbf{z} \in \mathrm{B}(\mathbf{y}) \tag{3.11}$$

where $\mathrm{B}(\mathbf{y}) = \{\mathbf{z} : \mathbf{A}\mathbf{z} = \mathbf{y}\}$. Again, when the measurements have been corrupted by noise, one could consider $\mathrm{B}(\mathbf{y}) = \{\mathbf{z} : \|\mathbf{A}\mathbf{z} - \mathbf{y}\|_2 \leq \epsilon\}$.

The l_0 minimization algorithms directly attempt to solve (P_0). Examples of l_0 minimization algorithms include OMP, Stagewise OMP (StOMP), Regularized OMP (ROMP), Compressive Sampling MP (CoSaMP), Iterative Hard Thresholding (IHT) etc.

The l_1 minimization algorithms find sparse solutions by solving (P_1). l_1 minimization algorithms include Basis Pursuit, LASSO, Weighted Least Squares, iterative algorithms based on gradient thresholding etc.

In order to ensure the recovery of the sparse signal $\mathbf{x}$ from the measurements of $\mathbf{y}$, the measurement matrix $\mathbf{A}$ with $m \ll n$ should satisfy a number of desirable properties. In the following subsections, the uniqueness conditions for the solutions to problems (P_0) and (P_1) are given, then the conditions under which (P_0) has the same solution as (P_1) are provided.

3.6 The Sparsest Solution of $\mathbf{y} = \mathbf{A}\mathbf{x}$

Sufficient conditions to coincide the solutions of (P_0) and (P_1) do not only depend on the sparsity of the original signal $\mathbf{x}$, but also on the coherence of the measurement matrix $\mathbf{A}$. The equivalence of l_0 and l_1 problems in this case can also be phrased in terms of restricted isometry property.

3.6.1 Uniqueness

3.6.1.1 Uniqueness via Spark

The null space of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is denoted as

$$\mathcal{N}(\mathbf{A}) = \{\mathbf{z} : \mathbf{A}\mathbf{z} = \mathbf{0}\}. \tag{3.12}$$

In order to recover all sparse signals $\mathbf{x}$ from the measurements $\mathbf{y}$, one must have $\mathbf{A}\mathbf{x} \neq \mathbf{A}\mathbf{x}'$ for any pair of distinct vectors $\mathbf{x}$, $\mathbf{x}' \in \sum_k$, since otherwise it would be impossible to distinguish $\mathbf{x}$ from $\mathbf{x}'$ based solely on the measurements $\mathbf{y}$. Formally, if $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}'$ then $\mathbf{A}(\mathbf{x} - \mathbf{x}') = \mathbf{0}$ with $\mathbf{x} - \mathbf{x}' \in \sum_{2k}$. So, it is clear that $\mathbf{A}$ uniquely represents all $\mathbf{x} \in \sum_k$ if and only if $\mathcal{N}(\mathbf{A})$ contains no vectors in $\sum_{2k}$. This property can be characterized by using spark definition.

The word $spark$ comes from a verbal fusion of "sparse" and "rank" and spark of $\mathbf{A}$ denoted by Spark($\mathbf{A}$) is the smallest number of linearly dependent columns of $\mathbf{A}$. By definition, Spark($\mathbf{A}$) $\in [2, m+1]$. From this property we can say that Theorem 3.1, given below yields the requirement of $m \geq 2k$ to ensure uniqueness.

Theorem 3.1 *[23] For any vector* $\mathbf{y} \in \mathrm{R}^m$*, there exists at most one solution* $\mathbf{x} \in \sum_k$ *such that* $\mathbf{y} = \mathbf{A}\mathbf{x}$ *if and only if* $k < Spark(\mathbf{A})/2$.

It is clear from Theorem 3.1 that if there exists a solution $\mathbf{x}$ satisfying the condition $\|\mathbf{x}\|_0 < \text{Spark}(\mathbf{A})/2$, then this solution is the sparsest possible.

For instance, if random matrix $\mathbf{A}$ comprises independent and identically distributed (i.i.d.) entries with continuous distributions, then we can say that Spark($\mathbf{A}$) $= m+1$ with high probability. This implies that no m columns are linearly dependent. In this case, uniqueness is ensured for every solution with $m/2$ or fewer non-zero entries.

3.6.1.2 Uniqueness via Mutual Coherence

Calculating the spark involves checking the dependence of combinations of columns of the matrix $\mathbf{A}$. This can be expensive and difficult to evaluate. A simpler way that guarantees uniqueness is the use of mutual coherence of the matrix $\mathbf{A}$. The mutual coherence of a given matrix $\mathbf{A}$ is the largest absolute normalized inner product between different columns from $\mathbf{A}$.

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$. Denoting the i^{th} column in $\mathbf{A}$ by $\mathbf{a_i}$, the mutual coherence $\mu(\mathbf{A})$ is given by

$$\mu(\mathbf{A}) = \max_{1 \leq i,j \leq n, i \neq j} \frac{|\langle \mathbf{a_i}, \mathbf{a_j} \rangle|}{\|\mathbf{a_i}\|_2 \|\mathbf{a_j}\|_2}. \tag{3.13}$$

The mutual coherence is a way to characterize the dependence between columns of the matrix $\mathbf{A}$. The measurement matrices are required to have a small coherence. The coherence of a matrix is always in the range $\mu(\mathbf{A}) \in \left[\sqrt{\frac{n-m}{m(n-1)}}, 1\right]$. The maximal coherence of a matrix is 1 in the case that two columns coincide. The lower bound is known as the Welch bound. Note that when $m \ll n$, the lower bound is approximately equals to $1/\sqrt{m}$.

For any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, the following relationship holds [24]:

$$\text{Spark}(\mathbf{A}) \geq 1 + \frac{1}{\mu(\mathbf{A})}. \tag{3.14}$$

By merging Theorem 3.1 and inequality (3.14), one can pose the following condition on $\mathbf{A}$ that guarantees the uniqueness.

Theorem 3.2 *[25] If*

$$k < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{A})}\right) \tag{3.15}$$

then for each measurement vector $\mathbf{y} \in \mathbb{R}^m$ *there exists at most one signal* $\mathbf{x} \in \sum_k$ *such that* $\mathbf{y} = \mathbf{A}\mathbf{x}$.

To conclude, if $\mathbf{y} = \mathbf{A}\mathbf{x}$ where $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m < n$ has a solution $\mathbf{x} \in \mathbb{R}^n \setminus \{0\}$ obeying $\|\mathbf{x}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{A})}\right)$, we can say that this solution is the sparsest and unique solution of both l_0 and l_1 minimization.

Theorem 3.2, together with the Welch bound, provides an upper bound on the sparsity level k that guarantees uniqueness using coherence: $k = \mathcal{O}(\sqrt{m})$.

Comparison of Theorem 3.1 with Theorem 3.2:

Theorem 3.1 and Theorem 3.2 are similar in form, but have different assumptions. Theorem 3.1 is more powerful than Theorem 3.2 which uses the coherence and so only a lower bound on spark. The coherence cannot be smaller than $1/\sqrt{m}$, therefore the cardinally bound of Theorem 3.2 is never larger than $\sqrt{m}/2$. However, the spark can easily be as large as m and Theorem 3.1 gives a bound as large as $m/2$.

3.6.2 The Restricted Isometry Property

When measurements are corrupted by additive noise, Restricted Isometry Property (RIP) should be taken into consideration. If a matrix $\mathbf{A}$ satisfies the RIP, then this is sufficient for a variety of algorithms to have a stable recovery in the presence of additive noise [26].

Let $\mathbf{A}$ be an $m \times n$ matrix. Then, matrix $\mathbf{A}$ satisfies the RIP of order k if, for all $\mathbf{x} \in \sum_k$, there exists a $\delta_k \in (0, 1)$ such that

$$(1 - \delta_k)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_k)\|\mathbf{x}\|_2^2. \tag{3.16}$$

If the RIP holds, then l_1 minimization gives an accurate reconstruction, i.e., it is possible to recover $\mathbf{x} \in \sum_k$. RIP can be achieved with high probability by selecting $\mathbf{A}$ as a random matrix. Random matrices from Gaussian distribution satisfies RIP of order k with high probability if [27]

$$m = \mathcal{O}(k \ln(n/k)/\delta_k^2). \tag{3.17}$$

Moreover, if $\mathbf{A}$ has unit-norm columns and coherence $\mu = \mu(\mathbf{A})$, then $\mathbf{A}$ satisfies the RIP of order k with $\delta_k \leq (k - 1)\mu$.

RIP enables recovery guarantee that is much stronger than those based on spark and coherence. However, checking whether a matrix $\mathbf{A}$ satisfies the RIP property has a combinatorial computational complexity.

Recently, researchers have observed that sparse matrices may satisfy a related property, called RIP-1, even when they do not satisfy inequality (3.16). RIP-1 can also be used to analyse sparse approximation algorithms [28].

3.7 Sparse Signal Recovery Methods

In the literature, there are several methods discussing the sparse signal recovery. These methods can roughly be divided into 3 main categories:

1. greedy pursuit algorithms include OMP, StOMP, ROMP, CoSaMP or IHT
2. convex relaxation methods include BP, LASSO, interior-point methods, projected gradient methods or iterative thresholding
3. Bayesian framework

Pursuit algorithms iteratively refine a sparse solution by successively identifying the components that yield the greatest improvement in quality. They build up the sparse signal by greedy decisions which iterate between 2 main steps:

- Support Update: The support of a signal is the locations of the non-zero entries and is sometimes called its "sparsity pattern". The algorithm makes a guess about the columns (or atoms) of the dictionary which have been used to generate observations.
- Coefficient Update: The estimate of the signal is updated by taking into account the latest decision about the support.

Convex relaxation algorithms replace the combinatorial problem with a convex optimization problem. These techniques solve a convex program whose minimizer is known to approximate the target signal. One of the commonly used convex relaxation algorithm is the BP. It finds the solution of non-quadratic optimization problems by l_1 minimization.

Bayesian framework makes use of a prior distribution for the unknown coefficients of the sparse signal that favours the sparsity and develops a maximum a posteriori estimator that incorporates the observation.

There are also some combinatorial approaches to the sparse recovery problem. For instance, in [29], the authors propose to combine Bayesian approach with the pursuit algorithms to produce Bayesian OMP (BOMP), Bayesian StOMP (BStOMP) and Bayesian CoSaMP (BCoSaMP) which result in a more efficient signal recovery with respect to the non-Bayesian pursuit algorithms.

3.7.1 Pursuit Algorithms

Pursuit methods build up a sparse approximation by making locally optimal choices at each iteration. They basically differ in the way they implement support update (identification step) and coefficient update (estimation step) of the signal.

One of the most used and simplest greedy approach is the Orthogonal Matching Pursuit. OMP considers finding the column of the given matrix $\mathbf{A}_{m\times n}$ which promotes the maximum correlation with the measurement. The identification step is the most expensive part of the computation which costs $\mathcal{O}(mn)$ for an unstructured dense matrix. The estimation step requires the solution of a least squares problem. The algorithm then repeats these two steps by correlating the columns of $\mathbf{A}$ with the residual, which is obtained by subtracting the contribution of the partial estimate from the original measurement vector.

It is clear that there is a need for a stopping rule to halt the iterations. The natural stopping criteria are given below:

- Cease after a fixed number of iterations: k
- Cease when residual has small magnitude: $||\mathbf{r_t}||_2 \leq \epsilon$
- Cease when no column explains a significant amount of energy in the residual: $||\mathbf{A}^T\mathbf{r_{t-1}}||_\infty \leq \epsilon$

These criteria can all be implemented at minimal cost.

Tropp and Gilbert [30] investigated the performance of OMP algorithm by measurement. It is reported in [30] that if the measurement matrices satisfy some properties, then OMP algorithm can recover the sparse signals with high probability. It has been shown that by considering random matrices for $\mathbf{A}$, OMP can recover k-sparse signals with high probability using $m \approx \mathcal{O}(k\ \ln(n))$ measurements.

The overall OMP algorithm is presented below:

OMP Algorithm

Task: Approximate the solution $(\mathrm{P}_0) : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad$ subject to $\mathbf{y} = \mathbf{A}\mathbf{x}$

GET: $\mathbf{A} \in \mathbb{R}^{m\times n}$ where columns of $\mathbf{A}$ are denoted as $(\mathbf{a_i})_{i=1}^n$, measurement $\mathbf{y}$ and the "stopping criterion"

SET: the residual $\mathbf{r_0} = \mathbf{y}$, index set $\Lambda_0 = \varnothing$, $S_0 = \varnothing$ and the counter $t = 1$

1. Find the index λ_t that solves the optimization problem:

$$\lambda_t = \mathrm{argmax}_{i=1,\ldots,n} |\langle \mathbf{r_{t-1}}, \mathbf{a_i} \rangle|$$

2. Augment the index set and the matrix of chosen columns:

$$\Lambda_t = \Lambda_{t-1} \bigcup \{\lambda_t\}$$

and

$$\mathbf{S_t} = \begin{bmatrix} \mathbf{S_{t-1}} & \mathbf{a}_{\lambda_t} \end{bmatrix}$$

3. Solve the least squares problem to obtain signal estimate:

$$\hat{\mathbf{x}}_\mathbf{t} = \text{argmin}_\mathbf{x} \|\mathbf{y} - \mathbf{S_t}\mathbf{x}\|_2$$

4. Calculate the approximation of the measurement and update the residual:

$$\mathbf{z_t} = \mathbf{S_t}\hat{\mathbf{x}}_\mathbf{t}$$

$$\mathbf{r_t} = \mathbf{y} - \mathbf{z_t}$$

Increment t. Repeat steps (1)-(4) until stopping criterion holds.

OUTPUT: $\hat{\mathbf{x}}$ is obtained after t iterations depending on the stopping criterion. The estimate $\hat{\mathbf{x}}$ has non-zero indices at the components listed in Λ such that $\hat{x}(\lambda) = \hat{x}_t(\lambda)$ for $\lambda \in \Lambda_t$ and $\hat{x}(\lambda) = 0$ otherwise.

For many applications, OMP does not offer adequate performance, so researchers have developed more sophisticated pursuit methods that perform better in practice. These techniques depend on several enhancements to the basic greedy framework:

1. selecting multiple columns per iteration
2. pruning the set of active columns at each step
3. solving the least squares problems iteratively
4. theoretical analysis using the RIP bound

StOMP [31] selects multiple columns at each step. ROMP [32] was the first greedy technique whose analysis was supported by a RIP bound. CoSaMP [33] was the first algorithm to assemble these ideas to obtain essentially optimal performance guarantees.

3.7.2 Convex Relaxation Methods

Another fundamental approach to sparse approximation replaces the combinatorial l_0 function with the l_1 norm, yielding convex optimization problems. Indeed, l_1 norm is the closest convex function to the l_0 function, so this relaxation is quite natural [34].

In Section 3.5 we have denoted the convex form of the sparse approximation problem by (P_1) and in (P_1), if we take $\mathrm{B}(\mathbf{y}) = \{\mathbf{z} : \mathbf{Az} = \mathbf{y}\}$, the minimization problem can be posed as a linear program. In the case of $\mathrm{B}(\mathbf{y}) = \{\mathbf{z} : \|\mathbf{Az} - \mathbf{y}\|_2 \leq \epsilon\}$, (P_1) becomes a convex problem and can be solved by a convex optimization approach.

The dual form of the convex problem can be formulated by

$$\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \tau\|\mathbf{x}\|_1. \tag{3.18}$$

Here, $\tau \geq 0$ is a regularization parameter whose value governs the sparsity level of the estimate. Large values of τ typically produce sparser estimates. So, there is a need to solve (3.18) repeatedly for different choices of this parameter or to trace systematically the path of solutions as τ decreases toward zero.

Another variant is the LASSO formulation such that

$$\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \quad \text{subject to } \|\mathbf{x}\|_1 \leq \lambda. \tag{3.19}$$

The LASSO which is in some areas known as Basis Pursuit Denoising is equivalent to (3.18) in the sense that the path of solutions to (3.19) parametrized by positive λ matches the solution path for (3.18) as τ varies [34].

Theorem 3.3 says that l_1 minimization recovers the signal with a high probability when a certain condition is met.

Theorem 3.3 *[22] Let* $\mathbf{A}$ *be an* $m \times n$ *matrix that satisfies RIP. Suppose that measurements obey,*

$$m \geq C\, k \ln\frac{n}{k} \tag{3.20}$$

then minimizing l_1 *reconstructs* $\mathbf{x} \in \sum_k$ *with a high probability.*

If the constant $C = 22(\delta + 1)$, then the probability of success exceeds $1 - \mathcal{O}(n^{-\delta})$.

3.7.3 Bayesian Framework

There are variety of approximate Bayesian inference methods such as Variational Bayes, Expectation Propagation, Laplace Approximation and Markov Chain Monte Carlo.

In this thesis, we study the approximate Bayesian inference method called expectation propagation method [11], [12], [13]. For details, please refer to Chapter 5.

3.8 Multiple Measurement Vectors

For the multiple measurement vector (MMV) case, there exist l signals each of which is sparse with the same indices for their non-zero coefficients. In this setting, one attempts to simultaneously recover the set of jointly sparse signals $\{\mathbf{x_i}\}_{i=1}^{l}$ from incomplete measurements rather than recovering the l signals separately. This problem

is an extension of single sparse signal recovery. By putting $\mathbf{x_i}$'s into the columns of a matrix $\mathbf{X}$, there will be at most k non-zero rows in $\mathbf{X}$. That is, not only each signal is k-sparse, but also the non-zero values occur on a common location set. Therefore, $\mathbf{X}$ is row-sparse and the notation $\Lambda = \text{supp}(\mathbf{X})$ is used to denote the index set corresponding to the non-zero rows of $\mathbf{X}$.

Assume that measurements are given to be $\{\mathbf{y_i}\}_{i=1}^{l}$ where each vector is of length $m < n$, and $\mathbf{Y}$ is an $m \times l$ matrix with columns $\mathbf{y_i}$. MMV problem tries to recover $\mathbf{X}$ assuming a known matrix $\mathbf{A}$ so that $\mathbf{Y} = \mathbf{AX}$. Multichannel reconstruction techniques mostly provide better performances because of the joint support gained from multiple measurements.

Since $\mathbf{x_i}$'s may not be linearly independent from each other, rank of $\mathbf{X}$ satisfies $\text{Rank}(\mathbf{X}) \leq k$. When $\text{Rank}(\mathbf{X}) = 1$, all $\mathbf{x_i}$'s are multiples of each other resulting in no advantage to their joint processing. In this case, MMV and Single Measurement Vector (SMV) problems become identical. However, when $\text{Rank}(\mathbf{X})$ is large, we can benefit from the joint recovery. When $\mathbf{X}$ is generated at random, repeated columns are not likely to occur and it allows an improved recovery.

Theorem 3.4 *[35] Necessary and sufficient condition for the measurement* $\mathbf{Y} = \mathbf{AX}$ *to uniquely determine the jointly sparse matrix* $\mathbf{X}$ *is that*

$$|\mathit{supp}(\mathbf{X})| < \frac{\mathit{Spark}(\mathbf{A}) - 1 + \mathit{Rank}(\mathbf{X})}{2} \tag{3.21}$$

where $|\text{supp}(\mathbf{X})| = |\Lambda| = k$. A direct consequence of Theorem 3.4 is that matrices $\mathbf{X}$ with larger rank can be recovered from fewer measurements. When $\text{Rank}(\mathbf{X}) = k$ and $\text{Spark}(\mathbf{A})$ takes on its largest possible value which is equal to $m + 1$, condition (3.21) becomes $m \geq k + 1$. Therefore, in this best-case scenario, only $k + 1$ measurements per signal are necessary to ensure uniqueness. This is much lower than the value of $2k$ obtained via Theorem 3.1.

As in the Single Measurement Vector (SMV) case, there are two main approaches for solving MMV problems which are based on greedy methods and convex optimization. They are the natural expansions of the well known SMV versions of the algorithms and reduce to the SMV versions when applied to the measurements of a single sparse signal.

CHAPTER 4

APPLICATION OF SPARSE SIGNAL ESTIMATION ALGORITHMS ON CHANNEL ESTIMATION

Channel estimation methods, such as LS algorithm, lead to bandwidth inefficiency since it is necessary to use long training sequences. If the channel impulse response follows a sparse distribution, we can apply more efficient methods to acquire channel information. As a result, the training sequence length can be shortened compared with the linear estimation methods [36].

In this chapter, we present OMP and LASSO methods for the sparse channel estimation problems. In Section 4.1, we analyse the OMP algorithm and present simulation results for two different types of measurement matrices. In Section 4.2, we analyse the LASSO approach and finally in Section 4.3, we compare the performances of the two methods.

4.1 Channel Recovery via Orthogonal Matching Pursuit

4.1.1 Application of OMP Algorithm to Linear Models with Gaussian Measurement Matrix

In order to investigate the sparse channel recovery performance of the OMP method under a noisy environment, we have implemented the OMP algorithm presented in Section 3.7.1. Measurement matrix $\mathbf{A}$ is drawn to be an $m \times n$ Gaussian matrix with $\mathcal{N}(0,1)$ i.i.d. entries and $\mathbf{x} \in \mathbb{R}^n$ is drawn with sparsity level k where non-zero taps are normally distributed. Measurements are corrupted by additive white Gaussian noise $\mathbf{w}$ with $\mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$. Stopping criterion of the OMP algorithm is set to a fixed number k which assures that the recovered channel has k non-zero coefficients.

Figure 4.1 describes channel recovery percentage as a function of number of measurements m when $n = 256$ and SNR $= 10$ dB. Each curve represents a different sparsity level k. For each curve, we have run the OMP algorithm for 1000 independent trials. We assumed that channel recovery is achieved when l_2 norm of the estimation error

is smaller than 0.5 under noisy environment.

As seen from the figure, for the channels with the same length n and sparsity level k, an increase in m results in an increase in the recovery percentage. We can also conclude from the same figure that for the channels with the same n, when number of non-zeros (k) increase, more measurements are necessary to guarantee channel recovery.

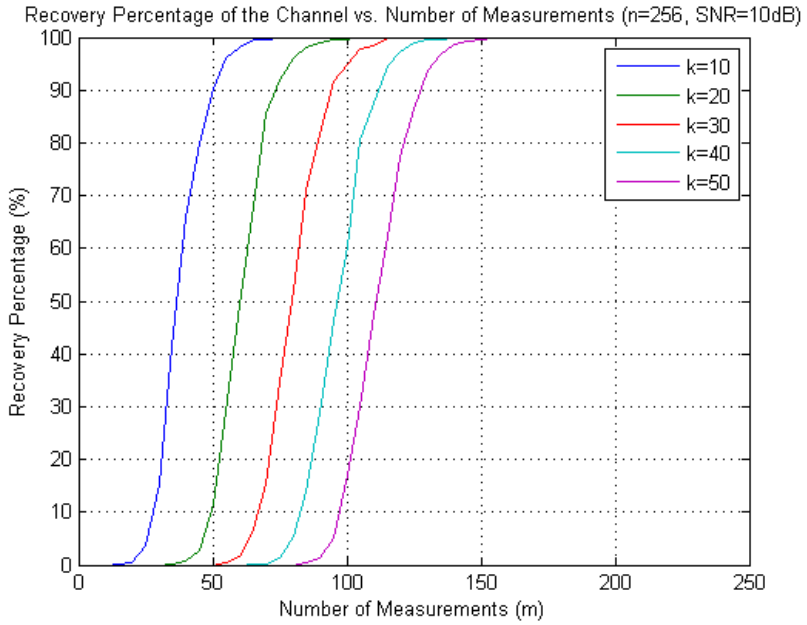

Figure 4.1: Recovery percentage of the channel as a function of m

Figure 4.2 displays the percentage of the channels recovered correctly as a function of k. It is clear from the figure that for a fixed n, m and SNR, an increase in k results in a decrease in the recovery percentage. Moreover, for a fixed n, SNR and k, the recovery probability increases when more measurements are taken. These two results support the previous results derived from Figure 4.1.

