John Torjus Flåm

The Linear Model under Gaussian Mixture Inputs

Selected Problems in Communications

Thesis for the degree of Philosophiae Doctor

Trondheim, May 2013

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Electronics and Telecommunications

NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Electronics and Telecommunications

ISBN 978-82-471-4421-3 (printed ver.)
ISBN 978-82-471-4422-0 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2013:157

Printed by NTNU-trykk

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for partial fulfillment of the requirements for the degree of philosophiae doctor.

The work has been undertaken at the Signal Processing group of the Department of Electronics and Telecommunications, NTNU, Trondheim. Supervisors were Tor Audun Ramstad, Geir Egil Øien and Lars Lundheim.

Abstract

Consider a linear model, $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$, where $\mathbf{y}$ is an observed vector, $\mathbf{H}$ is a known matrix, $\mathbf{x}$ is the unknown vector of interest, and $\mathbf{n}$ is noise. Such a linear model can describe, or approximate, a multitude of systems.

In this thesis, it is assumed that $\mathbf{x}$ and $\mathbf{n}$ are distributed as independent Gaussian mixtures (GM). Besides their ability to approximate other distributions, Gaussian mixtures can account for asymmetry, heavy tails and/or multi modality. They can in principle model *any* random variable, and therefore Gaussian mixtures provide great realism.

Assuming mixed Gaussian inputs, we study problems related to the minimum mean square error (MMSE) when estimating $\mathbf{x}$ from the observation $\mathbf{y}$. Characterizing or manipulating the MMSE is non-trivial, mainly for the following reason. In the special case when both $\mathbf{x}$ and $\mathbf{n}$ are purely Gaussian inputs, then both the MMSE estimator and the MMSE have analytical, closed form expressions. In the more general case, however, when one or both of the inputs are multi-component Gaussian mixtures, then the MMSE estimator remains analytical, but the MMSE does not. The implication is that the optimal estimator can be implemented, but its performance cannot be exactly characterized.

One consequence, is that implementing MMSE reducing measures becomes a quite difficult task. For example, again in the purely Gaussian setting, an MMSE reducing linear precoder can be derived as the solution of a convex program. When inputs are Gaussian mixtures, however, this task is much more difficult. Then the problem not only turns non-convex, but the objective function (the MMSE) takes the form of a non-analytical integral.

Among the contributions of the thesis, two important ones can be summarized as follows: (i) We bound the MMSE, both from above and below, by analytical expressions. We also show that these bounds approach each other with increasing signal to noise ratio (SNR). Therefore, from moderate to high SNR, the MMSE can be bracketed rather accurately. (ii) We describe a procedure for designing the matrix $\mathbf{H}$, so as to minimize MMSE. This design problem is motivated by two applications in signal processing. One concerns the design of error-reducing precoders; the other deals with selection of pilot signals for channel estimation.

Acknowledgements

Many thanks to my supervisors Tor A. Ramstad, Geir E. Øien and Lars Lundheim for accepting me as a PhD student at NTNU. The freedom and encouragement they have given me has been invaluable.

Likewise, many thanks to the other members of the Signal Processing group at NTNU. Among the academic staff, I would especially like to mention Kimmo Kansanen, Torbjörn Ekman and Ralf Müller who repeatedly took time to discuss ideas with me. Among the Ph.D. students, sincere thanks to Babak Moussakhani and Trung Hieu Nguyen. I am truly glad that we shared the same office.

During my studies, I was generously accepted as visitor to the KTH School of Electrical Engineering on two different occasions. There I had the pleasure of collaborating with the members of the Signal Processing and Communication Theory groups. For this I am very grateful. I am particularly indebted to Saikat Chatterjee for his enthusiasm, his interest in my work, and the good times our families shared.

Financial support from the Research Council of Norway under the NORDITE/VERDIKT program, Project CROPS2 (Grant 181530/S10), and from the European Community's Seventh Framework Program (FP7/2007-2013) under grant 216076 (FP7-SENDORA) is gratefully acknowledged.

Finally, a most deserved thanks to my family; Kathrine for being a wonderful wife; our children Theodor and Thora for the endless joy they give us; and my parents Martha and Sjur for everything.

Trondheim, 2013
John Torjus Flåm

Contents

Part I

Introduction

0.1 Project Background

The radio spectrum is a precious and limited resource, and the demand for wireless services is steadily increasing. In spite of this, parts of the radio spectrum are inefficiently used [1, 2]. One reason is that many wireless communication systems have been assigned fixed portions of the spectrum. This protects the system against interference, but may also cause under-utilization because the actual usage is often sporadic across time and across geographical areas [3, 4].

The main idea behind cognitive radio, first proposed by Mitola and Maguire [5], is a transceiver which opportunistically exploits available radio resources. In a broad context, cognitive radio refers to a transceiver which can sense its environment, and adjust its operation accordingly. In [6], a cognitive radio is regarded as a transceiver which can exploit available frequency bands; it should be able to sense a vacant band, use it for transmission, and back off when it senses that the band is required by a licensed radio. If such a technology could be implemented, it would greatly improve the way in which the radio spectrum is utilized.