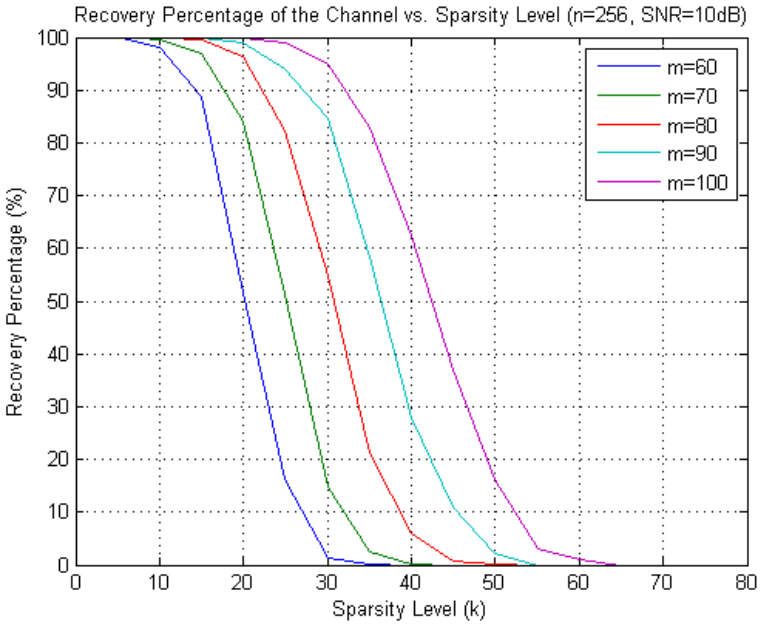

Figure 4.2: Recovery percentage of the channel as a function of k

Figure 4.3 represents the number of measurements m necessary to recover a k-sparse channel with probability 80% and 95% when $n = 256$ and SNR $= 10$ dB. This result supports the idea that number of measurements m should be proportional to the sparsity level k to ensure recovery with high probability.

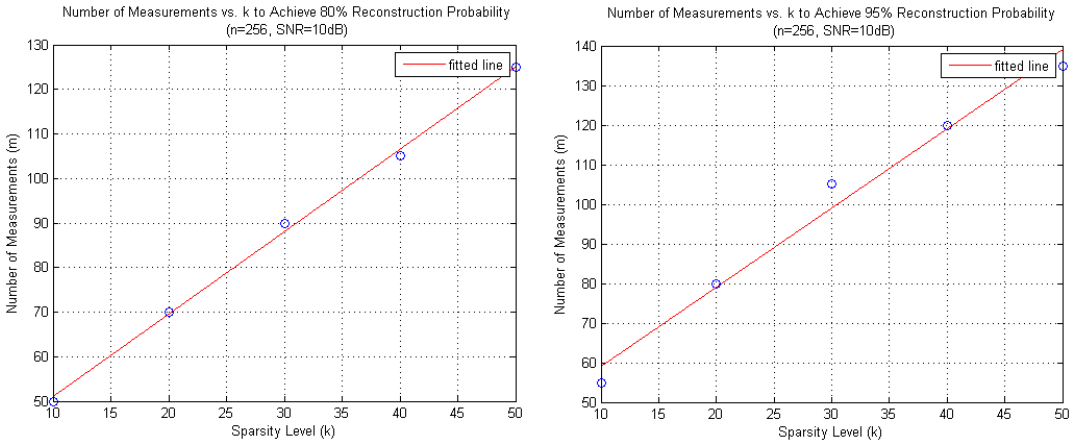

Figure 4.3: Number of measurements m, necessary to recover a k-sparse channel at least 80% and 95% of the time

Table 4.1 is provided to examine the relationship between n, k and m to achieve a recovery probability of 99% for two different channel lengths when SNR $= 10$ dB.

Table 4.1: Number of measurements, m necessary to recover a k-sparse channel at least 99% of the time in dimensions $n = 256$ and $n = 1024$ when SNR $= 10$ dB

Recovery probability of 99%					
n=256			**n=1024**		
k	**m**	**m/(k ln(n))**	**k**	**m**	**m/(k ln(n))**
10	69	1.24	10	84	1.21
20	90	0.81	20	115	0.83
30	108	0.65	30	144	0.69
40	128	0.58	40	175	0.63
50	147	0.53	50	204	0.59

Figure 4.4 represents the relationship between the channel recovery percentage and SNR where SNR $= \frac{(\mathbf{x}^T\mathbf{x})/n}{\sigma_n^2}$. As expected, high SNR results in an increase in recovery percentage and produces a better estimate with smaller Root Mean Square Error (RMSE).

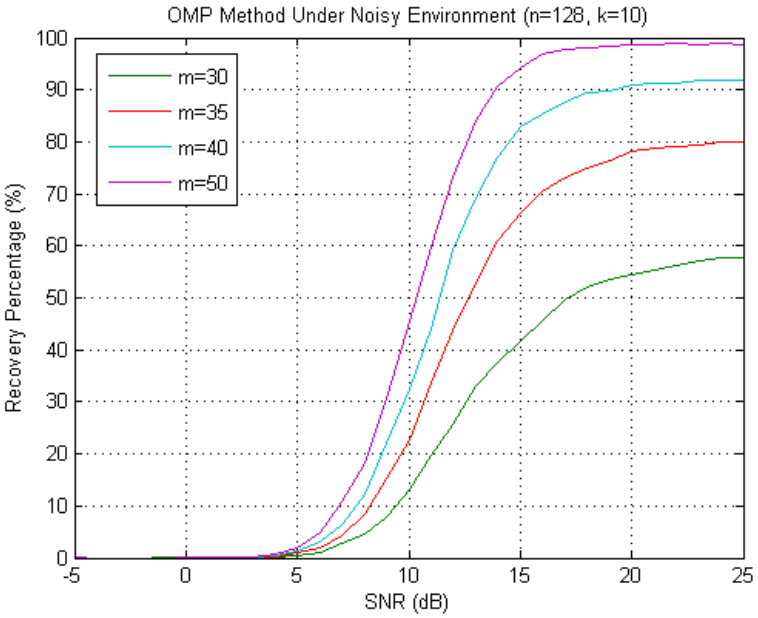

Figure 4.4: Recovery percentage vs. SNR (dB)

The relationship between RMSE and SNR is provided in Figure 4.5. It is clear from the figure that high SNR results in a decrease in RMSE.

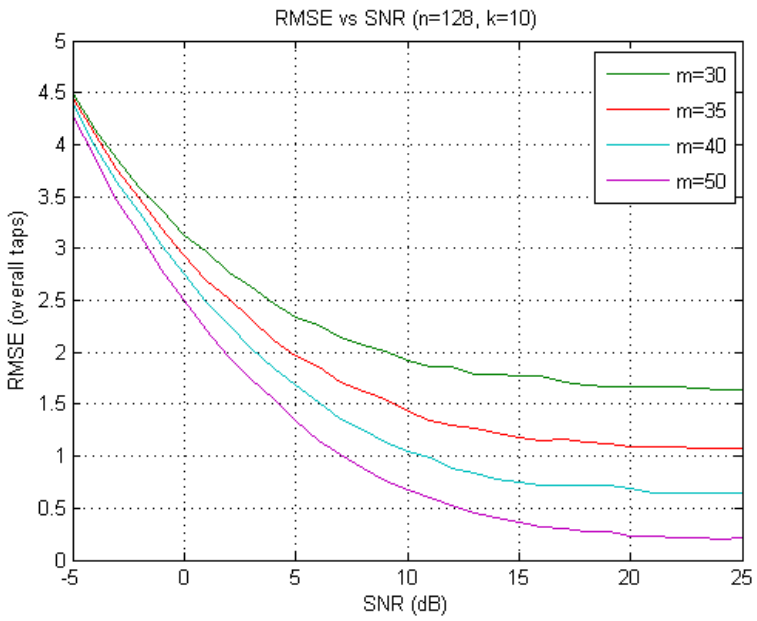

Figure 4.5: RMSE vs. SNR

In the above mentioned cases, OMP iteration stops after k-steps in order to ensure that the approximated sparse channel has a pre-specified sparsity level k. It is also possible to define a stopping rule other than the sparsity level k such that the maximum residual energy is limited to some value, say T. In other words, we could desire to find a sparse approximation $\hat{\mathbf{x}}$ satisfying the inequality $\|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}\|_2 < T$. Defining such a stopping rule is more relevant to de-noising in cases when the noise power is fixed and known.

MP algorithms differ from each other in the sense of their stopping rules or in choosing the relevant vector from the sensing matrix, $\mathbf{A}$. As stated before; in the OMP algorithm, next atom to be chosen is the one yielding the maximum correlation between residual and the atoms $\{\mathbf{a_i}\}_{i=1}^{n}$. Note that if there are several candidate atoms that show a relatively high correlation, the highest is chosen regardless of the proximity of the others to it. This brings us the randomization approach. In [37], Elad and Yavneh proposed an algorithm which they call RandOMP (Randomized OMP) in which the choice of the atom of $\mathbf{A}$ is randomized with a probability proportional to $|\mathbf{a_i}^T \mathbf{r}^t|$ where $i = \{1, 2, \ldots, n\}$ and t is the iteration number.

Running the proposed algorithm J_0 times leads us to J_0 solutions, $\{\hat{\mathbf{x}}_\mathbf{j}\}_{j=1}^{J_0}$. Common to all these approximations are the fact that,

- Their representation error $\|\mathbf{y} - \mathbf{S}\hat{\mathbf{x}}_{\text{RandOMP}}\|_2$ is below a threshold, T due to the stopping rule enforced,
- All of them tend to be relatively sparse due to the greedy nature of this algorithm that aims to decrease the residual energy, giving preference to those atoms that serve this goal better.

To compare the sparse channel recovery performance of the proposed RandOMP with the OMP, we run both algorithms $J_0 = 1000$ times independently and obtain the histogram graphs of the sparsity level of the estimates and the representation errors given in Figure 4.6. In both algorithms, iteration stops when we find an estimate $\hat{\mathbf{x}}$ satisfying the $\|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}\|_2 < T$ condition.

In OMP, all sparsity levels are observed to be fixed at 5 which is less than the sparsity level of the original channel. Indeed, this is an expected result as choosing the next atom is not randomized. Besides this, all representation errors are observed to vary around the threshold T.

In RandOMP, all the approximations are relatively sparse with sparsity level in the range $[5, 32]$ indicating that OMP produces the sparsest solution and all representation errors are observed to be slightly smaller than the threshold T.

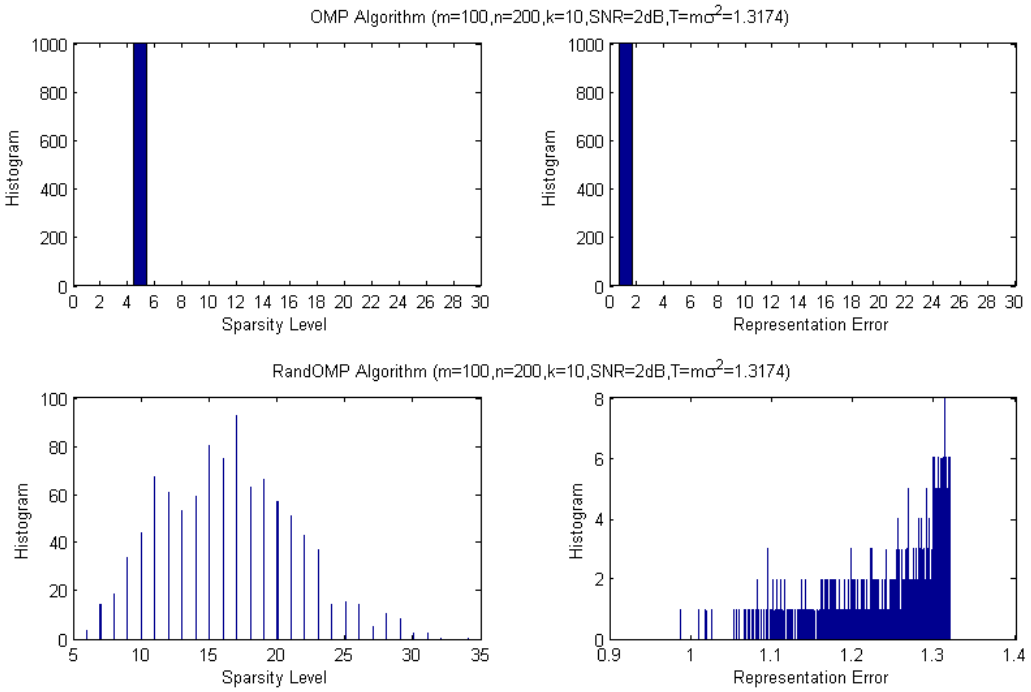

Figure 4.6: Comparison of OMP and RandOMP algorithms where representation error is defined as $\|\mathbf{y} - \mathbf{S}\hat{\mathbf{x}}\|_2$

Note that, for the RandOMP case we can use the formula:

$$\hat{\mathbf{x}}^{\text{ave}} = \frac{1}{J_0} \sum_{j=1}^{J_0} \hat{\mathbf{x}}_{\mathbf{j}}^{\text{RandOMP}}. \tag{4.1}$$

The averaged approximation $\hat{\mathbf{x}}^{\text{ave}}$ is no longer sparse, however its de-noising factor is better than OMP.

4.1.2 Application of OMP Algorithm to Linear Models with Toeplitz Measurement Matrix

Consider a broadband communication system over a sparse channel. The input-output system relation is described by:

$$y(t) = \int_0^{\tau_{max}} x(\tau)u(t-\tau)\,\mathrm{d}\tau + w(t) \tag{4.2}$$

where $y(t)$ and $u(t)$ denotes the received and transmitted waveforms, respectively, τ_{max} is defined as the maximum tap delay introduced by the channel and $w(t)$ is a zero-mean additive white Gaussian noise. Commonly, after sampling, such channels can be characterized as discrete, linear, time-invariant system as shown in Figure 4.7.

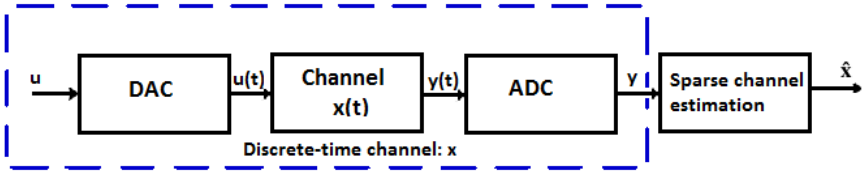

Figure 4.7: Framework on sparse channel estimation

Therefore, the discrete equivalent linear convolution system model is written in matrix form as follows:

$$\mathbf{y} = \mathbf{u} * \mathbf{x} + \mathbf{w} = \mathbf{A}\mathbf{x} + \mathbf{w}. \tag{4.3}$$

In this case, measurement matrix $\mathbf{A}$ is an $m \times n$ Toeplitz matrix given as follows:

$$\mathbf{A} = \begin{bmatrix} u[0] & 0 & 0 & \dots & 0 \\ u[1] & u[0] & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ u[l_t-1] & u[l_t-2] & u[l_t-3] & \dots & u[l_t-N] \\ 0 & u[l_t-1] & u[l_t-2] & \dots & u[l_t-N+1] \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & u[l_t-1] & u[l_t-2] \\ 0 & 0 & \dots & 0 & u[l_t-1] \end{bmatrix} \tag{4.4}$$

where $\mathbf{u} = \begin{bmatrix} u[0] & u[1] & \dots & u[l_t-1] \end{bmatrix}^T$ is the training sequence of length l_t and m is restricted to be

$$m = l_t + n - 1. \tag{4.5}$$

Observe from (4.5) that there is a direct relationship between n and m. Now, let's define the compression rate ρ as:

$$\rho = \frac{m}{n} = \frac{l_t + n - 1}{n} = 1 + \frac{1}{n}(l_t - 1). \tag{4.6}$$

It is clear from (4.6) that an increase in n results in a decrease in ρ producing a positive effect on channel recovery percentage.

In order to investigate the sparse channel recovery performance of the OMP method under a noisy environment, we have run the algorithm presented in Section 3.7.1. Measurement matrix $\mathbf{A}$ is drawn to be an $m \times n$ Toeplitz matrix provided in (4.4) and channel $\mathbf{x} \in \mathbb{R}^n$ is drawn with sparsity level k where non-zero taps are normally distributed. Measurements are corrupted by additive white Gaussian noise $\mathbf{w}$ with $\mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$.

Cross-correlation is the measure of agreement between two different codes. When the cross-correlation is zero, the codes are called orthogonal. In practice, the codes are not perfectly orthogonal, hence the cross-correlation between user codes introduces performance degradation. In the simulations, we tried to use the best suitable training sequences with suitable autocorrelation properties along with low cross-correlation values called minimum peak sidelobe (MPS) codes [38]. Finally, stopping criterion of the OMP algorithm is set to a fixed number k which assures that the recovered channel has k non-zero coefficients.

In the simulations, we assumed that channel recovery is achieved when l_2 norm of the estimation error is smaller than 0.5 under noisy environment. Recovery percentage of the channels as a function of n are plotted in Figure 4.8. Each curve in the figure represents a different sparsity level k. For each curve, we have run the OMP algorithm for 1000 independent trials.

Figure 4.8 displays that an increase in n (thus a decrease in ρ) results in an increase in the recovery percentage for a fixed l_t, SNR and k. In addition; it is clear that a longer channel is required in order to have the same recovery percentage when the sparsity level of the channel increases.

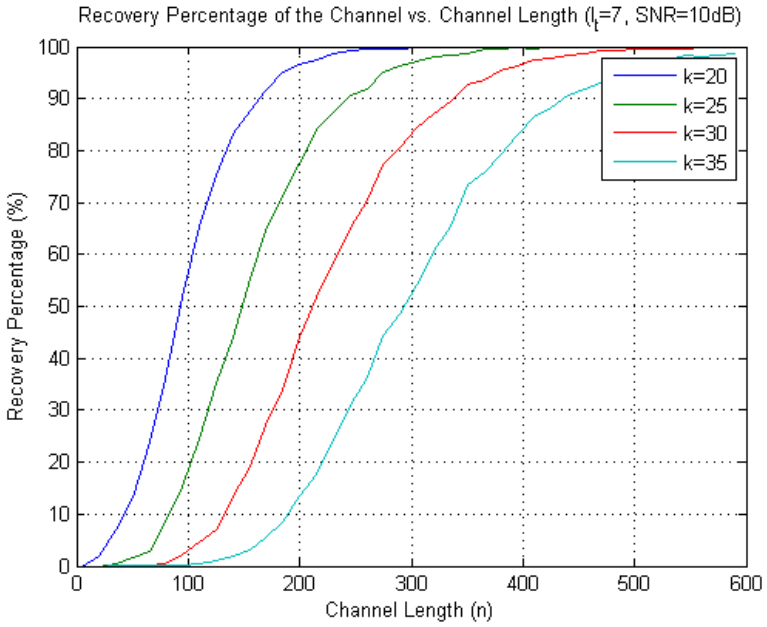

Figure 4.8: Recovery percentage of the channel as a function of n

Figure 4.9 represents the recovery percentage of the 1000 independent trials as a function of k. We can conclude from the figure that there is an indirect relationship between recovery probability and k for a fixed l_t, SNR, n and so is m. Figure also shows that for a fixed l_t, SNR and k, the recovery probability increases when more measurements are taken.

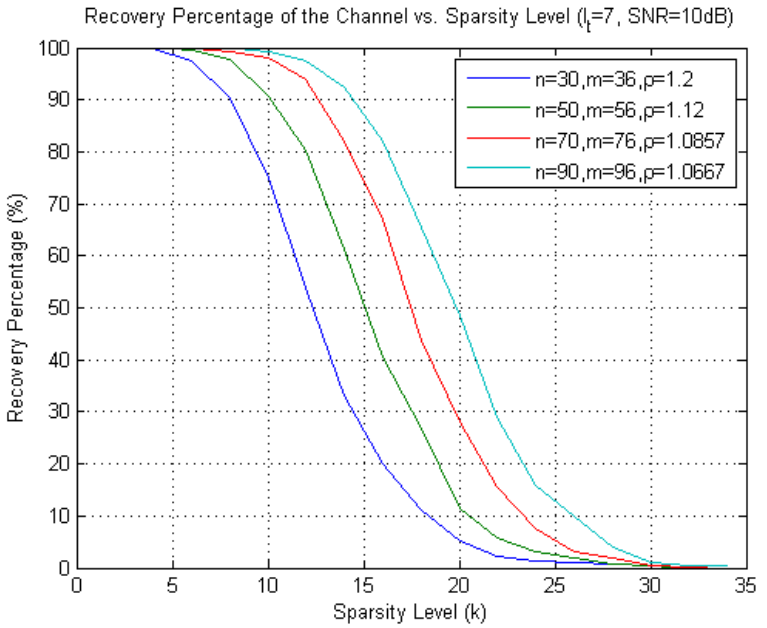

Figure 4.9: Recovery percentage of the channel as a function of k

Figure 4.10 represents the relationship between n and k necessary to achieve a recovery probability of 95% for a fixed l_t and SNR. The result supports the outcome of the

Figure 4.8 such that to achieve a predefined reconstruction probability for a fixed l_t and SNR, n is required to increase whenever k increases.

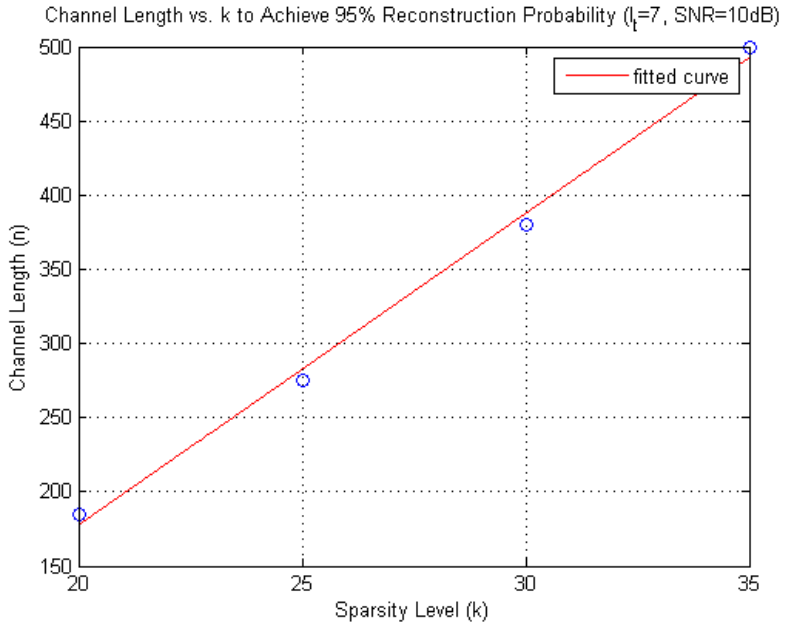

Figure 4.10: Channel size n, necessary to recover a k-sparse channel at least 95% of the time

From Figure 4.11, we can say that a decrease in compression rate ρ results in an increase in the recovery percentage for a fixed k, l_t and SNR supporting the previous results. The relationship between RMSE and SNR, where SNR $= \frac{(\mathbf{x}^T\mathbf{x})/n}{\sigma_n^2}$, is provided in Figure 4.12. As expected, high SNR forces RMSE to decrease.

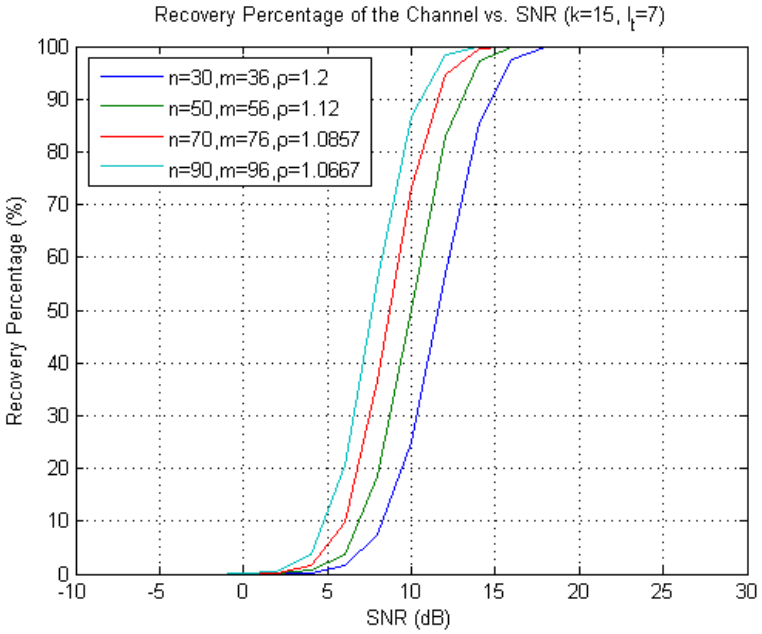

Figure 4.11: Recovery percentage vs. SNR

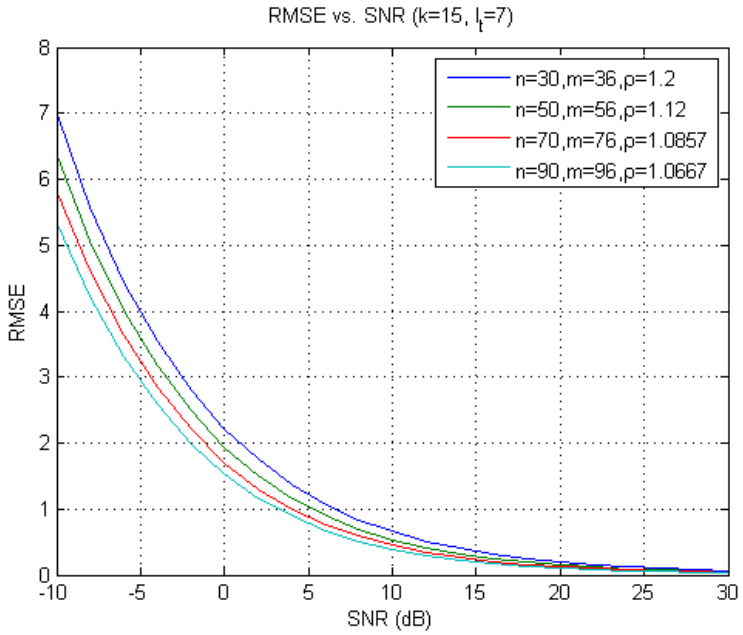

Figure 4.12: RMSE vs. SNR

For a fixed k, SNR and n; increasing length of training sequence l_t which is the case in Figure 4.13, results in an increase in the recovery percentage of the channel with small RMSE so that one should choose longer training sequences to guarantee channel recovery.

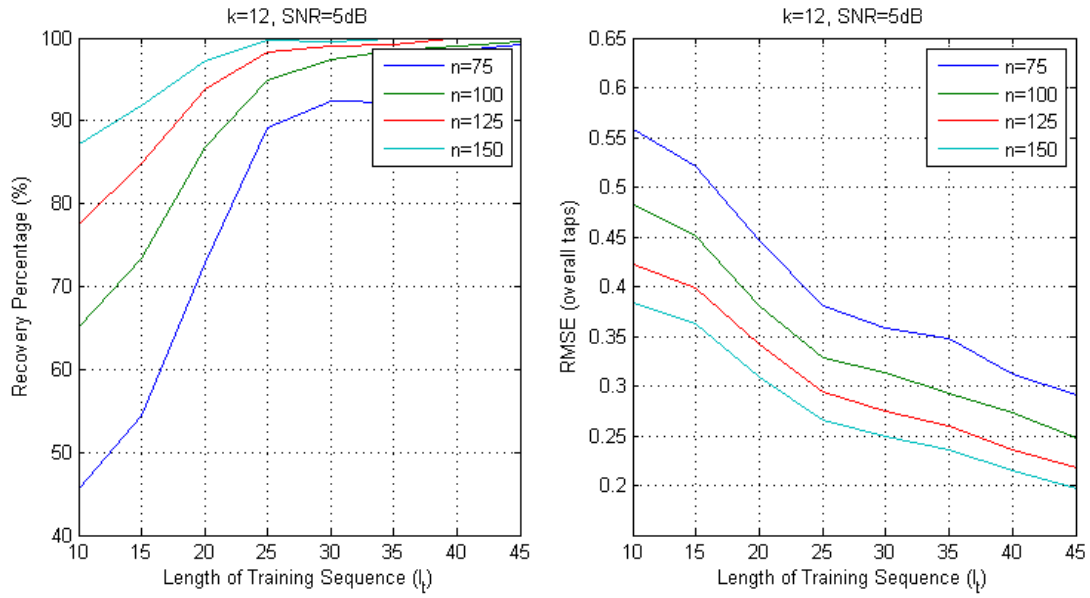

Figure 4.13: Effect of l_t on recovery percentage and RMSE

4.2 Channel Recovery Via LASSO

In Section 3.7.2, we have presented the theory behind LASSO approach and mentioned that it is an optimization principle rather than an algorithm. In this section, we present the simulation results considering the LASSO approach.

In the simulations, measurement matrix $\mathbf{A}$ is drawn to be an $m \times n$ Gaussian matrix with $\mathcal{N}(0,1)$ i.i.d entries and $\mathbf{x} \in \mathbb{R}^n$ is a k-sparse channel where non-zero taps are drawn from a uniform distribution on $[-2,-1] \bigcup [1,2]$. Measurements are corrupted by additive white Gaussian noise $\mathbf{w}$ with $\mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I})$. For each curve, 2500 independent Monte Carlo trials have been conducted.

First graph of Figure 4.14 describes l_0 and l_1 norm of the channel estimate as a function of λ when $m = 50$, $n = 40$, $k = 4$ and SNR $= 10$ dB. According to the LASSO method $\|\hat{\mathbf{x}}\|_1$ should be less than or equal to λ, so a reference line is provided to see how close the l_1 norm of the channel estimate is to λ. $\|\hat{\mathbf{x}}\|_0$ is calculated by considering the estimated channel taps satisfying condition $|\hat{x_i}| > 0.1$, where $\hat{x_i}$ is the estimate of the i^{th} component of $\mathbf{x}$ for $i = \{1, 2, \ldots, n\}$. The graph demonstrates that an increase in λ promotes an increase in $\|\hat{\mathbf{x}}\|_0$ only up to a point.

Because of the channel distribution, l_1 norm of the true channel is in between $|$minimum possible value of the channel taps$| \times k = 1 \times 4 = 4$ and $|$maximum possible value of the channel taps$| \times k = 2 \times 4 = 8$. Thus, we expect the channel estimate to have similar property. That is the reason of $\|\hat{\mathbf{x}}\|_1 \leq \lambda$ condition in the LASSO method disappears and it tries to solve LS estimate for $\lambda > 8$. As a result of this, l_0 and l_1 norm of the channel estimate stabilizes after that point on. It is also worth to declare that because of the differences between l_0 and l_1 norm, it is possible for the LASSO method not being able to find the sparsest possible approximation.

Second graph of Figure 4.14 describes the relationship between RMSE and λ when $m = 50$, $n = 40$, $k = 4$ and SNR $= 10$ dB. Since non-zero channel taps are drawn from a uniform distribution on $[-2, -1] \bigcup [1, 2]$, l_1 norm of the true channel is likely to be $(|$minimum possible value of the channel taps$| + |$maximum possible value of the channel taps$|)/2 \times k = (1 + 2)/2 \times 4 = 6$. That is the reason of having high RMSE for $\lambda < 6$. Therefore, an approximation would have to be obtained while deciding the value for λ.

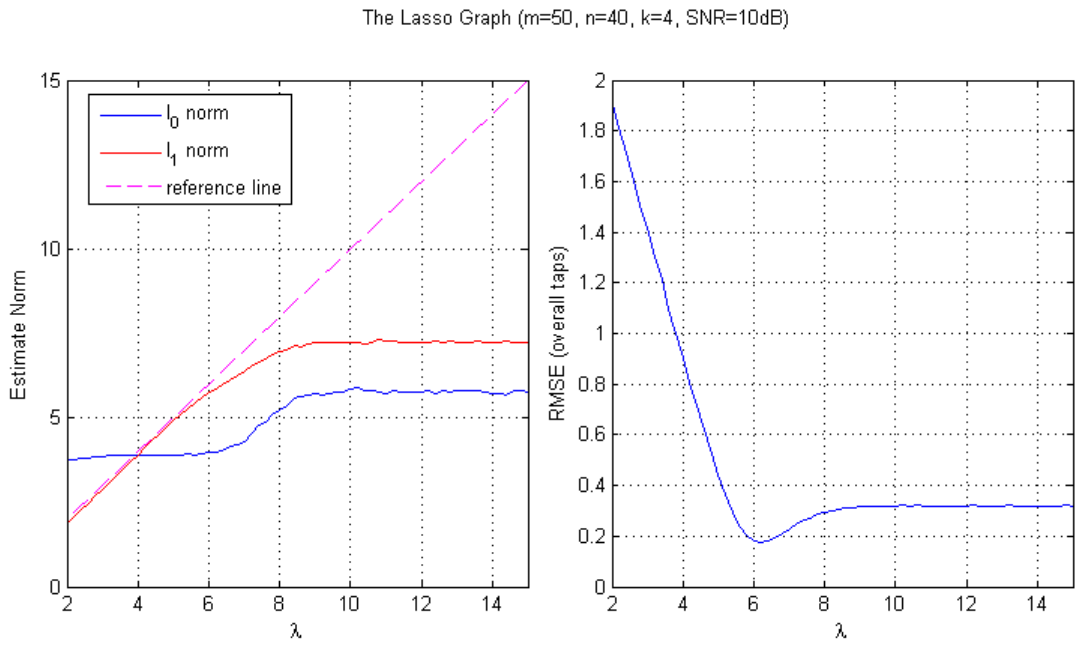

Figure 4.14: Effect of λ on the estimates

SNR effect on l_0 and l_1 norm of the channel estimate is shown in the first graph of Figure 4.15 when $m = 50$, $n = 40$, $k = 4$ and $\lambda = 12$. $\|\hat{\mathbf{x}}\|_0$ is calculated by considering the elements satisfying condition $|\hat{x_i}| > 0.1$. As seen from the graph, both norms tend to decrease as SNR increases and l_1 norm of the channel estimate is always below the reference line which intersects the y-axis at 12 which is equal to the λ value. Moreover, for the given conditions, sparsity level of the channel estimate approaches the true sparsity level for SNR > 10 dB.

The relationship between RMSE ans SNR is provided in the second graph of Figure 4.15. It is clear that high SNR produces a better estimate with smaller RMSE. We have also conducted simulations for smaller λ's and obtained similar results for both graphs.

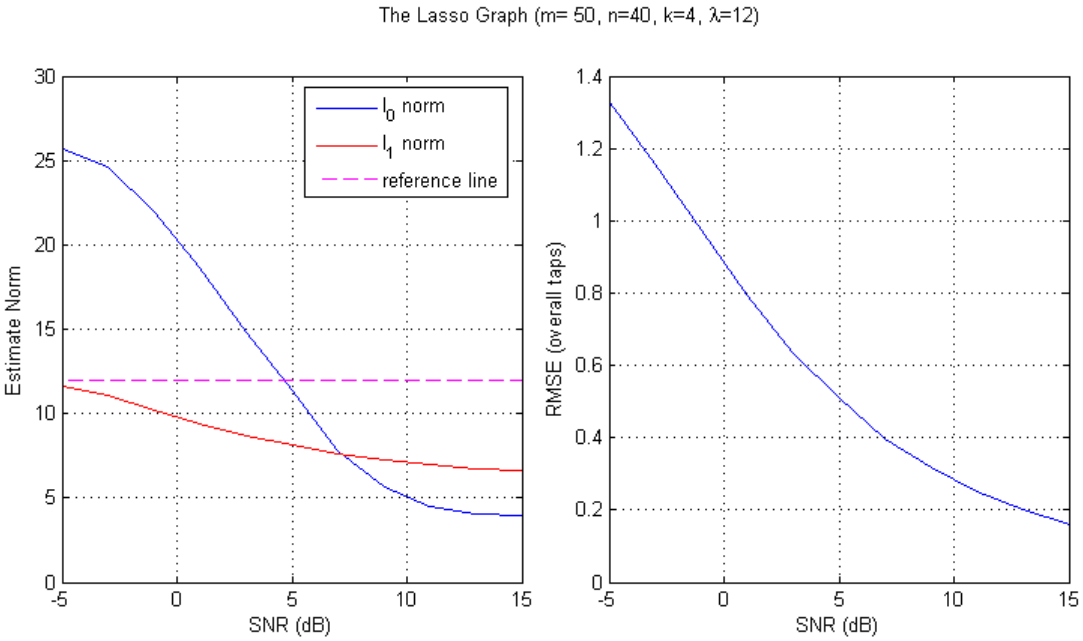

Figure 4.15: Effect of SNR on l_0 & l_1 norm of the channel estimates and RMSE

4.3 Comparison of OMP and LASSO

This section offers a brief comparison between known results for the greedy algorithm and results for the convex relaxation approach.

In the previous sections, we have tried to estimate sparse channel $\mathbf{x} \in \mathbb{R}^n$ based on the linear models

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{4.7}$$

where $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{A}$ is the known measurement matrix and $\mathbf{w}$ is additive white Gaussian noise, $\mathcal{N}(\mathbf{0}, \sigma_n^2\mathbf{I})$.

In many problems, the sparsity level k is not known a-priori and must be detected as a part of the estimation process. In OMP, the sparsity level of estimated channel is precisely the number of iterations conducted before the algorithm terminates. Thus, a good stopping condition is needed for a reliable estimation. The effect of different stopping conditions on RMSE is shown in Figure 4.16 where RMSE and SNR are

defined as

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\|\mathbf{x}-\hat{\mathbf{x}}_\mathbf{m}\|_2^2}$$
$$\text{SNR} = \frac{(\mathbf{x}^T\mathbf{x})/n}{\sigma_n^2} \tag{4.8}$$

and where M is the number of independent trials.

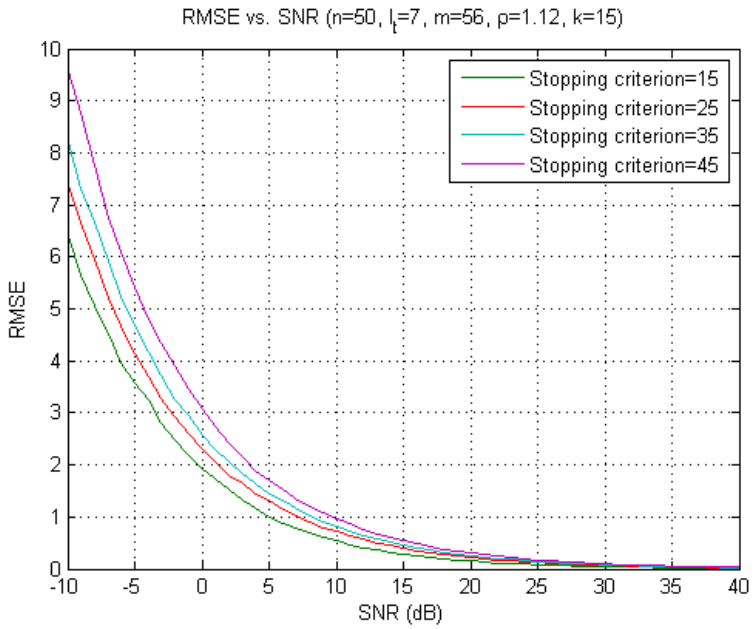

Figure 4.16: Performance of OMP algorithm with different stopping rules ($M = 1000$, $\mathbf{A}$ is drawn to be a Toeplitz structured matrix which uses length-7 MPS code as the training sequence and non-zero taps of the channel are normally distributed)

Tropp and Gilbert [30] investigated the performance of OMP for a set of $m \times n$ ($m \ll n$) sensing matrices. They claim that OMP can recover the channels with high probability provided that the number of measurements is proportional to the sparsity level of the channel such that $m \approx k\ \ln(n)$. However, OMP's condition on sensing matrix given in [30] is more restrictive than the restricted isometry property. Kunis and Rauhut [39] claimed that the first iteration in OMP is likely to identify the correct column from the sensing matrix given $\mathcal{O}(k\ \ln(n))$ measurements of a k-sparse channel in $\mathbb{R}^n$. Unfortunately, because of the unavoidable correlation between the columns of the sensing matrix, it is difficult to analyse subsequent iterations of the algorithm. OMP can acquire non-zero taps with high probability, but with high probability fails to estimate all non-zero taps of a channel leading to instability.

On the other hand, LASSO tries to solve the quadratic program

$$\hat{\mathbf{x}} = \text{argmin}_\mathbf{x}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 \quad \text{subject to } \|\mathbf{x}\|_1 \leq \lambda \tag{4.9}$$

where $\lambda \geq 0$ is an algorithm parameter that trades the prediction error with the sparsity of the solution. The convex program method is based on linear programming however, it has disadvantages in computational cost and implementation complexity.

Figure 4.17 provides the overall taps estimation error using RMSE standard for MMSE, LS, OMP and LASSO. In the simulation, measurement matrix $\mathbf{A}$ is drawn to be an $m \times n$ Toeplitz structured matrix which uses length-7 MPS code as the training sequence. $\mathbf{x} \in \mathbb{R}^n$ is a k-sparse channel where non-zero taps are drawn from a uniform distribution on $[-2,-1] \bigcup [1,2]$. Stopping criterion in OMP algorithm is set to channel sparsity level k. Measurements are corrupted by additive white Gaussian noise $\mathbf{w}$ with $\mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I}_{m\times m})$.

Parameter settings are as follows: $n = 60$, $l_t = 7$, $m = 66$, $k = 6$ and estimation constraint in LASSO is taken as $\|\hat{\mathbf{x}}\|_1 \leq \lambda_{LASSO} = 10$ and 100 separately. For each curve, 1000 independent Monte Carlo trials have been conducted.

From the figure, we can conclude that OMP performance is better than LASSO, MMSE and LS for the given parameter settings. Note that performance of the OMP algorithm depends on the stopping criteria used for halting the iterations and performance of the LASSO method depends on the selection of the parameter λ. A-priori knowledge of the channel distribution or channel sparsity is not taken into consideration in LS so that its performance is lower than MMSE, LASSO with $\lambda_{LASSO} = 10$ and OMP. On the other hand, if we choose λ_{LASSO} large enough such that the effect of the constraint in LASSO disappears, then LS and LASSO estimates coincide.

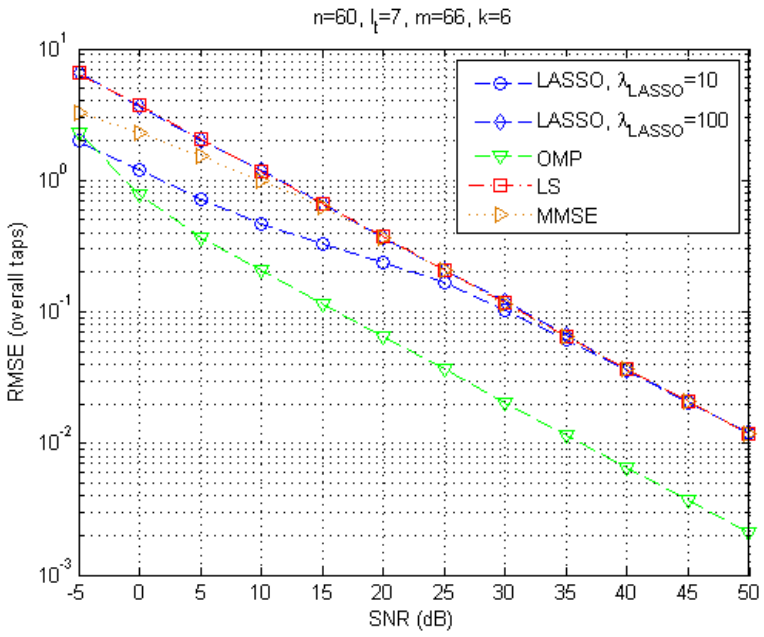

Figure 4.17: Performance comparison via RMSE of different channel estimation methods at different SNR's

While the optimization in LASSO is convex, the running time of LASSO is signifi-

cantly longer than the OMP unless $\mathbf{A}$ has some particular structure [30]. Dependence of the running times of the two algorithms on the number of columns of $\mathbf{A}$, number of rows of $\mathbf{A}$ and sparsity level of the channel presented in [1] are provided in Figures 4.18, 4.19 and 4.20.

Figure 4.18 shows the run time of the algorithms as a function of number of rows m. As seen from the graph, LASSO takes the maximum time which increases linearly with m.

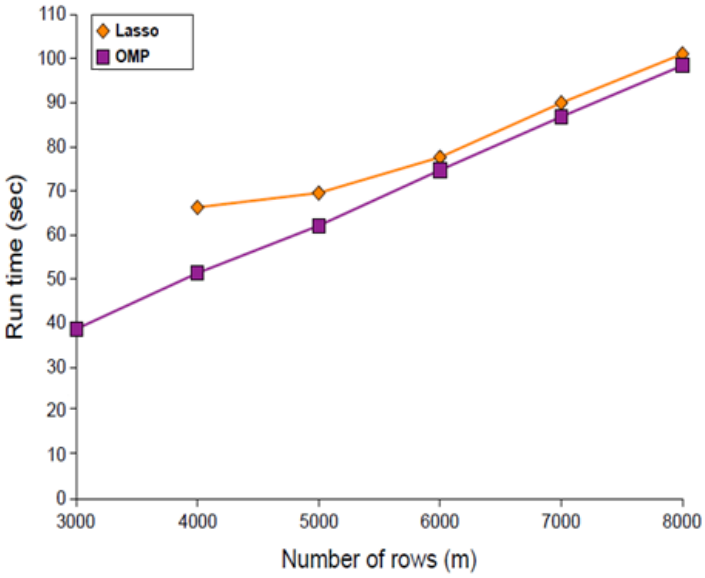

Figure 4.18: Dependence of running times on the number of rows ($n = 8000$, $k = 500$), [1]

Figure 4.19 displays the run time of the algorithms as a function of n. The algorithms seem to scale linearly with n. The output of each algorithm is compared with the correct solution $\mathbf{x}$ and authors in [1] say that LASSO did not give correct results for $n > 8000$ and hence its run time is omitted from the graph. Finally, Figure 4.20 shows the run time of the algorithms as a function of the sparsity level.

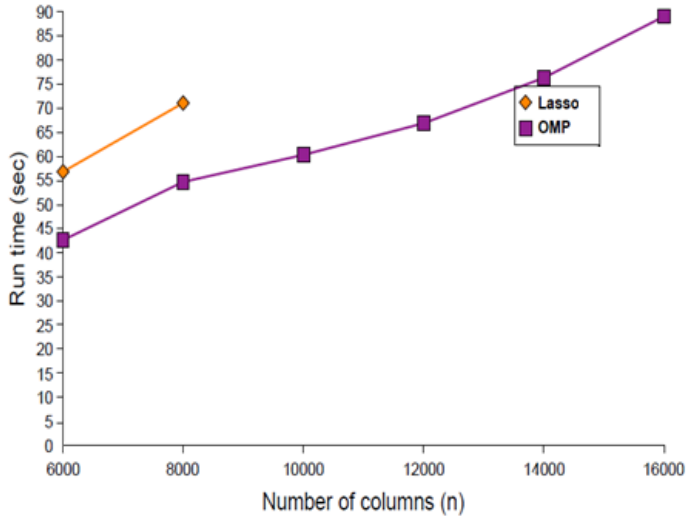

Figure 4.19: Dependence of running times on the number of columns ($m = 4000$, $k = 500$), [1]

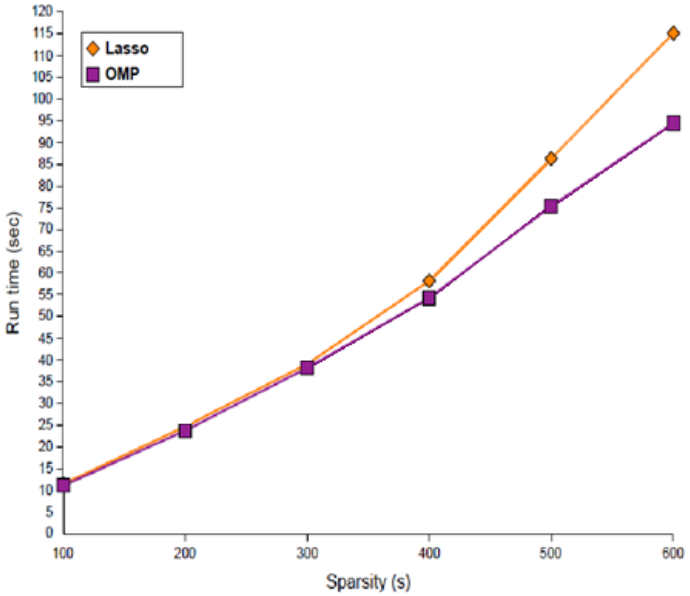

Figure 4.20: Dependence of running times on the sparsity levels, [1]

CHAPTER 5

SPARSE CHANNEL ESTIMATION FOR BAYESIAN LINEAR MODELS

In chapter 4, we have investigated some of the commonly used algorithms to recover sparse channels. In this chapter, we present Bayesian approach for modelling sparse channels and examine sparse channel estimation problem for Bayesian linear models.

5.1 Posterior Density Calculation for Mixture of Gaussians

In this section, we examine different methods for the calculation of the posterior density function of x given the observation y denoted as $\mathrm{p}(x|y)$ where x is a mixture of Gaussians having probability density function given below:

$$\mathrm{p}(x) = p_1\mathcal{N}(x;\mu_1,\gamma_1^2) + p_2\mathcal{N}(x;\mu_2,\gamma_2^2) \tag{5.1}$$

where $\mathcal{N}(x;\mu_i,\gamma_i^2) = \frac{1}{\sqrt{2\pi\gamma_i^2}}\,\mathrm{e}^{-\frac{(x-\mu_i)^2}{2\gamma_i^2}}$ for $i = \{1,2\}$ is the univariate Gaussian distribution with the component probabilities p_1 and p_2 where $p_1 + p_2 = 1$.

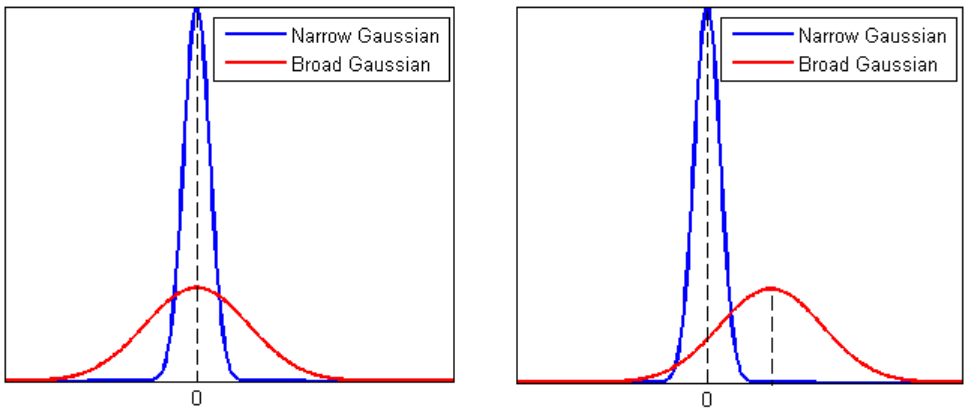

Figure 5.1: Mixture of Gaussians

Since the signal in our concern is assumed to be sparse, it is reasonable to think that

the narrow Gaussian models the smaller peaks with zero-mean whereas the broad one models the higher peaks. Some possible distributions for the one dimensional case are provided in Figure 5.2.

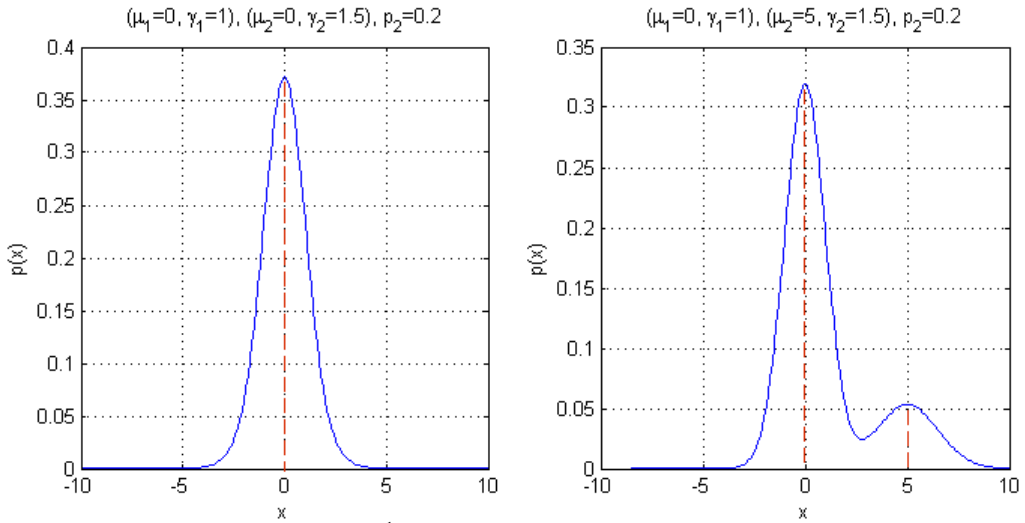

Figure 5.2: Pdf for $x \in \mathbb{R}^1$ where x is composed of mixture of Gaussians

For simplicity, we assume that the random variable x is observed under linear model with additive white Gaussian noise with distribution $\mathcal{N}(w; 0, \sigma_n^2)$ such that

$$y = x + w. \tag{5.2}$$

Method 1: Posterior Density Calculation Through An Algebraic Approach

In the calculation of $\mathrm{p}(x|y)$, we use the following formulas:

$$\mathcal{N}(x; \mu_1, \gamma_1^2)\mathcal{N}(x; \mu_2, \gamma_2^2) = \mathcal{N}(\mu_1; \mu_2, \gamma_1^2 + \gamma_2^2)\mathcal{N}(x; \mu_{12}, \gamma_{12}^2) \tag{5.3}$$

where

$$\begin{aligned} \frac{1}{\gamma_{12}^2} &= \frac{1}{\gamma_1^2} + \frac{1}{\gamma_2^2} \\ \mu_{12} &= \gamma_{12}^2 \left(\frac{\mu_1}{\gamma_1^2} + \frac{\mu_2}{\gamma_2^2} \right). \end{aligned} \tag{5.4}$$

Note that $\mathcal{N}(\mu_1; \mu_2, \gamma_1^2 + \gamma_2^2)$ is just a scaling constant. Therefore, the product of two Gaussian functions having independent variable x yields a scaled Gaussian function of x [40].

As in the case of multiplication of two Gaussian function, division of two Gaussian functions can also be written in the usual Gaussian form with a scaling constant such that:

$$\mathcal{N}(x; \mu_1, \gamma_1^2)/\mathcal{N}(x; \mu_2, \gamma_2^2) = \frac{\gamma_2^2\, \mathcal{N}(x; \mu_{12}, \gamma_{12}^2)}{(\gamma_2^2 - \gamma_1^2)\, \mathcal{N}(\mu_1; \mu_2, \gamma_2^2 - \gamma_1^2)} \tag{5.5}$$

where

$$\begin{aligned}\frac{1}{\gamma_{12}^2} &= \frac{1}{\gamma_1^2} + \frac{1}{\gamma_2^2} \\ \mu_{12} &= \gamma_{12}^2 \left(\frac{\mu_1}{\gamma_1^2} - \frac{\mu_2}{\gamma_2^2} \right).\end{aligned} \tag{5.6}$$

The posterior density for the linear model given in (5.2) can be expressed in terms of the likelihood function and the prior distribution as follows:

$$\begin{aligned}\mathrm{p}(x|y) &= \frac{\mathrm{p}(x,y)}{\mathrm{p}(y)} \\ &= \frac{\mathrm{p}(y|x)\mathrm{p}(x)}{\int_x \mathrm{p}(x,y)\,\mathrm{d}x} \\ &= \frac{\mathrm{p}(y|x)\mathrm{p}(x)}{\int_x \mathrm{p}(y|x)\mathrm{p}(x)\,\mathrm{d}x} \\ &= \frac{\mathcal{N}(y;x,\sigma_n^2)\mathrm{p}(x)}{\int_x \mathcal{N}(y;x,\sigma_n^2)\mathrm{p}(x)\,\mathrm{d}x}.\end{aligned} \tag{5.7}$$

The numerator of the posterior can be rewritten as

$$\begin{aligned}\mathcal{N}(y;x,\sigma_n^2)\mathrm{p}(x) &= \mathcal{N}(x;y,\sigma_n^2)\mathrm{p}(x) \\ &= \mathcal{N}(x;y,\sigma_n^2)\left(p_1\mathcal{N}(x;\mu_1,\gamma_1^2) + p_2\mathcal{N}(x;\mu_2,\gamma_2^2)\right).\end{aligned} \tag{5.8}$$

From (5.3), it should be noted that

$$\mathcal{N}(x;y,\sigma_n^2)\mathcal{N}(x;\mu_i,\gamma_i^2) = \mathcal{N}(y;\mu_i,\sigma_n^2+\gamma_i^2)\mathcal{N}(x;\hat{\mu}_i,\hat{\gamma}_i^2) \tag{5.9}$$

where $i = \{1,2\}$, $\frac{1}{\hat{\gamma}_i^2} = \frac{1}{\sigma_n^2} + \frac{1}{\gamma_i^2}$ and $\hat{\mu}_i = \hat{\gamma}_i^2\left(\frac{y}{\sigma_n^2} + \frac{\mu_i}{\gamma_i^2}\right)$. It should be noted that $\hat{\mu}_i$ and $\hat{\gamma}_i^2$ are the mean and variance of the mixture components. Hence, the numerator of $\mathrm{p}(x|y)$ becomes

$$\mathcal{N}(y;x,\sigma_n^2)\mathrm{p}(x) = p_1\mathcal{N}(y;\mu_1,\sigma_n^2+\gamma_1^2)\mathcal{N}(x;\hat{\mu}_1,\hat{\gamma}_1^2)+p_2\mathcal{N}(y;\mu_2,\sigma_n^2+\gamma_2^2)\mathcal{N}(x;\hat{\mu}_2,\hat{\gamma}_2^2). \tag{5.10}$$

The denominator of $\mathrm{p}(x|y)$ given in (5.7) is a constant obtained by integrating (5.10) with respect to variable x. Since each $\mathcal{N}(x;\hat{\mu}_i,\hat{\gamma}_i^2)$ term has unit area, the denominator can be simplified to

$$\int_x \mathcal{N}(y;x,\sigma_n^2)\mathrm{p}(x)\,\mathrm{d}x = p_1\mathcal{N}(y;\mu_1,\sigma_n^2+\gamma_1^2) + p_2\mathcal{N}(y;\mu_2,\sigma_n^2+\gamma_2^2). \tag{5.11}$$

Combining these results, we have the following

$$\begin{aligned}\mathrm{p}(x|y) &= \frac{\mathcal{N}(y; x, \sigma_n^2)\mathrm{p}(x)}{\int_x \mathcal{N}(y; x, \sigma_n^2)\mathrm{p}(x)\,\mathrm{d}x} \\ &= \frac{p_1\mathcal{N}(y; \mu_1, \sigma_n^2 + \gamma_1^2)}{p_1\mathcal{N}(y; \mu_1, \sigma_n^2 + \gamma_1^2) + p_2\mathcal{N}(y; \mu_2, \sigma_n^2 + \gamma_2^2)}\mathcal{N}(x; \hat{\mu}_1, \hat{\gamma}_1^2) \\ &\qquad + \frac{p_2\mathcal{N}(y; \mu_2, \sigma_n^2 + \gamma_2^2)}{p_1\mathcal{N}(y; \mu_1, \sigma_n^2 + \gamma_1^2) + p_2\mathcal{N}(y; \mu_2, \sigma_n^2 + \gamma_2^2)}\mathcal{N}(x; \hat{\mu}_2, \hat{\gamma}_2^2).\end{aligned} \tag{5.12}$$

By defining

$$\hat{p}_i = \frac{p_i\mathcal{N}(y; \mu_i, \sigma_n^2 + \gamma_i^2)}{p_1\mathcal{N}(y; \mu_1, \sigma_n^2 + \gamma_1^2) + p_2\mathcal{N}(y; \mu_2, \sigma_n^2 + \gamma_2^2)} \tag{5.13}$$

for $i = \{1, 2\}$, we can express the posterior density as follows:

$$\mathrm{p}(x|y) = \hat{p}_1\mathcal{N}(x; \hat{\mu}_1, \hat{\gamma}_1^2) + \hat{p}_2\mathcal{N}(x; \hat{\mu}_2, \hat{\gamma}_2^2). \tag{5.14}$$

From the last relation, it is clear that the posterior distribution is a mixture of Gaussians with the updated parameters.

Method 2: Posterior Density Calculation Through A Latent Variable Approach

The random variable x can be written as follows:

$$x = \begin{cases} x_1 & \theta = 1 \\ x_2 & \theta = 2 \end{cases} \tag{5.15}$$

where $x \sim \mathcal{N}(x; \mu_i, \gamma_i^2)$ for $i = \{1, 2\}$ and θ is the latent variable which is independent of x_i's. It can be noted that the density of random variable x appearing in Methods 1 and 2 is indeed the same, i.e. the mixture of Gaussians.

The goal is to calculate the posterior density $\mathrm{p}(x|y) = \frac{\mathrm{p}(y|x)\mathrm{p}(x)}{\mathrm{p}(y)}$. To do that, we calculate the posterior for $\mathrm{p}(x, \theta|y)$ and then marginalize over the random variable θ.

$$\begin{aligned}\mathrm{p}(x|y) &= \sum_{\theta=1}^{2} \mathrm{p}(x, \theta|y) \\ &= \sum_{\theta=1}^{2} \frac{\mathrm{p}(y|x, \theta)\mathrm{p}(x, \theta)}{\mathrm{p}(y)} \\ &= \sum_{\theta=1}^{2} \frac{\mathrm{p}(y|x)\mathrm{p}(x, \theta)}{\mathrm{p}(y)} \\ &= \sum_{\theta=1}^{2} \frac{\mathrm{p}(y|x)\mathrm{p}(x|\theta)p(\theta)}{\mathrm{p}(y)}.\end{aligned} \tag{5.16}$$

It can be noted that

$$\begin{aligned}\sum_{\theta=1}^{2}\frac{\mathrm{p}(y|x)\mathrm{p}(x|\theta)p(\theta)}{\mathrm{p}(y)} &= \sum_{i=1}^{2}\frac{\mathcal{N}(x;y,\sigma_n^2)\mathcal{N}(x;\mu_i,\gamma_i^2)p_i}{\mathrm{p}(y)} \\ &= \sum_{i=1}^{2}\frac{\mathcal{N}(y;\mu_i,\sigma_n^2+\gamma_i^2)p_i}{\mathrm{p}(y)}\mathcal{N}(x;\hat{\mu}_i,\hat{\gamma}_i^2)\end{aligned} \tag{5.17}$$

where we have used the definition given in (5.3) for Method 1. It is worth to mention that the final result of (5.17) coincides with the final result of Method 1.

The mean and variance of the posterior distribution can be written as follows:

$$\begin{aligned}\mathrm{E}[x|y] &= \hat{p}_1\hat{\mu}_1 + \hat{p}_2\hat{\mu}_2 \\ &= \bar{x} \\ \mathrm{var}[x|y] &= \mathrm{E}[(x-\bar{x})^2|y] \\ &= \mathrm{E}[x^2|y] - \bar{x}^2 \\ &= \mathrm{E}_\theta[\mathrm{E}_x[x^2|y,\theta]] - \bar{x}^2 \\ &= \hat{p}_1\mathrm{E}_x[x^2|y,\theta=1] + \hat{p}_2\mathrm{E}_x[x^2|y,\theta=2] - \bar{x}^2 \\ &= \hat{p}_1(\hat{\gamma}_1^2+\hat{\mu}_1^2) + \hat{p}_2(\hat{\gamma}_2^2+\hat{\mu}_2^2) - \bar{x}^2 \\ &= \hat{p}_1\hat{\gamma}_1^2 + \hat{p}_2\hat{\gamma}_2^2 + [\hat{p}_1\hat{\mu}_1^2 + \hat{p}_2\hat{\mu}_2^2 - (\hat{p}_1\hat{\mu}_1+\hat{p}_2\hat{\mu}_2)^2].\end{aligned} \tag{5.18}$$

The mean and variance of the posterior distribution are the only two parameters required to approximate the posterior distribution with a Gaussian distribution. As noted later, this fact is utilized in the expectation propagation method which is a method for approximate posterior calculation.

Extension to Mixture of Gaussian Random Vectors

We first present a short brief for finding out the posterior distribution of a random vector $\mathbf{x}$ which is observed under a linear system model with additive white Gaussian noise $\mathbf{w}$. For this purpose, let's consider the linear system below:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{5.19}$$

where $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{x} \in \mathbb{R}^n$ is known to have Gaussian distribution $\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_\mathbf{x},\mathbf{C}_\mathbf{x})$ and $\mathbf{w}$ is assumed to be independent of $\mathbf{x}$ with distribution $\mathcal{N}(\mathbf{w};\mathbf{0},\mathbf{C}_\mathbf{n})$ where $\mathbf{C}_\mathbf{n} = \sigma_n^2\mathbf{I}$.

In this system model, $\mathbf{x}$ and $\mathbf{w}$ are Gaussian, then $\mathbf{y}$ is also Gaussian, that is

$$\mathbf{y} \sim \mathcal{N}(\mathbf{y};\boldsymbol{\mu}_\mathbf{y},\mathbf{C}_\mathbf{y}) \tag{5.20}$$

where

$$\begin{aligned}\boldsymbol{\mu}_\mathbf{y} &= \mathbf{A}\boldsymbol{\mu}_\mathbf{x} \\ \mathbf{C}_\mathbf{y} &= \mathbf{A}\mathbf{C}_\mathbf{x}\mathbf{A}^T + \mathbf{C}_\mathbf{n}.\end{aligned} \tag{5.21}$$

The conditional density $\mathrm{p}(\mathbf{x}|\mathbf{y})$ is also Gaussian which allows us to interpret $\mathbf{x}|\mathbf{y}$ such that

$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}(\mathbf{x}|\mathbf{y}; \boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}}, \mathbf{C}_{\mathbf{x}|\mathbf{y}}) \tag{5.22}$$

where

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}} &= \boldsymbol{\mu}_{\mathbf{x}} + \mathbf{C}_{\mathbf{xy}}{\mathbf{C}_{\mathbf{y}}}^{-1}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{y}}) \\ \mathbf{C}_{\mathbf{x}|\mathbf{y}} &= \mathbf{C}_{\mathbf{x}} - \mathbf{C}_{\mathbf{xy}}{\mathbf{C}_{\mathbf{y}}}^{-1}{\mathbf{C}_{\mathbf{xy}}}^{T} \\ \mathbf{C}_{\mathbf{xy}} &= \mathbf{C}_{\mathbf{x}}\mathbf{A}^{T}. \end{aligned} \tag{5.23}$$

In practice, the prior parameters $\mathbf{q} = [\boldsymbol{\mu}_{\mathbf{x}}, \mathbf{C}_{\mathbf{x}}, \mathbf{C}_{\mathbf{n}}]$ are rarely known and methods such as expectation-maximization algorithm [41], [42] are used to compute these unknown statistical parameters. However, for the sake of simplicity we treat these prior parameters as fixed and known so that there is no need to estimate them while simultaneously recovering the signal.

Our aim is to recover $\mathbf{x} \in \mathbb{R}^n$ from noisy linear measurements by finding out the posterior distribution of $\mathbf{x}$ given $\mathbf{y}$. To do this, we need to define a method for finding $\mathrm{p}(\mathbf{x}|\mathbf{y})$. Bayes' rule is the foundation of Bayesian inference. It provides a means of updating the distribution from the prior to the posterior distribution in the light of the observed data. $\mathrm{p}(\mathbf{x}|\mathbf{y})$ is the joint posterior distribution that expresses uncertainty about parameter set $\mathbf{x}$ after taking both the prior and the observation data into account.

In theory, the posterior distribution captures all information inferred from the data. This posterior is then used to make optimal decisions, or to select between models [43]. Now, let's assume that the random vector $\mathbf{x}$ is a mixture of two Gaussian vector, that is

$$\mathrm{p}(\mathbf{x}) = p_1\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{1}}, \mathbf{C}_{\mathbf{1}}) + p_2\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{2}}, \mathbf{C}_{\mathbf{2}}) \tag{5.24}$$

where $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{i}}, \mathbf{C}_{\mathbf{i}}) = \frac{1}{\sqrt{\det(2\pi\mathbf{C}_{\mathbf{i}})}}\, \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{i}})^T{\mathbf{C}_{\mathbf{i}}}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{i}})}$ for $i = \{1, 2\}$. The parameter vectors $\boldsymbol{\mu}_{\mathbf{i}}$ and $\mathbf{C}_{\mathbf{i}}$ are the mean and covariance matrix of random vector $\mathbf{x}$, respectively.

Our goal is again to derive the posterior distribution of $\mathbf{x}$ given the vector $\mathbf{y}$. Let's start by using the definition of the posterior density:

$$\begin{aligned} \mathrm{p}(\mathbf{x}|\mathbf{y}) &= \frac{\mathrm{p}(\mathbf{x}, \mathbf{y})}{\mathrm{p}(\mathbf{y})} \\ &= \frac{\mathrm{p}(\mathbf{y}|\mathbf{x})\mathrm{p}(\mathbf{x})}{\int_{\mathbf{x}} \mathrm{p}(\mathbf{x}, \mathbf{y})\, \mathrm{d}\mathbf{x}} \\ &= \frac{\mathrm{p}(\mathbf{y}|\mathbf{x})\mathrm{p}(\mathbf{x})}{\int_{\mathbf{x}} \mathrm{p}(\mathbf{y}|\mathbf{x})\mathrm{p}(\mathbf{x})\, \mathrm{d}\mathbf{x}} \\ &= \frac{\mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C}_{\mathbf{n}})\mathrm{p}(\mathbf{x})}{\int_{\mathbf{x}} \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C}_{\mathbf{n}})\mathrm{p}(\mathbf{x})\, \mathrm{d}\mathbf{x}} \\ &= \frac{\mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C}_{\mathbf{n}})[p_1\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{1}}, \mathbf{C}_{\mathbf{1}}) + p_2\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{2}}, \mathbf{C}_{\mathbf{2}})]}{\int_{\mathbf{x}} \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C}_{\mathbf{n}})\mathrm{p}(\mathbf{x})\, \mathrm{d}\mathbf{x}}. \end{aligned} \tag{5.25}$$

Using the identity for multiplication of Gaussian densities, we can express the component i, where $i = \{1, 2\}$, appearing in the numerator of $\mathrm{p}(\mathbf{x}|\mathbf{y})$ as follows:

$$\mathcal{N}(\mathbf{y}; \mathbf{Ax}, \mathbf{C_n})\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{i}, \mathbf{C_i}) = \mathcal{N}(\mathbf{y}; \mathbf{A}\boldsymbol{\mu}_\mathbf{i}, \mathbf{S_i})\mathcal{N}(\mathbf{x}; \hat{\boldsymbol{\mu}}_\mathbf{i}, \hat{\mathbf{C}}_\mathbf{i}) \tag{5.26}$$

where

$$\begin{aligned} \mathbf{S_i} &= \mathbf{AC_iA^T} + \mathbf{C_n} \\ \hat{\boldsymbol{\mu}}_\mathbf{i} &= \boldsymbol{\mu}_\mathbf{i} + \mathbf{C_iA^TS_i^{-1}}(\mathbf{y} - \mathbf{A}\boldsymbol{\mu}_\mathbf{i}) \\ \hat{\mathbf{C}}_\mathbf{i} &= \mathbf{C_i} - \mathbf{C_iA^TS_i^{-1}AC_i}. \end{aligned} \tag{5.27}$$

It should be noted that $\hat{\boldsymbol{\mu}}_\mathbf{i}$ is the mean of the posterior density for the component i, i.e., $\hat{\boldsymbol{\mu}}_\mathbf{i}$ is the estimate produced by the linear minimum mean square error (LMMSE) filter, i.e. Wiener filter, given that the observation belongs to the component i. Similarly, the matrix $\hat{\mathbf{C}}_\mathbf{i}$ is the error covariance matrix associated with the estimate $\hat{\boldsymbol{\mu}}_\mathbf{i}$.

By progressing similar to the scalar case, we can express the posterior density as

$$\mathrm{p}(\mathbf{x}|\mathbf{y}) = \hat{p}_1\mathcal{N}(\mathbf{x}; \hat{\boldsymbol{\mu}}_\mathbf{1}, \hat{\mathbf{C}}_\mathbf{1}) + \hat{p}_2\mathcal{N}(\mathbf{x}; \hat{\boldsymbol{\mu}}_\mathbf{2}, \hat{\mathbf{C}}_\mathbf{2}) \tag{5.28}$$

where

$$\hat{p}_i = \frac{p_i\mathcal{N}(\mathbf{y}; \mathbf{A}\boldsymbol{\mu}_\mathbf{i}, \mathbf{S_i})}{p_1\mathcal{N}(\mathbf{y}; \mathbf{A}\boldsymbol{\mu}_\mathbf{1}, \mathbf{S_1}) + p_2\mathcal{N}(\mathbf{y}; \mathbf{A}\boldsymbol{\mu}_\mathbf{2}, \mathbf{S_2})} \tag{5.29}$$

for $i = \{1, 2\}$.

The mean and covariance of the posterior distribution can be expressed as follows:

$$\begin{aligned} \mathrm{E}[\mathbf{x}|\mathbf{y}] &= \hat{p}_1\hat{\boldsymbol{\mu}}_\mathbf{1} + \hat{p}_2\hat{\boldsymbol{\mu}}_\mathbf{2} \\ &= \bar{\mathbf{x}} \\ \mathrm{cov}[\mathbf{x}|\mathbf{y}] &= \mathrm{E}[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T|\mathbf{y}] \\ &= \mathrm{E}[\mathbf{x}\mathbf{x}^T|\mathbf{y}] - \bar{\mathbf{x}}\bar{\mathbf{x}}^T \\ &= \mathrm{E}_\theta[\mathrm{E}_\mathbf{x}[\mathbf{x}\mathbf{x}^T|\mathbf{y}, \theta]] - \bar{\mathbf{x}}\bar{\mathbf{x}}^T \\ &= \hat{p}_1\mathrm{E}_\mathbf{x}[\mathbf{x}\mathbf{x}^T|\mathbf{y}, \theta = 1] + \hat{p}_2\mathrm{E}_\mathbf{x}[\mathbf{x}\mathbf{x}^T|\mathbf{y}, \theta = 2] - \bar{\mathbf{x}}\bar{\mathbf{x}}^T \\ &= \hat{p}_1(\hat{\mathbf{C}}_\mathbf{1} + \hat{\boldsymbol{\mu}}_\mathbf{1}\hat{\boldsymbol{\mu}}_\mathbf{1}^{\ T}) + \hat{p}_2(\hat{\mathbf{C}}_\mathbf{2}^\mathbf{2} + \hat{\boldsymbol{\mu}}_\mathbf{2}\hat{\boldsymbol{\mu}}_\mathbf{2}^{\ T}) - \bar{\mathbf{x}}\bar{\mathbf{x}}^T \\ &= \hat{p}_1\hat{\mathbf{C}}_\mathbf{1} + \hat{p}_2\hat{\mathbf{C}}_\mathbf{2} + [\hat{p}_1\hat{\boldsymbol{\mu}}_\mathbf{1}\hat{\boldsymbol{\mu}}_\mathbf{1}^{\ T} + \hat{p}_2\hat{\boldsymbol{\mu}}_\mathbf{2}\hat{\boldsymbol{\mu}}_\mathbf{2}^{\ T} - (\hat{p}_1\hat{\boldsymbol{\mu}}_\mathbf{1} + \hat{p}_2\hat{\boldsymbol{\mu}}_\mathbf{2})(\hat{p}_1\hat{\boldsymbol{\mu}}_\mathbf{1} + \hat{p}_2\hat{\boldsymbol{\mu}}_\mathbf{2})^T]. \end{aligned} \tag{5.30}$$

5.2 Formulation of the Sparse Channel Recovery for Linear Observation Models

In this section, we present an approximate Bayesian inference algorithm, expectation propagation [11], [12], [13], to estimate channel posterior density from noisy linear measurements for three special cases:

- when prior distribution of the channel taps are uncorrelated Gaussian,
- when prior distribution of the channel taps are Bernoulli-Gaussian and
- when prior distribution of the channel taps are correlated Gaussian.

5.2.1 Case 1: Estimation of Sparse Channels by Expectation Propagation Method When Prior Distribution of the Channel Taps are Uncorrelated Gaussian

Consider the following linear system model:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{5.31}$$

where $\mathbf{y} \in \mathbb{R}^m$ and random vector $\mathbf{w}$ is jointly Gaussian distributed with $\mathcal{N}(\mathbf{w}; \mathbf{0}, \mathbf{C_n})$ where $\mathbf{C_n} = \sigma_n^2\mathbf{I}$. $\mathbf{x} \in \mathbb{R}^n$ is a sparse random vector whose i^{th} entry, say x_i, is defined as a Gaussian mixture random variable. More specifically, the random variable x_i is defined as

$$\mathrm{p}(x_i) = p_L\mathcal{N}(x_i; \mu_L, \gamma_L^2) + p_H\mathcal{N}(x_i; \mu_H, \gamma_H^2). \tag{5.32}$$

Here, p_L and p_H denote the probability of selecting x_i from the Gaussian component with low and high variance, respectively. It is implicitly assumed that $\gamma_L^2 \ll \gamma_H^2$. The mixture component with the low variance denotes the channel coefficients which are not active. Similarly, the mixture component with the large variance models the active channel coefficients.

Typically, the number of active components is around $p_H \times n$ where n is the length of vector $\mathbf{x}$. The channel coefficients forming the vector $\mathbf{x}$ is assumed to be independent from each other. The prior distribution for the unknown channel vector $\mathbf{x}$ can then be written as

$$\mathrm{p}(\mathbf{x}) = \mathrm{p}(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} \mathrm{p}(x_i) = \prod_{i=1}^{n} \left(p_L\mathcal{N}(x_i; \mu_L, \gamma_L^2) + p_H\mathcal{N}(x_i; \mu_H, \gamma_H^2)\right). \tag{5.33}$$

Some possible distributions for the two dimensional case are provided in Figure 5.3.

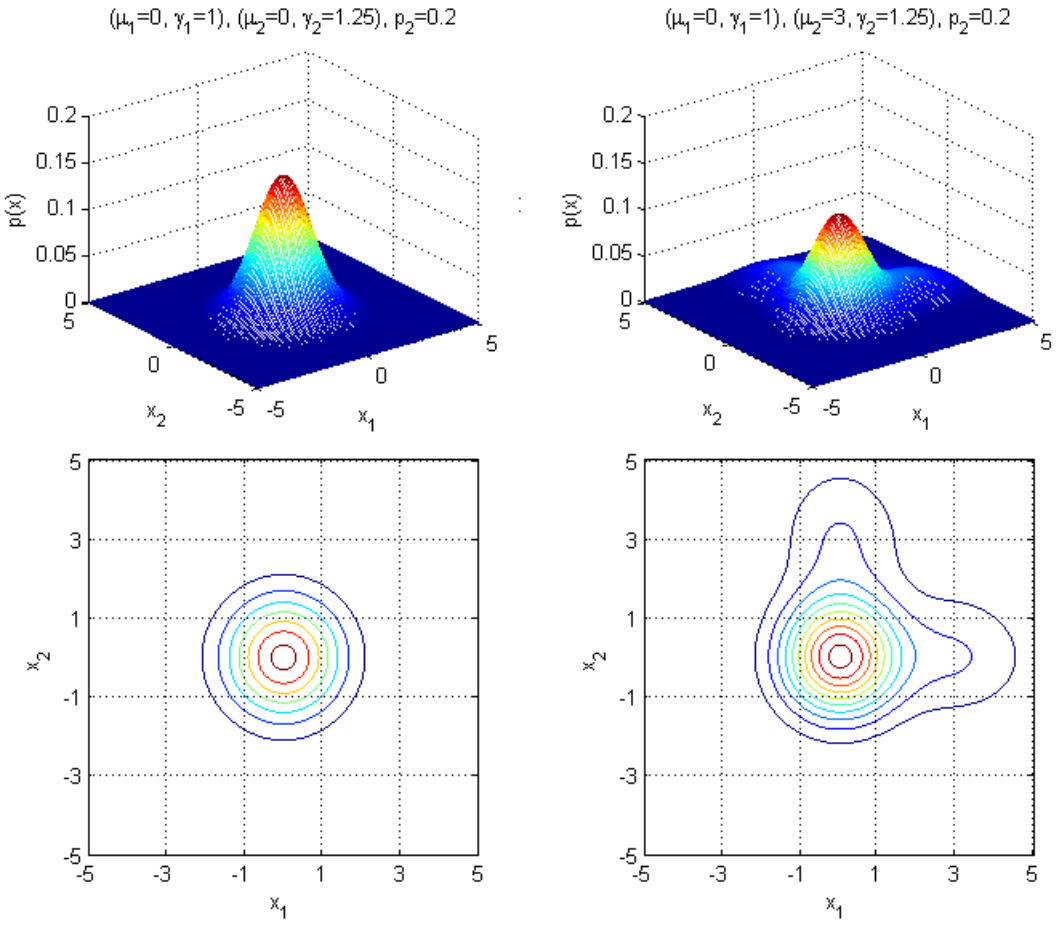

Figure 5.3: Pdf for $\mathbf{x} = [x_1\ x_2]^T$ where x_i's for $i = \{1, 2\}$ are assumed to be i.i.d. and composed of mixture of Gaussians

It should be noted that the random vector $\mathbf{x}$ is a mixture of 2^n components. It is clear that computational complexity for posterior calculation, i.e., determining the "right" mixture components, increases when n increases. Our goal is to estimate the channel coefficients of the vector $\mathbf{x}$ given the observation vector $\mathbf{y}$. We start with the posterior density calculation. The posterior density can be expressed as follows:

$$\begin{aligned}
\mathrm{p}(\mathbf{x}|\mathbf{y}) &= \frac{\mathrm{p}(\mathbf{x},\mathbf{y})}{\mathrm{p}(\mathbf{y})} \\
&= \frac{\mathrm{p}(\mathbf{y}|\mathbf{x})\mathrm{p}(\mathbf{x})}{\int_{\mathbf{x}} \mathrm{p}(\mathbf{x},\mathbf{y})\,\mathrm{d}\mathbf{x}} \\
&= \frac{\mathrm{p}(\mathbf{y}|\mathbf{x})\mathrm{p}(\mathbf{x})}{\int_{\mathbf{x}} \mathrm{p}(\mathbf{y}|\mathbf{x})\mathrm{p}(\mathbf{x})\,\mathrm{d}\mathbf{x}} \\
&= \frac{\mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})\mathrm{p}(\mathbf{x})}{\int_{\mathbf{x}} \mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})\mathrm{p}(\mathbf{x})\,\mathrm{d}\mathbf{x}} \\
&= \frac{\mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})\prod_{i=1}^{n}\left(p_L\mathcal{N}(x_i;\mu_L,\gamma_L^2)+p_H\mathcal{N}(x_i;\mu_H,\gamma_H^2)\right)}{\int_{\mathbf{x}} \mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})\mathrm{p}(\mathbf{x})\,\mathrm{d}\mathbf{x}}.
\end{aligned} \tag{5.34}$$

Since the joint density in the numerator of $\mathrm{p}(\mathbf{x}|\mathbf{y})$ is difficult to handle, we use a sequential method, called expectation propagation, to approximate this density with a computationally tractable density.

The approximation process starts with the assumption that all components of the random vector $\mathbf{x}$, that is $\{x_1, x_2, ..., x_n\}$, except the i^{th} component say x_i, is sufficiently well approximated with a Gaussian distribution with a known mean and variance. If this is the case, the joint density $\mathrm{p}(\mathbf{x}, \mathbf{y})$ is approximated as follows:

$$\mathrm{p}(\mathbf{x}, \mathbf{y}) \approx \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C_n}) \left(\prod_{k=1, k\neq i}^{n} \mathcal{N}(x_k; \mu_k, \gamma_k^2) \right) \left(p_L \mathcal{N}(x_i; \mu_L, \gamma_L^2) + p_H \mathcal{N}(x_i; \mu_H, \gamma_H^2) \right). \quad (5.35)$$

It should be noted that after this approximation, the joint distribution is a mixture of only two components and we can write the approximate distribution as

$$\mathrm{p}(\mathbf{x}, \mathbf{y}) \approx \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C_n}) \left(p_L \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{L}, \mathbf{C_L}) + p_H \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{H}, \mathbf{C_H}) \right) \quad (5.36)$$

where

$$\begin{aligned} \boldsymbol{\mu}_\mathbf{L} &= \begin{bmatrix} \mu_1 & \mu_2 & \dots & \mu_{i-1} & \mu_L & \mu_{i+1} & \dots & \mu_n \end{bmatrix}^T \\ \boldsymbol{\mu}_\mathbf{H} &= \begin{bmatrix} \mu_1 & \mu_2 & \dots & \mu_{i-1} & \mu_H & \mu_{i+1} & \dots & \mu_n \end{bmatrix}^T \\ \mathbf{C_L} &= \mathrm{diag}(\gamma_1^2, \gamma_2^2, ..., \gamma_{i-1}^2, \gamma_L^2, \gamma_{i+1}^2, ..., \gamma_n^2) \\ \mathbf{C_H} &= \mathrm{diag}(\gamma_1^2, \gamma_2^2, ..., \gamma_{i-1}^2, \gamma_H^2, \gamma_{i+1}^2, ..., \gamma_n^2). \end{aligned} \quad (5.37)$$

Substituting, the right hand side of (5.36) for $\mathrm{p}(\mathbf{x}, \mathbf{y})$ into $\mathrm{p}(\mathbf{x}|\mathbf{y}) = \frac{\mathrm{p}(\mathbf{x},\mathbf{y})}{\mathrm{p}(\mathbf{y})}$, we get

$$\mathrm{p}(\mathbf{x}|\mathbf{y}) = \frac{\mathrm{p}(\mathbf{x}, \mathbf{y})}{\mathrm{p}(\mathbf{y})} \approx \frac{\mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C_n}) \left(p_L \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{L}, \mathbf{C_L}) + p_H \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{H}, \mathbf{C_H}) \right)}{\int_\mathbf{x} \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C_n}) \left(p_L \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{L}, \mathbf{C_L}) + p_H \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{H}, \mathbf{C_H}) \right) \mathrm{d}\mathbf{x}}. \quad (5.38)$$

Using the results obtained in the previous section titled "Extension to Mixture of Gaussian Random Vectors", the posterior density of the approximate relation given in the right hand side of (5.38) can be written as

$$\mathrm{p}(\mathbf{x}|\mathbf{y}) = \hat{p}_L \mathcal{N}(\mathbf{x}; \hat{\boldsymbol{\mu}}_\mathbf{L}, \hat{\mathbf{C}}_\mathbf{L}) + \hat{p}_H \mathcal{N}(\mathbf{x}; \hat{\boldsymbol{\mu}}_\mathbf{H}, \hat{\mathbf{C}}_\mathbf{H}) \quad (5.39)$$

where

$$\begin{aligned} \mathbf{S_i} &= \mathbf{A}\mathbf{C_i}\mathbf{A}^\mathbf{T} + \mathbf{C_n} \\ \hat{\boldsymbol{\mu}}_\mathbf{i} &= \boldsymbol{\mu}_\mathbf{i} + \mathbf{C_i}\mathbf{A}^\mathbf{T}\mathbf{S_i^{-1}}(\mathbf{y} - \mathbf{A}\boldsymbol{\mu}_\mathbf{i}) \\ \hat{\mathbf{C}}_\mathbf{i} &= \mathbf{C_i} - \mathbf{C_i}\mathbf{A}^\mathbf{T}\mathbf{S_i^{-1}}\mathbf{A}\mathbf{C_i} \end{aligned} \quad (5.40)$$

and

$$\hat{p}_i = \frac{p_i \mathcal{N}(\mathbf{y}; \mathbf{A}\boldsymbol{\mu}_\mathbf{i}, \mathbf{S_i})}{p_L \mathcal{N}(\mathbf{y}; \mathbf{A}\boldsymbol{\mu}_\mathbf{L}, \mathbf{S_L}) + p_H \mathcal{N}(\mathbf{y}; \mathbf{A}\boldsymbol{\mu}_\mathbf{H}, \mathbf{S_H})} \quad (5.41)$$

for $i = \{L, H\}$. Once the posterior density is approximated, the density of i^{th} component of $\mathbf{x}$, x_i, is updated using this approximation. To update the density of x_i, the joint density of $x_1, x_2, ..., x_n$ given the observation $\mathbf{y}$, is marginalized and $\mathrm{p}(x_i|\mathbf{y})$ is retrieved. It can be easily seen that

$$\mathrm{p}(x_i|\mathbf{y}) = \hat{p}_L \mathcal{N}(x_i; \hat{\mu}_{L,i}, \hat{\gamma}^2_{L,i}) + \hat{p}_H \mathcal{N}(x_i; \hat{\mu}_{H,i}, \hat{\gamma}^2_{H,i}). \tag{5.42}$$

In the final relation, $\hat{\mu}_{L,i}$ is the i^{th} entry of the vector $\hat{\boldsymbol{\mu}}_{\boldsymbol{L}}$ and $\hat{\gamma}^2_{L,i}$ is the (i,i) element of matrix $\hat{\mathbf{C}}_{\mathbf{L}}$. Similar definitions apply for $\hat{\mu}_{H,i}$ and $\hat{\gamma}^2_{H,i}$. The mean and variance of $\mathrm{p}(x_i|\mathbf{y})$ can be written as

$$\begin{aligned} \mathrm{E}[x_i|\mathbf{y}] &= \hat{p}_L \hat{\mu}_{L,i} + \hat{p}_H \hat{\mu}_{H,i} \\ \mathrm{var}[x_i|\mathbf{y}] &= \hat{p}_L \hat{\gamma}^2_{L,i} + \hat{p}_H \hat{\gamma}^2_{H,i} + [\hat{p}_L \hat{\mu}^2_{L,i} + \hat{p}_H \hat{\mu}^2_{H,i} - (\hat{p}_L \hat{\mu}_{L,i} + \hat{p}_H \hat{\mu}_{H,i})^2]. \end{aligned} \tag{5.43}$$

The final step of expectation propagation is to approximate the updated density of x_i given the observation $\mathbf{y}$ with the Gaussian distribution with the mean $\mathrm{E}[x_i|\mathbf{y}]$ and variance $\mathrm{var}[x_i|\mathbf{y}]$. Once, the posterior distribution of x_i is updated, the expectation propagation process continues with the update of the next random variable, say, x_{i+1}.

The process terminates once the conducted updates do not result in a significant change, i.e., until the convergence to the fixed point of the iterations.

In the expectation propagation method, the computationally most complex operation is the computation of the inverse of the matrix $\boldsymbol{S_i} = \boldsymbol{A}\boldsymbol{C_i}\boldsymbol{A}^T + \boldsymbol{C_n}$ for $i = \{L, H\}$. It is possible to reduce the computation complexity with the application of matrix inversion lemma which is provided in Appendix A.

Clairvoyant Estimator

If we have knowledge about the type of the channel coefficients, that is whether each channel coefficient is generated from the low variance or high variance Gaussian distribution, then we would have a better estimate than the approximate Bayesian inference algorithm. Let's assume that we have the clairvoyant knowledge about the type of the channel coefficients such that

$$\mathbf{x} \sim \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}}, \mathbf{C}_{\mathbf{x}}) \tag{5.44}$$

where

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{x}} &= \begin{bmatrix} \mu_1 & \mu_2 & \dots & \mu_n \end{bmatrix}^T \\ \mathbf{C}_{\mathbf{x}} &= \mathrm{diag}(\gamma_1^2, \gamma_2^2, \dots, \gamma_n^2) \end{aligned} \tag{5.45}$$

and

$$\mu_i = \begin{cases} \mu_L & \text{if i}^{\text{th}} \text{ tap is L} \\ \mu_H & \text{if i}^{\text{th}} \text{ tap is H} \end{cases} \tag{5.46}$$

$$\gamma_i^2 = \begin{cases} \gamma_L^2 & \text{if i}^{\text{th}} \text{ tap is L} \\ \gamma_H^2 & \text{if i}^{\text{th}} \text{ tap is H} \end{cases} \tag{5.47}$$

for $i = \{1, 2, \ldots, n\}$. Then, the joint density can be written as

$$\mathrm{p}_{\text{clairvoyant}}(\mathbf{x}, \mathbf{y}) = \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C_n})\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{x}, \mathbf{C_x}). \tag{5.48}$$

By using the results obtained in the previous section titled "Extension to Mixture of Gaussian Random Vectors", the posterior density can be rewritten as

$$\mathrm{p}_{\text{clairvoyant}}(\mathbf{x}|\mathbf{y}) = \frac{\mathrm{p}(\mathbf{x}, \mathbf{y})}{\mathrm{p}(\mathbf{y})} = \mathcal{N}(\mathbf{x}; \hat{\boldsymbol{\mu}}_\mathbf{x}, \hat{\mathbf{C}}_\mathbf{x}) \tag{5.49}$$

where

$$\begin{aligned} \mathbf{S} &= \mathbf{A}\mathbf{C}_\mathbf{x}\mathbf{A}^\mathbf{T} + \mathbf{C_n} \\ \hat{\boldsymbol{\mu}}_\mathbf{x} &= \boldsymbol{\mu}_\mathbf{x} + \mathbf{C}_\mathbf{x}\mathbf{A}^\mathbf{T}\mathbf{S}^{-\mathbf{1}}(\mathbf{y} - \mathbf{A}\boldsymbol{\mu}_\mathbf{x}) \\ \hat{\mathbf{C}}_\mathbf{x} &= \mathbf{C}_\mathbf{x} - \mathbf{C}_\mathbf{x}\mathbf{A}^\mathbf{T}\mathbf{S}^{-\mathbf{1}}\mathbf{A}\mathbf{C}_\mathbf{x}. \end{aligned} \tag{5.50}$$

Clairvoyant knowledge on the channel taps does not always exist in practice, however clairvoyant estimator serves a lower bound for RMSE.

5.2.1.1 Performance Comparison of Sparse Channel Estimation Methods

In this section, we first provide the simulation result which presents a general working principle of the expectation propagation method and then provide a performance comparison of the sparse channel estimation techniques studied in this thesis.

We have implemented the expectation propagation algorithm by taking $\mathbf{A}$ to be $m \times n$ Gaussian matrix and SNR at a fixed value. A sample output of the algorithm is illustrated in Figure 5.4. Stem-function is used to illustrate the true and estimated coefficient of each channel tap. True values for the coefficients are shown with crosses and circles show the estimates at a given cycle. The length of x-axis is equal to the channel length n and y-axis shows the true and estimated coefficients of the channel taps. Simulation result shows us that under the given a-priori parameters, channel is recovered at the end of approximately 50 cycles with RMSE ≈ 0.94.

We should choose smaller p_H values in order to have sparser channels. Figure 5.5 is the evidence of this deduction. In Figure 5.4 p_H is set to 0.5, whereas it is set to 0.2 in Figure 5.5 and as a result of this, sparsity level of the estimated channel in Figure 5.5 is lower than the previous figure.

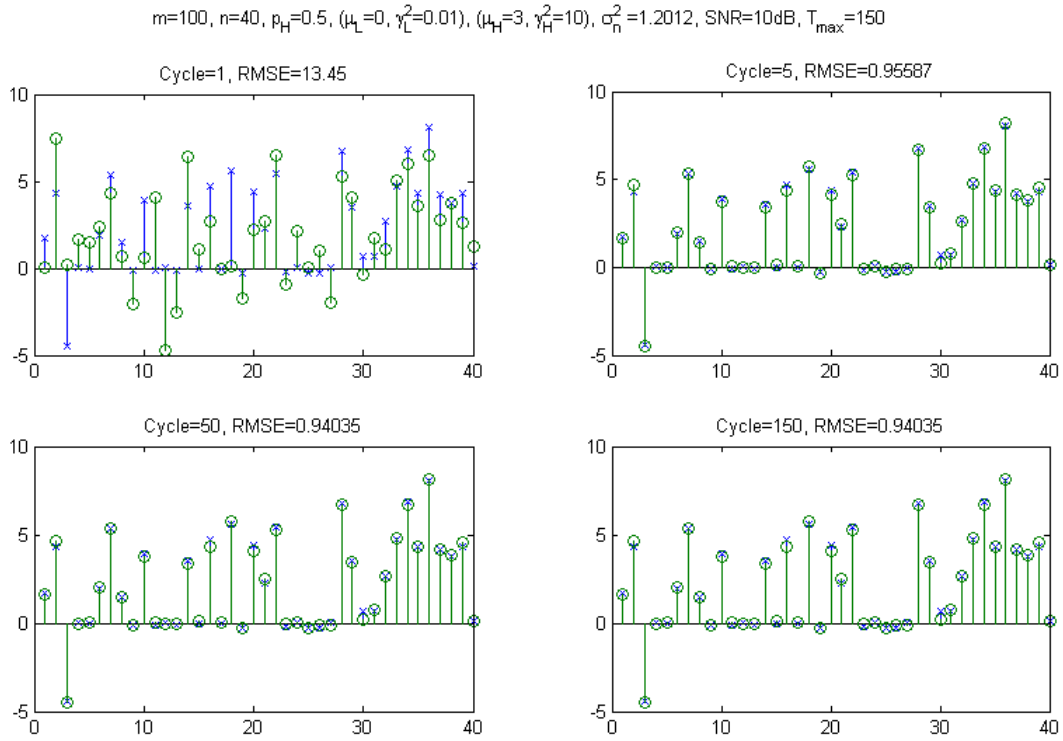

Figure 5.4: Performance analysis of approximate Bayesian inference algorithm

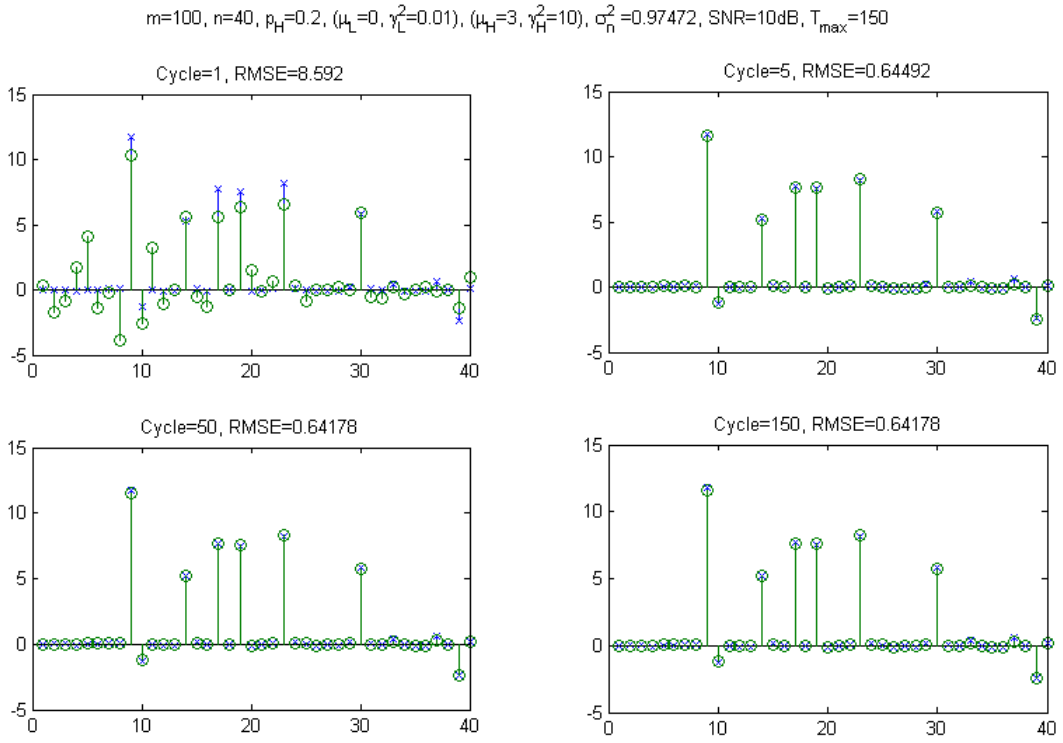

Figure 5.5: Performance analysis of approximate Bayesian inference algorithm

In Figure 5.6, measurement matrix $\mathbf{A}$ is drawn to be an $m \times n$ Toeplitz structured matrix which uses length-7 MPS code as the training sequence. $\mathbf{x} \in \mathbb{R}^n$ is a sparse channel where channel taps are drawn from a mixture of a narrow Gaussian $\mathcal{N}(\mu_L, \gamma_L^2)$ and a broad Gaussian $\mathcal{N}(\mu_H, \gamma_H^2)$. Stopping criterion in OMP algorithm is set to channel sparsity level k where k is calculated by considering the estimated channel taps satisfying condition $|\hat{x}_i| > 0.1$. Making a good guess for the parameter λ in the LASSO method is difficult due to the channel distribution. Because of this reason, for each iteration, λ is chosen such that its value is consistent with the l_1 norm of the sparse channel. Measurements are corrupted by additive white Gaussian noise $\mathbf{w}$ with $\mathcal{N}(\mathbf{0}, \sigma_n^2\mathbf{I})$.

Parameter settings are as follows: $n = 50$, $l_t = 7$, $m = 56$, $p_H = 0.2$, $(\mu_L = 0, \gamma_L^2 = 0.01)$, and $(\mu_H = 3, \gamma_H^2 = 4)$. As mentioned before, p_H should be chosen small enough to construct a sparse channel. In according to our setting where $p_H = 0.2$, nearly 80% of the channel taps are chosen from the narrow Gaussian $\mathcal{N}(0, 0.01)$ assuring sparsity. For each curve, 1000 independent Monte Carlo trials have been conducted.

Figure 5.6 presents the channel estimation error that uses RMSE standard for the LS, MMSE, LASSO, OMP, expectation propagation method and clairvoyant estimator. As mentioned before, note that performances of the OMP and LASSO methods depend on the selection of the stopping criteria used for halting the iterations and parameter λ, respectively. From the figure, we can conclude that performance of the expectation propagation method is better than the classical estimation techniques. This is due to fact that it takes advantage of the a-priori knowledge of the channel distribution.

In the same figure we also provide the estimation performance when clairvoyant knowledge of the channel coefficients is available. The clairvoyant covariance matrices contain the true second-order statistical information so that if one knows the locations of the active taps of the channel, then this would be the best estimator that can be realized. Clairvoyant knowledge on the channel taps does not always exist in practice, however this algorithm presents a performance bound for the other algorithms.

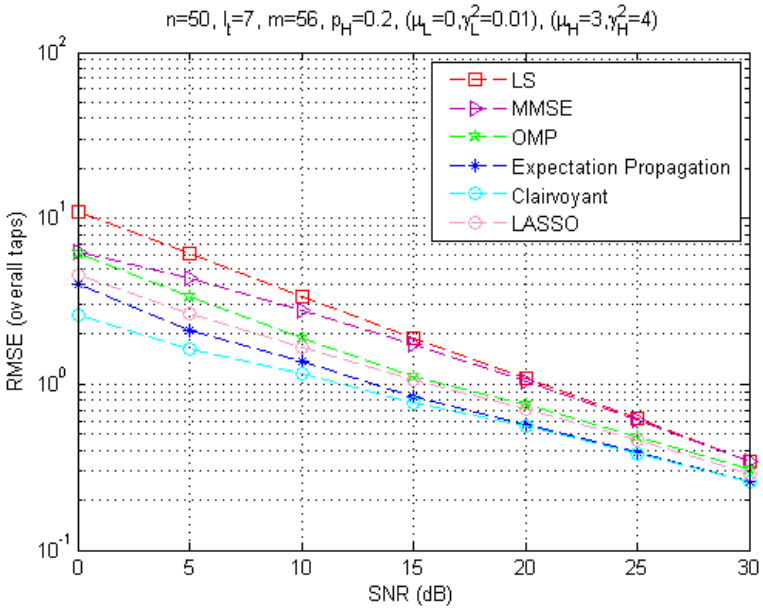

Figure 5.6: Performance comparison of different channel estimation methods at different SNR's

5.2.2 Case 2: Estimation of Sparse Channels by Expectation Propagation Method When Prior Distribution of the Channel Taps are Bernoulli-Gaussian

Let's assume that the prior distribution of channel tap x is composed of a mixture of Dirac delta and Gaussian, then $\mathrm{p}(x)$ can then be written as follows:

$$\mathrm{p}(x) = p_1\delta(x) + p_2\mathcal{N}(x;\mu,\gamma^2) \tag{5.51}$$

where $\mathcal{N}(x;\mu,\gamma^2) = \frac{1}{\sqrt{2\pi\gamma^2}}\,\mathrm{e}^{-\frac{(x-\mu)^2}{2\gamma^2}}$ is the univariate Gaussian distribution with the component probability p_2 where $p_1 + p_2 = 1$.

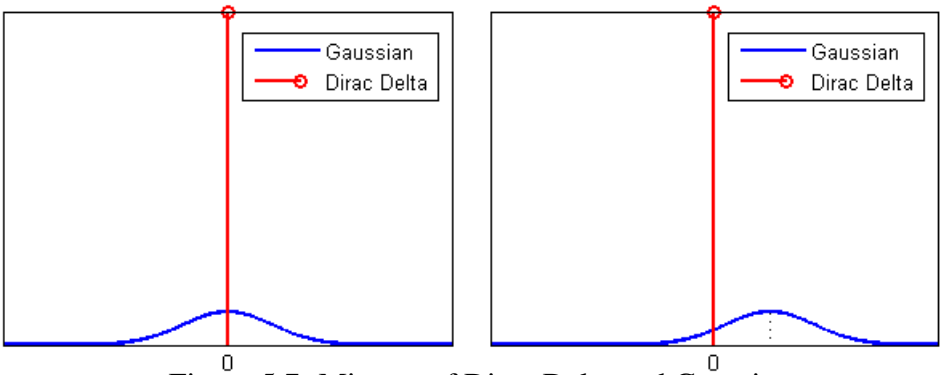

Figure 5.7: Mixture of Dirac Delta and Gaussian

Parameter p_2 can be thought as the sparsity level i.e., the ratio of k/n where k is the number of non-zero coefficients in x. Note that this is the specific case of (5.1) where $\gamma_1^2 \to 0$ such that the narrow distribution is taken as a Dirac delta and the broader one

is taken as $\mathcal{N}(x;\mu,\gamma^2)$. Some possible distributions for the one dimensional case are provided in Figure 5.8.

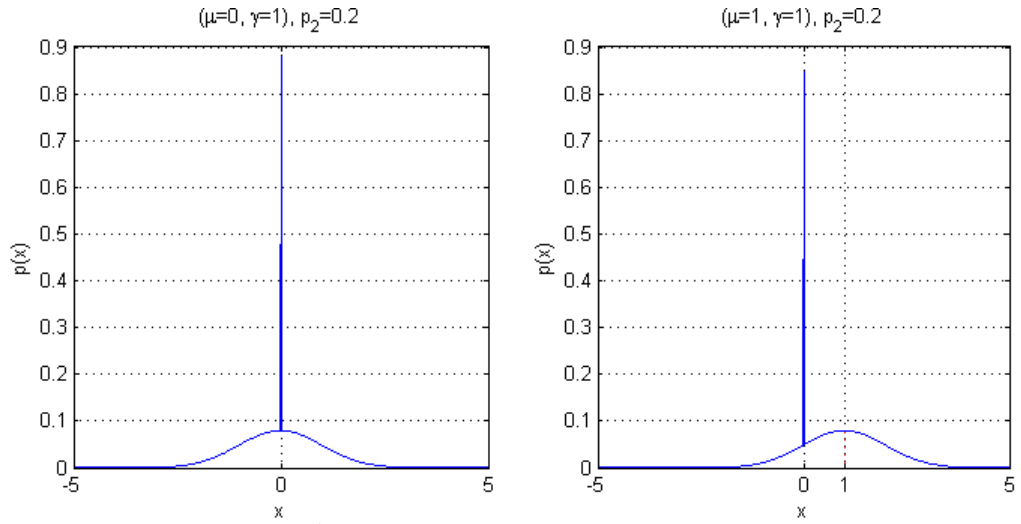

Figure 5.8: Pdf for $x \in \mathbb{R}^1$ where x is composed of Dirac Delta and Gaussian distribution

Now, let's define approximate Bayesian inference algorithm to recover the Bernoulli-Gaussian modelled sparse channels. The general scenario has similar steps as in the case of mixture of Gaussians. Consider the following linear system model:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{5.52}$$

where $\mathbf{y} \in \mathbb{R}^m$, random vector $\mathbf{w}$ is jointly Gaussian distributed with $\mathcal{N}(\mathbf{w};\mathbf{0},\mathbf{C_n})$ where $\mathbf{C_n} = \sigma_n^2\mathbf{I}$ and $\mathbf{x} \in \mathbb{R}^n$ is a random vector whose i^{th} entry, say x_i, is composed of a mixture of Dirac delta and Gaussian. More specifically, prior distribution of x_i is defined as

$$\mathrm{p}(x_i) = p_L\delta(x_i) + p_H\mathcal{N}(x_i;\mu_H,\gamma_H^2). \tag{5.53}$$

Here, p_L and p_H denote the probability of selecting x_i from the Bernoulli and the Gaussian component, respectively. The component with the Bernoulli distribution denotes the channel coefficients which are not active. Similarly, the component with the Gaussian distribution models the active channel coefficients. Typically, the number of active components is around $p_H \times n$ where n is the length of vector $\mathbf{x}$.

The channel coefficients forming the vector $\mathbf{x}$ is assumed to be independent from each other. The prior distribution for the unknown channel vector $\mathbf{x}$ can then be written as

$$\mathrm{p}(\mathbf{x}) = \mathrm{p}(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} \mathrm{p}(x_i) = \prod_{i=1}^{n} \left(p_L\delta(x_i) + p_H\mathcal{N}(x_i;\mu_H,\gamma_H^2)\right). \tag{5.54}$$

Some possible distributions for the two dimensional case are provided in Figure 5.9.

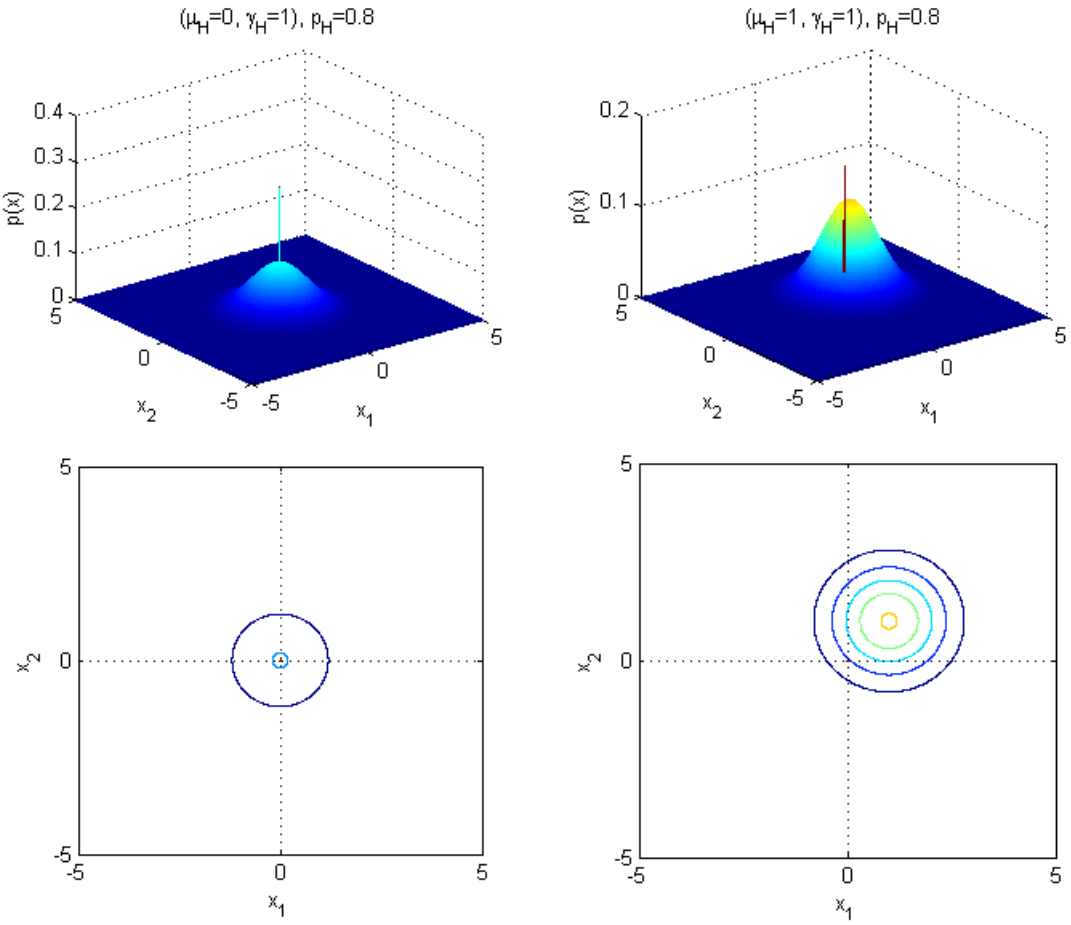

Figure 5.9: Pdf for $\mathbf{x} = [x_1\ x_2]^T$ where x_i's for $i = \{1, 2\}$ are assumed to be i.i.d. and composed of Dirac Delta and Gaussian distribution

Our goal is to estimate the channel coefficients of the vector $\mathbf{x}$ given the observation vector $\mathbf{y}$. We start with the posterior density calculation. The posterior density can be expressed as follows:

$$\begin{aligned}
\mathrm{p}(\mathbf{x}|\mathbf{y}) &= \frac{\mathrm{p}(\mathbf{x},\mathbf{y})}{\mathrm{p}(\mathbf{y})} \\
&= \frac{\mathrm{p}(\mathbf{y}|\mathbf{x})\mathrm{p}(\mathbf{x})}{\int_{\mathbf{x}} \mathrm{p}(\mathbf{x},\mathbf{y})\,\mathrm{d}\mathbf{x}} \\
&= \frac{\mathrm{p}(\mathbf{y}|\mathbf{x})\mathrm{p}(\mathbf{x})}{\int_{\mathbf{x}} \mathrm{p}(\mathbf{y}|\mathbf{x})\mathrm{p}(\mathbf{x})\,\mathrm{d}\mathbf{x}} \\
&= \frac{\mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})\mathrm{p}(\mathbf{x})}{\int_{\mathbf{x}} \mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})\mathrm{p}(\mathbf{x})\,\mathrm{d}\mathbf{x}} \\
&= \frac{\mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})\prod_{i=1}^{n}\left(p_L\delta(x_i) + p_H\mathcal{N}(x_i;\mu_H,\gamma_H^2)\right)}{\int_{\mathbf{x}} \mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})\mathrm{p}(\mathbf{x})\,\mathrm{d}\mathbf{x}}.
\end{aligned} \tag{5.55}$$

Since the joint density in the numerator of $\mathrm{p}(\mathbf{x}|\mathbf{y})$ is difficult to handle, we use ex-

pectation propagation method to approximate this density. The approximation process starts with the assumption that all components of the random vector $\mathbf{x}$, that is $\{x_1, x_2, ..., x_n\}$, except the i^{th} component say x_i, is sufficiently well approximated with a Gaussian distribution with a given mean and variance. If this is the case, the joint density $\mathrm{p}(\mathbf{x}, \mathbf{y})$ can be approximated as follows:

$$\begin{aligned}
\mathrm{p}(\mathbf{x}, \mathbf{y}) &\approx \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C_n}) \left(\prod_{k=1, k\neq i}^{n} \mathcal{N}(x_k; \mu_k, \gamma_k^2) \right) \left(p_L \delta(x_i) + p_H \mathcal{N}(x_i; \mu_H, \gamma_H^2) \right) \\
&= \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C_n}) \left(\prod_{k=1, k\neq i}^{n} \mathcal{N}(x_k; \mu_k, \gamma_k^2) \right) p_L \delta(x_i) \\
&\qquad + \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C_n}) p_H \mathcal{N}(\mathbf{x}; \boldsymbol{\mu_H}, \mathbf{C_H}) \\
&= \mathcal{N}(\mathbf{y}; \mathbf{A}_{\mathbf{s_L}}\mathbf{x}_{\mathbf{s_L}}, \mathbf{C_n}) \mathcal{N}(\mathbf{x}_{\mathbf{s_L}}; \boldsymbol{\mu_L}, \mathbf{C_L}) p_L \delta(x_i) \\
&\qquad + \mathcal{N}(\mathbf{y}; \mathbf{A}_{\mathbf{s_H}}\mathbf{x}_{\mathbf{s_H}}, \mathbf{C_n}) p_H \mathcal{N}(\mathbf{x}_{\mathbf{s_H}}; \boldsymbol{\mu_H}, \mathbf{C_H})
\end{aligned} \tag{5.56}$$

where

$$\begin{aligned}
\mathbf{A}_{\mathbf{s_L}} &= \begin{bmatrix} \mathbf{a_1} & \mathbf{a_2} & \dots & \mathbf{a_{i-1}} & \mathbf{a_{i+1}} & \dots & \mathbf{a_n} \end{bmatrix} \\
\mathbf{A}_{\mathbf{s_H}} = \mathbf{A} &= \begin{bmatrix} \mathbf{a_1} & \mathbf{a_2} & \dots & \mathbf{a_{i-1}} & \mathbf{a_i} & \mathbf{a_{i+1}} & \dots & \mathbf{a_n} \end{bmatrix} \\
\mathbf{x}_{\mathbf{s_L}} &= \begin{bmatrix} x_1 & x_2 & \dots & x_{i-1} & x_{i+1} & \dots & x_n \end{bmatrix}^T \\
\mathbf{x}_{\mathbf{s_H}} = \mathbf{x} &= \begin{bmatrix} x_1 & x_2 & \dots & x_{i-1} & x_i & x_{i+1} & \dots & x_n \end{bmatrix}^T \\
\boldsymbol{\mu_L} &= \begin{bmatrix} \mu_1 & \mu_2 & \dots & \mu_{i-1} & \mu_{i+1} & \dots & \mu_n \end{bmatrix}^T \\
\boldsymbol{\mu_H} &= \begin{bmatrix} \mu_1 & \mu_2 & \dots & \mu_{i-1} & \mu_H & \mu_{i+1} & \dots & \mu_n \end{bmatrix}^T \\
\mathbf{C_L} &= \mathrm{diag}(\gamma_1^2, \gamma_2^2, ..., \gamma_{i-1}^2, \gamma_{i+1}^2, ..., \gamma_n^2) \\
\mathbf{C_H} &= \mathrm{diag}(\gamma_1^2, \gamma_2^2, ..., \gamma_{i-1}^2, \gamma_H^2, \gamma_{i+1}^2, ..., \gamma_n^2).
\end{aligned} \tag{5.57}$$

The vector $\mathbf{a_i}$ is the i^{th} column of $\mathbf{A}$ that is used to generate $\mathbf{y}$. Substituting, (5.56) for $\mathrm{p}(\mathbf{x}, \mathbf{y})$ into $\mathrm{p}(\mathbf{x}|\mathbf{y}) = \frac{\mathrm{p}(\mathbf{x},\mathbf{y})}{\mathrm{p}(\mathbf{y})}$, we get

$$\begin{aligned}
\mathrm{p}(\mathbf{x}|\mathbf{y}) &= \frac{\mathrm{p}(\mathbf{x}, \mathbf{y})}{\mathrm{p}(\mathbf{y})} \\
&\approx \frac{1}{K} \mathcal{N}(\mathbf{y}; \mathbf{A}_{\mathbf{s_L}}\mathbf{x}_{\mathbf{s_L}}, \mathbf{C_n}) \mathcal{N}(\mathbf{x}_{\mathbf{s_L}}; \boldsymbol{\mu_L}, \mathbf{C_L}) p_L \delta(x_i) \\
&\qquad + \frac{1}{K} \mathcal{N}(\mathbf{y}; \mathbf{A}_{\mathbf{s_H}}\mathbf{x}_{\mathbf{s_H}}, \mathbf{C_n}) p_H \mathcal{N}(\mathbf{x}_{\mathbf{s_H}}; \boldsymbol{\mu_H}, \mathbf{C_H}).
\end{aligned} \tag{5.58}$$

Here K is a normalizing constant which is equal to the integral of the numerator over $\mathbf{x}$. Using the results obtained in the previous section titled "Extension to Mixture of Gaussian Random Vectors", the posterior density of the approximate relation given in the right hand side of (5.58) can be written as

$$\mathrm{p}(\mathbf{x}|\mathbf{y}) = \hat{p}_L \mathcal{N}(\mathbf{x}_{\mathbf{s_L}}; \hat{\boldsymbol{\mu}}_{\mathbf{L}}, \hat{\mathbf{C}}_{\mathbf{L}}) \delta(x_i) + \hat{p}_H \mathcal{N}(\mathbf{x}_{\mathbf{s_H}}; \hat{\boldsymbol{\mu}}_{\mathbf{H}}, \hat{\mathbf{C}}_{\mathbf{H}}) \tag{5.59}$$

where

$$\begin{aligned}\mathbf{S_i} &= \mathbf{A_{s_i}} \mathbf{C_i} \mathbf{A_{s_i}^T} + \mathbf{C_n} \\ \hat{\boldsymbol{\mu}}_\mathbf{i} &= \boldsymbol{\mu}_\mathbf{i} + \mathbf{C_i} \mathbf{A_{s_i}^T} \mathbf{S_i^{-1}} (\mathbf{y} - \mathbf{A_{s_i}} \boldsymbol{\mu}_\mathbf{i}) \\ \hat{\mathbf{C}}_\mathbf{i} &= \mathbf{C_i} - \mathbf{C_i} \mathbf{A_{s_i}^T} \mathbf{S_i^{-1}} \mathbf{A_{s_i}} \mathbf{C_i}\end{aligned} \tag{5.60}$$

and

$$\hat{p}_i = \frac{p_i \mathcal{N}(\mathbf{y}; \mathbf{A_{s_i}} \boldsymbol{\mu}_\mathbf{i}, \mathbf{S_i})}{p_L \mathcal{N}(\mathbf{y}; \mathbf{A_{s_L}} \boldsymbol{\mu}_\mathbf{L}, \mathbf{S_L}) + p_H \mathcal{N}(\mathbf{y}; \mathbf{A_{s_H}} \boldsymbol{\mu}_\mathbf{H}, \mathbf{S_H})} \tag{5.61}$$

for $i = \{L, H\}$. Once the posterior density is approximated, the density of i^{th} component of $\mathbf{x}$, x_i, is updated using this approximation. To update the density of x_i, the joint density of $x_1, x_2, ..., x_n$ given the observation $\mathbf{y}$, is marginalized and $\mathrm{p}(x_i|\mathbf{y})$ is retrieved. It can be easily seen that

$$\mathrm{p}(x_i|\mathbf{y}) = \hat{p}_L \delta(x_i) + \hat{p}_H \mathcal{N}(x_i; \hat{\mu}_{H,i}, \hat{\gamma}^2_{H,i}). \tag{5.62}$$

In the final relation, $\hat{\mu}_{H,i}$ is the i^{th} entry of the vector $\hat{\boldsymbol{\mu}}_{\boldsymbol{H}}$ and $\hat{\gamma}^2_{H,i}$ is the (i,i) element of matrix $\hat{\mathbf{C}}_\mathbf{H}$. The mean and variance of $\mathrm{p}(x_i|\mathbf{y})$ can be written as

$$\begin{aligned}\mathrm{E}[x_i|\mathbf{y}] &= \hat{p}_H \hat{\mu}_{H,i} \\ \mathrm{var}[x_i|\mathbf{y}] &= (\hat{p}_H - \hat{p}_H^2) \hat{\mu}^2_{H,i} + \hat{p}_H \hat{\gamma}^2_{H,i}.\end{aligned} \tag{5.63}$$

The final step of expectation propagation is to approximate the updated density of x_i given the observation $\mathbf{y}$ with the mean $\mathrm{E}[x_i|\mathbf{y}]$ and variance $\mathrm{var}[x_i|\mathbf{y}]$. Once, the posterior distribution of x_i is updated, the expectation propagation process continues with the update of the next random variable, say, x_{i+1}.

The process terminates once the conducted updates do not result in a significant change, i.e., until the convergence to the fixed point of the iterations.

5.2.2.1 Performance Comparison of Sparse Channel Estimation Methods

In this section, we provide the simulation result which presents the performance comparison of the LS, MMSE, LASSO, OMP, expectation propagation and clairvoyant methods which are used for sparse channel estimation purposes.

In the simulation, measurement matrix $\mathbf{A}$ is drawn to be an $m \times n$ Toeplitz structured matrix which uses length-7 MPS code as the training sequence. $\mathbf{x} \in \mathbb{R}^n$ is a sparse channel where channel taps are drawn from a mixture of a Dirac delta and a Gaussian $\mathcal{N}(\mu_H, \gamma_H^2)$. Stopping criterion in OMP algorithm is set to channel sparsity level k where k is the total number of estimated channel taps satisfying condition $|\hat{x_i}| > 0$. Making a good guess for the parameter λ in the LASSO method is difficult due to the channel distribution. Because of this reason, for each iteration, λ is chosen such

that its value is consistent with the l_1 norm of the sparse channel. Measurements are corrupted by additive white Gaussian noise $\mathbf{w}$ with $\mathcal{N}(\mathbf{0}, \sigma_n^2\mathbf{I})$.

Parameter settings are as follows: $n = 50$, $l_t = 7$, $m = 56$, $p_H = 0.2$, and $(\mu_H = 3, \gamma_H^2 = 4)$. As mentioned before, p_H should be chosen small enough to construct a sparse channel. When $p_H = 0.2$, nearly 80% of the channel taps are chosen from the Dirac delta assuring sparsity. For each curve, 1000 independent Monte Carlo trials have been conducted.

Figure 5.10 presents the channel estimation error (RMSE) for the LS, MMSE, LASSO, OMP, expectation propagation and clairvoyant methods. As mentioned before, note that performances of the OMP and LASSO methods depend on the selection of the stopping criteria used for halting the iterations and parameter λ, respectively. From this figure, we can conclude that the performance of the expectation propagation algorithm is better than the classical estimation techniques. This is due to fact that it takes advantage of the a-priori knowledge of the channel distribution.

In the same figure we also provide estimation performance when clairvoyant knowledge of the channel covariance is available. The clairvoyant covariance matrices contain the true second-order statistical information so that if one knows the locations of the active taps of the channel, then this would be the best estimator that could ever be realized. Clairvoyant knowledge on the channel taps does not always exist in practice, however this algorithm presents a performance bound for the other algorithms. As seen from the figure expectation propagation algorithm results in an estimation performance that is close to the clairvoyant case.

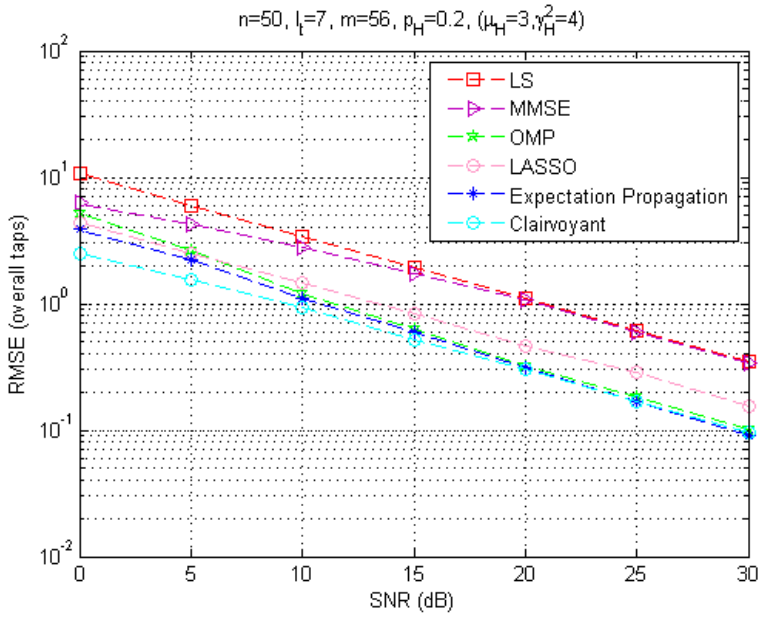

Figure 5.10: Performance comparison of different channel estimation methods at different SNR's

5.2.3 Case 3: Estimation of Sparse Channels by Expectation Propagation Method When Prior Distribution of the Channel Taps are Correlated Gaussian

We consider the following Markov chain:

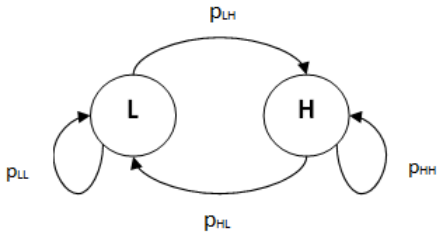

Figure 5.11: Markov chain

The chain contains two states shown with L and H. The transition probabilities between the states are given as p_{XY}, where p_{XY} is the probability of the transition from state X to state Y. Then, the transition matrix can be written as:

$$\mathbf{T} = \begin{bmatrix} p_{LL} & p_{LH} \\ p_{HL} & p_{HH} \end{bmatrix}. \tag{5.64}$$

The steady-state probabilities of the states, that is π_L and π_H, can be easily found through the solution of the following equations:

$$\begin{aligned} \pi_L p_{LH} &= \pi_H p_{HL} \\ \pi_L + \pi_H &= 1. \end{aligned} \tag{5.65}$$

The solution can be written as $\begin{bmatrix} \pi_L \\ \pi_H \end{bmatrix} = \frac{1}{p_{HL}+p_{LH}} \begin{bmatrix} p_{HL} \\ p_{LH} \end{bmatrix}$. For example, the choice of $p_{LH} = 0.1, p_{HL} = 0.4$ results in $\pi_L = 0.8$ and $\pi_H = 0.2$. Hence, if the states L and H represents the channel state with low valued channel taps (non-active taps) and high valued taps (active taps) respectively, then the number of non-active taps is expected to be 4 times than the number of active taps. It can be seen that with a suitable choice of transition probabilities any level of sparsity, in sense of active channel taps, can be achieved.

If the channel state is H at time n, the probability of switching to the L state for the first time at the time step $n + k$, i.e staying in H until $n + k - 1$ and then switching to L at the $n + k$ step, is $(p_{HH})^{k-1} p_{HL}$. From this distribution, the expected number of consecutive H states can be calculated as $E\{\text{number of consecutive } H \text{ states}\} = \frac{1}{p_{HL}}$.

For $p_{LH} = 0.1, p_{HL} = 0.4$, the expected number of consecutive number of H states, i.e., the number of steps that the channel remains in H state continuously, is 2.5. For the same set of parameters, $E\{\text{number of consecutive } L \text{ states}\} = \frac{1}{p_{LH}} = 10$.

A typical run of the Markov chain for the $p_{LH} = 0.1$, $p_{HL} = 0.4$ can be given as

$$[0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,0\,0].$$

Here 0 and 1 represents states L and H respectively.

The given channel model is aimed to represent the channels where a long string of non-active taps are followed by a string of active taps. Such channels may arise in the environments where the scatters are scarce in number such as underwater channels or ultra wide band channels.

Our goal in this section is to present a method to estimate channels whose active and non-active components obey a Markov chain. We focus on Gaussian channels where i^{th} channel tap, x_i for $i = \{1, 2, \ldots, n\}$, is distributed according the random variable θ_i indicating the state of the Markov chain such that

$$x_i \sim \begin{cases} \mathcal{N}(\mu_L, \gamma_L^2) & \theta_i = L \\ \mathcal{N}(\mu_H, \gamma_H^2) & \theta_i = H \end{cases}. \tag{5.66}$$

The joint distribution of $\mathbf{x}$ and $\boldsymbol{\theta}$ can be written as follows:

$$\begin{aligned} \mathrm{p}(\mathbf{x}, \boldsymbol{\theta}) &= \mathrm{p}(\boldsymbol{x}|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}) \\ &= \left(\prod_{i=1}^{n} \mathrm{p}(x_i|\theta_i)\right)\left(\mathrm{p}(\theta_0)\prod_{i=1}^{n}\mathrm{p}(\theta_i|\theta_{i-1})\right) \\ &= \left(\prod_{i=1}^{n} \mathcal{N}(x_i; \mu_{\theta_i}, \gamma_{\theta_i}^2)\right)\left(\mathrm{p}(\theta_0)\prod_{i=1}^{n}\mathrm{p}(\theta_i|\theta_{i-1})\right). \end{aligned} \tag{5.67}$$

Similar to the earlier problems, we consider the following linear observation model:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}. \tag{5.68}$$

Here, the random vector $\mathbf{w}$ is jointly Gaussian distributed with $\mathcal{N}(\mathbf{w}; \mathbf{0}, \mathbf{C_n})$. Our initial goal is to derive the posterior distribution of $\mathbf{x}$ and $\boldsymbol{\theta}$ given the vector $\mathbf{y}$.

$$\begin{aligned} \mathrm{p}(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) &= \frac{\mathrm{p}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y})}{\mathrm{p}(\boldsymbol{y})} \\ &= \frac{\mathrm{p}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})\mathrm{p}(\mathbf{x}, \boldsymbol{\theta})}{\int_{\mathbf{x}} \mathrm{p}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{y})\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta}} \\ &= \frac{\mathrm{p}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})\mathrm{p}(\mathbf{x}, \boldsymbol{\theta})}{\int_{\mathbf{x}} \mathrm{p}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})\mathrm{p}(\mathbf{x}, \boldsymbol{\theta})\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta}} \\ &= \frac{\mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C_n})\mathrm{p}(\mathbf{x}, \boldsymbol{\theta})}{\int_{\mathbf{x}} \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C_n})\mathrm{p}(\mathbf{x}, \boldsymbol{\theta})\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta}} \\ &= \frac{\mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C_n})\left(\prod_{i=1}^{n}\mathcal{N}(x_i; \mu_{\theta_i}, \gamma_{\theta_i}^2)\right)\left(\mathrm{p}(\theta_0)\prod_{i=1}^{n}\mathrm{p}(\theta_i|\theta_{i-1})\right)}{\int_{\mathbf{x}} \mathcal{N}(\mathbf{y}; \mathbf{A}\mathbf{x}, \mathbf{C_n})\mathrm{p}(\mathbf{x}, \boldsymbol{\theta})\,\mathrm{d}\mathbf{x}\,\mathrm{d}\boldsymbol{\theta}}. \end{aligned} \tag{5.69}$$

The posterior density of $\boldsymbol{x}$ can be derived by marginalizing $\mathrm{p}(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$ over $\boldsymbol{\theta}$. The posterior computation is not feasible due to exponentially increasing total number of components. Our aim is to approximate the posterior density calculation with the expectation propagation method.

We assume that the posterior density of the random variable x_i can be approximated with a Gaussian density. Furthermore, the posterior density of θ_i is assumed to be independent from the posterior density of x_i. With these assumptions,

$$\mathrm{p}(x_i, \theta_i|\mathbf{y}) \approx q(x_i, \theta_i|\mathbf{y}) = q(x_i|\mathbf{y})q(\theta_i|\mathbf{y}) \tag{5.70}$$

where $q(.)$ denotes the approximation to the posterior density. As noted before $q(x_i|\mathbf{y})$ is the Gaussian density with mean μ_i and variance γ_i^2. The function $q(\theta_i|\mathbf{y})$ is the probability mass function for the posterior density of the binary random variable θ_i.

By utilizing yet another independence assumption, the joint density for the posterior is approximated as follows:

$$\mathrm{p}(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}) \approx q(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}) = \prod_{i=1}^{n} q(x_i, \theta_i|\mathbf{y}). \tag{5.71}$$

The expectation propagation aims to refine this approximation sequentially. To do that, a single component, say i^{th} component, is selected for the refinement process. The i^{th} component in approximate density given by (5.71) is replaced with the terms of the i^{th} component appearing in the true posterior density:

$$\mathrm{p}(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}) \approx \frac{1}{K}\left(\prod_{k=1,k\neq i}^{n} q(x_k, \theta_k|\mathbf{y})\right) \mathcal{N}(\mathbf{y}; \mathbf{Ax}, \mathbf{C_n})\mathcal{N}(x_i; \mu_{\theta_i}, \gamma_{\theta_i}^2)\left(\mathrm{p}(\theta_{i+1}|\theta_i)\mathrm{p}(\theta_i|\theta_{i-1})\right). \tag{5.72}$$

Here, K is a normalizing constant which is equal to the integral of the numerator over $\mathbf{x}$ and $\boldsymbol{\theta}$.

The main idea of this method is to improve the approximate density via the utilization of terms from the true posterior. It should be noted that the terms from the true posterior establish "connections" between random variables and the joint density given in (5.72) is no longer multiplication of individual marginals.

1. To update $q(x_i|\mathbf{y})$, we find the marginal of the refined approximation, that is we integrate (5.72) with respect to $\mathbf{x}$ except x_i (denoted as $\mathbf{x}^{\setminus i}$) and $\boldsymbol{\theta}$.

$$q^u(x_i|\mathbf{y}) = \left(\int \mathrm{d}\mathbf{x}^{\setminus i} \frac{1}{K} \prod_{k=1,k\neq i}^{n} q(x_k|\mathbf{y})\mathcal{N}(\mathbf{y}; \mathbf{Ax}, \mathbf{C_n})\right) \left(\int \mathrm{d}\theta_{i-1}\, \mathrm{d}\theta_i\, \mathrm{d}\theta_{i+1}\, q(\theta_{i-1}|\mathbf{y})q(\theta_{i+1}|\mathbf{y})\mathcal{N}(x_i; \mu_{\theta_i}, \gamma_{\theta_i}^2)\left(\mathrm{p}(\theta_{i+1}|\theta_i)\mathrm{p}(\theta_i|\theta_{i-1})\right)\right). \tag{5.73}$$

The second line of (5.73) can be written as

$$\begin{aligned}\int \mathrm{d}\theta_{i-1}\,\mathrm{d}\theta_i\,\mathrm{d}\theta_{i+1}\; q(\theta_{i-1}|\mathbf{y})q(\theta_{i+1}|\mathbf{y})\mathcal{N}(x_i;\mu_{\theta_i},\gamma^2_{\theta_i})\,(\mathrm{p}(\theta_{i+1}|\theta_i)\mathrm{p}(\theta_i|\theta_{i-1}))\\ = \mathcal{N}(x_i;\mu_L,\gamma_L^2)\,[\gamma(L,L,L)+\gamma(H,L,L)+\gamma(L,L,H)+\gamma(H,L,H)]\\ +\mathcal{N}(x_i;\mu_H,\gamma_H^2)\,[\gamma(L,H,L)+\gamma(H,H,L)+\gamma(L,H,H)+\gamma(H,H,H)]\end{aligned} \tag{5.74}$$

where

$$\gamma(A,B,C) = q(\theta_{i-1}=A|\mathbf{y})q(\theta_{i+1}=C|\mathbf{y})\mathrm{p}(\theta_{i+1}=C|\theta_i=B)\mathrm{p}(\theta_i=B|\theta_{i-1}=A). \tag{5.75}$$

Hence, the second line of (5.73) is a mixture of 2 Gaussians. Denoting the mixture of Gaussians as $w_L\mathcal{N}(x_i;\mu_L,\gamma_L^2)+w_H\mathcal{N}(x_i;\mu_H,\gamma_H^2)$ and inserting into (5.73), we get

$$q^u(x_i|\mathbf{y}) = \int \mathrm{d}\mathbf{x}^{\setminus i}\,\frac{1}{K}\mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})[w_L\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_\mathbf{L},\mathbf{C_L})+w_H\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_\mathbf{H},\mathbf{C_H})] \tag{5.76}$$

where

$$\begin{aligned}\boldsymbol{\mu}_\mathbf{L} &= \begin{bmatrix}\mu_1 & \mu_2 & \dots & \mu_{i-1} & \mu_L & \mu_{i+1} & \dots & \mu_n\end{bmatrix}^T\\ \boldsymbol{\mu}_\mathbf{H} &= \begin{bmatrix}\mu_1 & \mu_2 & \dots & \mu_{i-1} & \mu_H & \mu_{i+1} & \dots & \mu_n\end{bmatrix}^T\\ \mathbf{C_L} &= \mathrm{diag}(\gamma_1^2,\gamma_2^2,...,\gamma_{i-1}^2,\gamma_L^2,\gamma_{i+1}^2,...,\gamma_n^2)\\ \mathbf{C_H} &= \mathrm{diag}(\gamma_1^2,\gamma_2^2,...,\gamma_{i-1}^2,\gamma_H^2,\gamma_{i+1}^2,...,\gamma_n^2)\\ w_L &= \gamma(L,L,L)+\gamma(H,L,L)+\gamma(L,L,H)+\gamma(H,L,H)\\ w_H &= \gamma(L,H,L)+\gamma(H,H,L)+\gamma(L,H,H)+\gamma(H,H,H).\end{aligned} \tag{5.77}$$

The integrand of (5.76) can be simplified as

$$\hat{p}_L\mathcal{N}(\mathbf{x};\hat{\boldsymbol{\mu}}_\mathbf{L},\hat{\mathbf{C}}_\mathbf{L})+\hat{p}_H\mathcal{N}(\mathbf{x};\hat{\boldsymbol{\mu}}_\mathbf{H},\hat{\mathbf{C}}_\mathbf{H}) \tag{5.78}$$

where

$$\begin{aligned}\mathbf{S_i} &= \mathbf{A}\mathbf{C_i}\mathbf{A^T}+\mathbf{C_n}\\ \hat{\boldsymbol{\mu}}_\mathbf{i} &= \boldsymbol{\mu}_\mathbf{i}+\mathbf{C_i}\mathbf{A^T}\mathbf{S_i^{-1}}(\mathbf{y}-\mathbf{A}\boldsymbol{\mu}_\mathbf{i})\\ \hat{\mathbf{C}}_\mathbf{i} &= \mathbf{C_i}-\mathbf{C_i}\mathbf{A^T}\mathbf{S_i^{-1}}\mathbf{A}\mathbf{C_i}\end{aligned} \tag{5.79}$$

and

$$\hat{p}_i = \frac{w_i\mathcal{N}(\mathbf{y};\mathbf{A}\boldsymbol{\mu}_\mathbf{i},\mathbf{S_i})}{w_L\mathcal{N}(\mathbf{y};\mathbf{A}\boldsymbol{\mu}_\mathbf{L},\mathbf{S_L})+w_H\mathcal{N}(\mathbf{y};\mathbf{A}\boldsymbol{\mu}_\mathbf{H},\mathbf{S_H})} \tag{5.80}$$

for $i=\{L,H\}$. Finally, by integrating over $\mathbf{x}^{\setminus i}$, we get

$$q^u(x_i|\mathbf{y}) = \hat{p}_L\mathcal{N}(x_i;\hat{\mu}_{L,i},\hat{\gamma}^2_{L,i})+\hat{p}_H\mathcal{N}(x_i;\hat{\mu}_{H,i},\hat{\gamma}^2_{H,i}). \tag{5.81}$$

In the final relation, $\hat{\mu}_{L,i}$ is the i^{th} entry of the vector $\hat{\boldsymbol{\mu}}_{\boldsymbol{L}}$ and $\hat{\gamma}^2_{L,i}$ is the (i,i) element of matrix $\hat{\mathbf{C}}_{\mathbf{L}}$. Similar definitions apply for $\hat{\mu}_{H,i}$ and $\hat{\gamma}^2_{H,i}$. The mean and variance of $\mathrm{p}(x_i|\mathbf{y})$ can be written as

$$\begin{aligned}\mathrm{E}[x_i|\mathbf{y}] &= \hat{p}_L\hat{\mu}_{L,i} + \hat{p}_H\hat{\mu}_{H,i} \\ \mathrm{var}[x_i|\mathbf{y}] &= \hat{p}_L\hat{\gamma}^2_{L,i} + \hat{p}_H\hat{\gamma}^2_{H,i} + [\hat{p}_L\hat{\mu}^2_{L,i} + \hat{p}_H\hat{\mu}^2_{H,i} - (\hat{p}_L\hat{\mu}_{L,i} + \hat{p}_H\hat{\mu}_{H,i})^2].\end{aligned} \tag{5.82}$$

2. To update $q(\theta_i|\mathbf{y})$, we find the marginal of the refined approximation, that is we integrate (5.72) with respect to $\mathbf{x}$ and $\boldsymbol{\theta}$ except θ_i (denoted as $\boldsymbol{\theta}^{\setminus i}$).

$$\begin{aligned}q^u(\theta_i|\mathbf{y}) = &\left(\int \mathrm{d}\mathbf{x}\, \frac{1}{K} \prod_{k=1,k\neq i}^{n} q(x_k|\mathbf{y})\mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})\right) \\ &\left(\int \mathrm{d}\theta_{i-1}\, \mathrm{d}\theta_{i+1}\, q(\theta_{i-1}|\mathbf{y})q(\theta_{i+1}|\mathbf{y})\mathcal{N}(x_i;\mu_{\theta_i},\gamma^2_{\theta_i})\left(\mathrm{p}(\theta_{i+1}|\theta_i)\mathrm{p}(\theta_i|\theta_{i-1})\right)\right).\end{aligned} \tag{5.83}$$

We start with the evaluation of $q^u(\theta_i = L|\mathbf{y})$:

$$\begin{aligned}q^u(\theta_i = L|\mathbf{y}) &= \left(\int \mathrm{d}\mathbf{x}\, \frac{1}{K} \prod_{k=1,k\neq i}^{n} q(x_k|\mathbf{y})\mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})\right) \\ &\quad \left(\int \mathrm{d}\theta_{i-1}\mathrm{d}\theta_{i+1} q(\theta_{i-1}|\mathbf{y})q(\theta_{i+1}|\mathbf{y})\right. \\ &\qquad \left.\mathcal{N}(x_i;\mu_L,\gamma^2_L)\left(\mathrm{p}(\theta_{i+1}|\theta_i = L)\mathrm{p}(\theta_i = L|\theta_{i-1})\right)\right) \\ &= \left(\int \mathrm{d}\mathbf{x}\, \frac{1}{K} \prod_{k=1,k\neq i}^{n} q(x_k|\mathbf{y})\mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})\mathcal{N}(x_i;\mu_L,\gamma^2_L)\right) \\ &\quad \left(\int \mathrm{d}\theta_{i-1}\, \mathrm{d}\theta_{i+1}\, q(\theta_{i-1}|\mathbf{y})q(\theta_{i+1}|\mathbf{y})\right. \\ &\qquad \left.\left(\mathrm{p}(\theta_{i+1}|\theta_i = L)\mathrm{p}(\theta_i = L|\theta_{i-1})\right)\right) \\ &= \left(\int \mathrm{d}\mathbf{x}\, \frac{1}{K}\mathcal{N}(\mathbf{y};\mathbf{A}\mathbf{x},\mathbf{C_n})\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_\mathbf{L},\mathbf{C_L})\right) w_L \\ &= \frac{\mathcal{N}(\mathbf{y};\mathbf{A}\boldsymbol{\mu}_\mathbf{L},\mathbf{S_L})\, w_L}{\mathcal{N}(\mathbf{y};\mathbf{A}\boldsymbol{\mu}_\mathbf{L},\mathbf{S_L})\, w_L + \mathcal{N}(\mathbf{y};\mathbf{A}\boldsymbol{\mu}_\mathbf{H},\mathbf{S_H})\, w_H}\end{aligned} \tag{5.84}$$

where the intermediate variables are as defined in Equation (5.77) and (5.79).

Repeating the same calculations for $q^u(\theta_i = H|\mathbf{y})$, we get

$$\begin{bmatrix} q^u(\theta_i = L|\mathbf{y}) \\ q^u(\theta_i = H|\mathbf{y}) \end{bmatrix} \propto \begin{bmatrix} \mathcal{N}(\mathbf{y};\mathbf{A}\boldsymbol{\mu}_\mathbf{L},\mathbf{S_L})\, w_L \\ \mathcal{N}(\mathbf{y};\mathbf{A}\boldsymbol{\mu}_\mathbf{H},\mathbf{S_H})\, w_H \end{bmatrix}. \tag{5.85}$$

From these results, the refined density is found as below.

$$\begin{bmatrix} q^u(\theta_i = L|\mathbf{y}) \\ q^u(\theta_i = H|\mathbf{y}) \end{bmatrix} = \begin{bmatrix} \hat{p}_L \\ \hat{p}_H \end{bmatrix}. \tag{5.86}$$

5.2.3.1 Performance Comparison of Sparse Channel Estimation Methods

We have implemented the expectation propagation algorithm by taking the measurement matrix $\mathbf{A}$ to be an $m \times n$ Toeplitz structured matrix which uses length-7 MPS code as the training sequence. $\mathbf{x} \in \mathbb{R}^n$ is a sparse channel whose active and non-active components obey a Markov chain. Channel taps are drawn from a narrow Gaussian $\mathcal{N}(\mu_L, \gamma_L^2)$ or a broad Gaussian $\mathcal{N}(\mu_H, \gamma_H^2)$. Stopping criterion in OMP algorithm is set to channel sparsity level k where k is calculated by considering the estimated channel taps satisfying condition $|\hat{x}_i| > 0.1$. Making a good guess for the parameter λ in the LASSO method is difficult due to the channel distribution. Because of this reason, for each iteration, λ is chosen such that its value is consistent with the l_1 norm of the sparse channel. Measurements are corrupted by additive white Gaussian noise $\mathbf{w}$ with $\mathcal{N}(\mathbf{0}, \sigma_n^2\mathbf{I})$.

Parameter settings are as follows: $n = 50$, $l_t = 7$, $m = 56$, $p_H = 0.2$, $\pi_H = 0.2$, $p_{HL} = 0.4$, $p_{LH} = 0.1$, $(\mu_L = 0, \gamma_L^2 = 0.01)$, and $(\mu_H = 3, \gamma_H^2 = 4)$. For each curve, 1000 independent Monte Carlo trials have been conducted.

Figure 5.12 presents the channel estimation error that uses RMSE standard for the LS, MMSE, LASSO, OMP, expectation propagation method, expectation propagation method modified for the Markov chains and clairvoyant case. As mentioned before, note that performances of the OMP and LASSO methods depend on the selection of the stopping criteria used for halting the iterations and parameter λ, respectively. From the figure, we can conclude that performance of the expectation propagation method modified for the Markov chains is better than the rest of the techniques except the clairvoyant estimator, as expected.

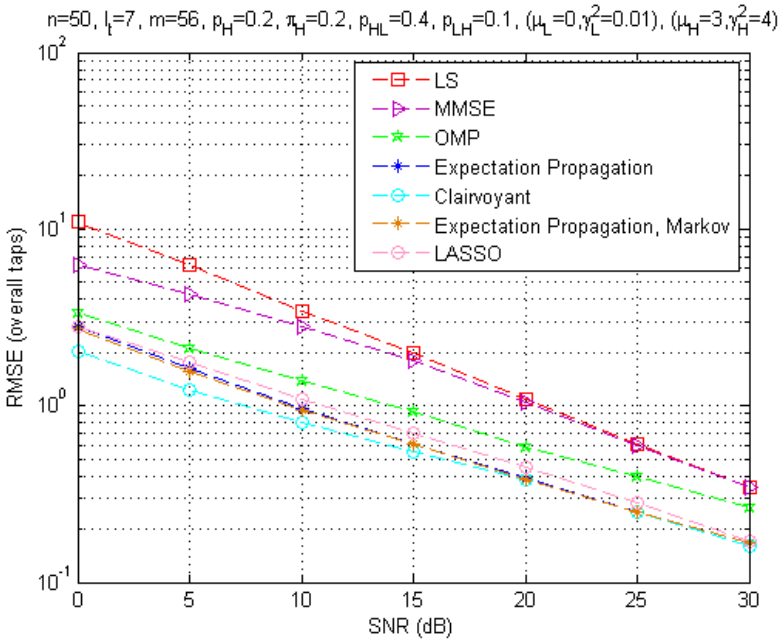

Figure 5.12: Performance comparison of different channel estimation methods at different SNR's when channel taps are correlated

CHAPTER 6

CONCLUSION

In order to provide reliability and high data rates at the receiver, a communication system needs an accurate estimate of the channel. In this thesis, our focus is the estimation of sparse channels. For sparse channels, among the large number of entries, only a small portion of them is significantly different from zero and by taking advantage of the sparsity, the channel impulse response can be recovered from relatively small number of received data and training sequences [8].

The conventional linear least squares solution to this problem is generally non-sparse. To obtain sparse solutions, there is a need for more sophisticated algorithms. Recent studies have demonstrated that, in many cases of interest, there are algorithms that can find good solutions to sparse approximation problems in a reasonable time.

Mathematically, the sparsest channel estimation can be obtained by solving the l_0 sparse constraint problem. However, finding the sparsest solution is an NP-Hard combinatorial problem [26]. In order to find a suboptimal but a sufficient sparse solution, several greedy algorithms and convex relaxation methods have been proposed. Their goal is to obtain not only an accurate but also the sparsest possible estimate.

OMP [30] is the simplest effective algorithm among the greedy suboptimal methods. We have mentioned that OMP relies on picking the atoms of the measurement matrix $\mathbf{A}$ that has the maximum correlation with the residual. However, in order to generate a set of distinct sparse approximations, a different approach is needed to randomize the choice of the next atom. Rather than choosing the atom that maximizes the correlation, one can choose the atom at random with a probability proportional to $|\mathbf{a_i}^T\ \mathbf{y}^t|$ which is the case in [37].

We have declared that the stopping criterion of the greedy algorithms can consist of either a limit on the number of iterations, which also limits the number of non-zeros in $\hat{\mathbf{x}}$, or a requirement that $\mathbf{y} \approx \mathbf{A}\hat{\mathbf{x}}$ in some sense.

The major advantages of OMP algorithm are such that it is fast and easy to implement. However, according to the sufficient condition developed by Tropp and Gilbert in [30], OMP suffers from Mutual Incoherent Property (MIP) and interference due to

coherency. As a result of large MIP, it is possible to find an inaccurate estimate.

The convex relaxation based sparse channel estimation methods find signal representations in over-complete dictionaries and reconstructs the channel by minimizing the l_1 norm. They work correctly as long as the RIP is satisfied. They solve the optimization problem by linear programming, however their complexity is very high and hard to implement.

Greedy pursuits and convex optimization approaches are computationally practical and lead to provably correct solutions under well defined conditions. Furthermore, approximate Bayesian estimation methods can also be used to handle sparse and sparse clustered channels. If additional information on channel taps are available for example if the channel taps are composed of a mixture of Gaussians, an approximate Bayesian inference algorithm, namely expectation propagation, results in a more accurate estimate. Simulation results ensure that performance of the approximate Bayesian inference approach is close to the case where clairvoyant knowledge of the channel covariance is available.

REFERENCES

[1] R. Garg and R. Khandekar. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 337–344. ACM, 2009.

[2] Y. Zhou, M. Herdin, A. M. Sayeed, and E. Bonek. Experimental study of mimo channel statistics and capacity via the virtual channel representation. *IEEE Transactions on Wireless Communications*, 2006.

[3] J. Kivinen, P. Suvikunnas, L. Vuokko, and P. Vainikainen. Experimental investigations of mimo propagation channels. In *IEEE Antennas and Propagation Society International Symposium*, volume 3, page 206. IEEE, 2002.

[4] A. F. Molisch, J. R. Foerster, and M. Pendergrass. Channel models for ultrawideband personal area networks. *IEEE Transactions on Wireless Communications*, 10(6):14–21, 2003.

[5] C. R. Berger, S. Zhou, J. C. Preisig, and P. Willett. Sparse channel estimation for multicarrier underwater acoustic communication: from subspace methods to compressed sensing. *IEEE Transactions on Signal Processing*, 58(3):1708–1721, 2010.

[6] R. Steele and L. Hanzo. *Mobile radio communications: second and third generation cellular and WATM systems*. IEEE Press-John Wiley, 1999.

[7] Y. Liu and D. Borah. Estimation of fading channels with large possible delay spreads. *Electronics Letters*, 39(1):130–131, 2003.

[8] S. F. Cotter and B. D. Rao. Sparse channel estimation via matching pursuit with application to equalization. *IEEE Transactions on Communications*, 50(3):374–377, 2002.

[9] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[10] N. H. Nguyen and T. D. Tran. The stability of regularized orthogonal matching pursuit algorithm, 2007.

[11] C. M. Bishop. *Pattern recognition and machine learning, Sec. 10.7*. Springer,

2006.

[12] T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

[13] T. Minka et al. Divergence measures and message passing. Technical report, Technical Report, Microsoft Research, 2005.

[14] H. Meyr, M. Moeneclaey, and S. A. Fechtel. *Frontmatter and index*. Wiley Online Library, 1998.

[15] E. G. Larsson and P. Stoica. *Space-time block coding for wireless communications*. Cambridge University Press, 2005.

[16] S. M. Kay. Fundamentals of statistical signal processing. 1993.

[17] H. Simon. Adaptive filter theory. *Prentice Hall*, 2:478–481, 2002.

[18] G. Z. Karabulut and A. Yongacoglu. Sparse channel estimation using orthogonal matching pursuit algorithm. In *IEEE 60th Vehicular Technology Conference*, volume 6, pages 3880–3884. IEEE, 2004.

[19] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[20] C. Carbonelli, S. Vedantam, and U. Mitra. Sparse channel estimation with zero tap detection. *IEEE Transactions on Wireless Communications*, 6(5):1743–1763, 2007.

[21] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1999.

[22] Y. C. Eldar and G. Kutyniok. *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

[23] G. Kutyniok. Theory and applications of compressed sensing. *arXiv preprint arXiv:1203.3815*, 2012.

[24] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81, 2009.

[25] M. F. Duarte and Y. C. Eldar. Structured compressed sensing: from theory to applications. *IEEE Transactions on Signal Processing*, 59(9):4053–4085, 2011.

[26] E. J. Candes and T. Tao. Near-optimal signal recovery from random projections: universal encoding strategies. *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[27] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[28] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss. Combining geometry and combinatorics: a unified approach to sparse signal recovery. In *46th Annual Allerton Conference on Communication, Control, and Computing*, pages 798–805. IEEE, 2008.

[29] C. Herzet and A. Drémeau. Bayesian pursuit algorithms. *arXiv preprint arXiv:1401.7538*, 2014.

[30] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[31] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 58(2):1094–1121, 2012.

[32] D. Needell and R. Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):310–316, 2010.

[33] D. Needell and J. A. Tropp. Cosamp: iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[34] J. A. Tropp and S. J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010.

[35] J. Chen and X. Huo. Theoretical results on sparse representations of multiple-measurement vectors. *IEEE Transactions on Signal Processing*, 54(12):4634–4643, 2006.

[36] G. Gui, Q. Wan, W. Peng, and F. Adachi. Sparse multipath channel estimation using compressive sampling matching pursuit algorithm. *arXiv preprint arXiv:1005.2270*, 2010.

[37] M. Elad and I. Yavneh. A plurality of sparse representations is better than the sparsest one alone. *IEEE Transactions on Information Theory*, 55(10):4701–4714, 2009.

[38] M. N. Cohen, M. R. Fox, and J. M. Baden. Minimum peak sidelobe pulse compression codes. In *Record of the IEEE 1990 International Radar Conference*, pages 633–638. IEEE, 1990.

[39] S. Kunis and H. Rauhut. Random sampling of sparse trigonometric polynomials, ii. orthogonal matching pursuit versus basis pursuit. *Foundations of Computational Mathematics*, 8(6):737–763, 2008.

[40] P. Bromiley. Products and convolutions of Gaussian probability density functions. *Tina-Vision Memo*, 3, 2003.

[41] I. Santamaría-Caballero, C. J. Pantaleón-Prieto, and A. Artés-Rodríguez. Sparse deconvolution using adaptive mixed-Gaussian models. *Signal Processing*, 54(2):161–172, 1996.

[42] J. Vila and P. Schniter. Expectation-maximization Bernoulli-Gaussian approximate message passing. In *Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 799–803. IEEE, 2011.

[43] M. J. Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, University of London, 2003.

APPENDIX A

EFFICIENT IMPLEMENTATION OF EXPECTATION PROPAGATION BASED SPARSE CHANNEL ESTIMATION METHOD

The main computational reduction tool is the application of matrix inversion lemma for the inversion of matrices involved in the calculations. The matrix inversion lemma can be written as follows:

$$(\boldsymbol{A}+\boldsymbol{UCV})^{-1}=\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1}+\boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}\boldsymbol{A}^{-1}, \tag{A.1}$$

where $\boldsymbol{A}$, $\boldsymbol{U}$, $\boldsymbol{C}$ and $\boldsymbol{V}$ all denote matrices of compatible sizes.

Here, we use the special case of the lemma for $\boldsymbol{A}=\boldsymbol{A}^T$ (symmetric $\boldsymbol{A}$ matrix) and $\boldsymbol{UCV}=c\boldsymbol{v}\boldsymbol{v}^T$ where c is a scalar and $\boldsymbol{v}$ is a column vector

$$(\boldsymbol{A}+c\boldsymbol{v}\boldsymbol{v}^T)^{-1}=\boldsymbol{A}^{-1}-\frac{(\boldsymbol{A}^{-1}\boldsymbol{v})(\boldsymbol{A}^{-1}\boldsymbol{v})^T}{\frac{1}{c}+\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{v}}. \tag{A.2}$$

In the expectation propagation method, the computationally most complex operation is the computation of the inverse of the matrix $\boldsymbol{S_i}=\boldsymbol{A}\boldsymbol{C_i}\boldsymbol{A}^T+\boldsymbol{C_n}$ for $i=\{L,H\}$, (see (5.40)). It should be remembered that $\boldsymbol{C_i}$ is the covariance matrix at an iteration of the expectation propagation method and it is a diagonal matrix with current variance values on its diagonal.

It can be noted from (5.40) that at an iteration of expectation propagation, the inverses of $\boldsymbol{S_L}=\boldsymbol{A}\boldsymbol{C_L}\boldsymbol{A}^T+\boldsymbol{C_n}$ and $\boldsymbol{S_H}=\boldsymbol{A}\boldsymbol{C_H}\boldsymbol{A}^T+\boldsymbol{C_n}$ are calculated. It should be noted that $\boldsymbol{C_L}$ and $\boldsymbol{C_H}$ only differ at a single entry, which is the diagonal entry whose index corresponds to index of the channel tap whose distribution is being refined by expectation propagation.

If we call, $\mathbf{C_{prev}}=\text{diag}(\gamma_1^2,\gamma_2^2,...,\gamma_n^2)$ as the covariance matrix of the previous expectation propagation step, then to update i^{th} random variable, say x_i, we need to form $\mathbf{C_L}=\text{diag}(\gamma_1^2,\gamma_2^2,...,\gamma_{i-1}^2,\gamma_L^2,\gamma_{i+1}^2,...,\gamma_n^2)$, $\mathbf{C_H}=\text{diag}(\gamma_1^2,\gamma_2^2,...,\gamma_{i-1}^2,\gamma_H^2,\gamma_{i+1}^2,...,\gamma_n^2)$ and insert into $\boldsymbol{S_L}$ and $\boldsymbol{S_H}$ matrices. As an example, the matrix $\boldsymbol{S_L}=\boldsymbol{A}\boldsymbol{C_L}\boldsymbol{A}^T+\boldsymbol{C_n}$

can be written as

$$\boldsymbol{S_L} = \boldsymbol{AC_{prev}A}^T + \boldsymbol{C_n} + (\gamma_L^2 - \gamma_i^2)\boldsymbol{a_i a_i}^T, \tag{A.3}$$

where $\boldsymbol{a_i}$ is i^{th} column of $\boldsymbol{A}$. If the inverse of $\boldsymbol{S_{prev}} = \boldsymbol{AC_{prev}A}^T + \boldsymbol{C_n}$ is available from the earlier iteration of expectation propagation, then

$$\boldsymbol{S_L^{-1}} = \boldsymbol{S_{prev}^{-1}} - \frac{(\boldsymbol{S_{prev}^{-1}a_i})(\boldsymbol{S_{prev}^{-1}a_i})^T}{\frac{1}{(\gamma_L^2 - \gamma_i^2)} + \boldsymbol{a_i}^T\boldsymbol{S_{prev}^{-1}a_i}}. \tag{A.4}$$

Hence, calculation of the required inverse can be accomplished with a rank-1 update on earlier inverse matrix. This can lead to significant computation reductions especially when the number of unknowns, i.e. dimension of the matrix $\boldsymbol{S}$, is large. Similar comments also apply to $\boldsymbol{S_H^{-1}}$.

It can be noted that at each iteration of expectation propagation, two rank-1 updates for the calculation of $\boldsymbol{S_L^{-1}}$ and $\boldsymbol{S_H^{-1}}$ and an another rank-1 update to form the inverse of $\boldsymbol{S_{final}} = \boldsymbol{AC_{final}A}^T + \boldsymbol{C_n}$ with $\mathbf{C_{final}} = \text{diag}(\gamma_1^2, \gamma_2^2, ..., \gamma_{i-1}^2, \gamma_{final}^2, \gamma_{i+1}^2, ..., \gamma_n^2)$ where γ_{final}^2 is the variance of i^{th} random variable after refined as shown in (5.42). The inverse of $\boldsymbol{S_{final}}$ becomes $\boldsymbol{S_{prev}^{-1}}$ in the next iteration.

As a summary, with the application of matrix inversion lemma the computation complexity of each expectation propagation iteration reduces from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$. The reduction can be very significant for n, say, greater than 10 which is almost always the case for the sparse channel estimation.