Clearly, it is challenging to develop a radio as described above [6, 7]. The sensing capabilities are particularly difficult to realize, especially for a stand-alone transceiver [8]. Therefore, it has been proposed to support the radio by a sensor network [6]. A sensor network consisting of many distributed nodes, can potentially scan multiple frequency bands, at multiple different locations simultaneously. The data collected by the sensors can in turn be used to decide if parts of the spectrum are available for usage.

The work presented in the thesis was mainly undertaken as a part of the CROPS 2 project [6]. The target application for this project was precisely to utilize sensor networks to support the operation of cognitive radios.

0.2 What this thesis is about

Reflecting on the above, a multitude of interesting and diverse research directions can be justified: [6] points out and describes many of these. This thesis focuses on some of the *inference* problems that the target application in [6] would entail: Regardless of what phenomenon the sensor network monitors, the data it collects shall invariably serve to estimate some quantity or to support some decision.

Hence, a number of statistical signal processing problems can be envisioned. We investigate some of these, for very general classes of signals, under a Bayesian framework. In particular, we focus on the linear model,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n},$$

where $\mathbf{y}$ is an observed vector, $\mathbf{H}$ is a known matrix, $\mathbf{x}$ is the unknown signal of interest, and $\mathbf{n}$ is noise.

Such a linear model can represent a multitude of systems, and it is therefore frequently seen in the engineering literature. Here, we mention just a few examples.

In telecommunications, the linear model commonly represents the input/output relation of a communication system [9]. In this case, the signal $\mathbf{x}$ carries the information that should be conveyed from the transmitter to the receiver. $\mathbf{H}$ represents the channel, which in general is time-varying. For sufficiently short periods, however, it can always be regarded as constant. The disturbance, $\mathbf{n}$, models background noise and/or interference.

In image processing [10], $\mathbf{x}$ represents a patch of an image to be estimated, $\mathbf{H}$ is typically a non-invertible degradation operator, and the noise is represented by $\mathbf{n}$. Since the model often is underdetermined, good patch models (a priori knowledge) become particularly important.

In speech enhancement [11], $\mathbf{Hx}$ typically models the clean speech signal. The columns of $\mathbf{H}$ represent basis vectors, whereas the coefficients in $\mathbf{x}$ determine how these combine. The noise is, as usual, collected in $\mathbf{n}$.

Although we have mentioned some applications where the linear model is useful, many, if not most, problems in engineering are actually governed by *non-linear* equations. Yet, linear models are widely used also in these cases. The simple explanation is that non-linear systems are much more difficult to analyze. A more reasonable explanation is that many non-linear systems behave almost linearly for the input range of interest. As an example, think of high fidelity stereo amplifiers. Clearly, higher order terms do exist, but the linear terms tend to dominate for most inputs. Finally, it is not uncommon to approximate systems to first order, around some point of interest. Such approximations are of course not exact, but the toolbox for linear models is much richer than for non-linear ones.

Throughout the thesis, $\mathbf{x}$ and $\mathbf{n}$ are assumed to be statistically independent and Gaussian mixture (GM) distributed. Besides their ability to accurately approximate other distributions, Gaussian mixtures can account for asymmetry, heavy tails, and/or multi modality. They can in principle model *any* random signal, and therefore Gaussian mixtures provide great realism [12], [13]. In a communication setting, for example, a GM distributed $\mathbf{x}$ can model most signal constellations: the limiting form of a GM distribution is a mixture of Dirac pulses (e.g. a discrete distribution). The latter can, of course, represent most signal constellations that we may imagine. As for the noise, $\mathbf{n}$, a GM distribution may model any type of disturbance - in particular, it may model phenomena that are *not* accurately described by a purely Gaussian distribution. For example, the multiple access interference in a dense ad hoc network is known to be non-Gaussian [14].

Assuming mixed Gaussian inputs, the present thesis studies problems related to the *minimum mean square error* (MMSE) when estimating $\mathbf{x}$ from the observation $\mathbf{y}$. In this case, the MMSE estimator has a closed form analytical expression, but the MMSE does not. The implication is that the optimal estimator can be implemented [10, 15–26], but its performance is harder to assess. The existing literature has, as far as we can see, neither properly bounded the MMSE, nor has it proposed MMSE reducing measures, such as

linear precoders. This thesis makes an attempt at contributing in these areas.

Above, we have drawn motivation mainly from communications and signal processing. For many of the problems that we will consider in the thesis, however, the analysis is essentially application independent. Thus, it should be relevant across a large variety of domains.

0.3 Thesis outline

The thesis has two main parts: This introduction, and a collection of papers. The introduction provides context and theoretical background. Its remaining part is outlined as follows. Section 0.4 indicates some of the related literature. Section 0.5 defines what a GM distribution is, and collects some properties of GM distributed random vectors. Section 0.6 lays the foundation for most of the problems that this thesis considers. It does so by recalling the MMSE estimator, and its associated MMSE, for the linear model under GM inputs. Complementing the literature, which tends to present this rather briefly, a detailed and full derivation is provided here. Section 0.7 discusses how the results of the thesis can be used, and in which contexts they could be interpreted. Section 0.8 lists the collected papers, together with a summary (extended abstract) of each paper. Section 0.8 also mentions the papers that the author has contributed to, but chosen not to include in the thesis.

The second part, and main body of the thesis, is a collection of papers, each paper corresponding to one chapter. To a large extent, the papers are self contained. Consequently, when assembled together, the reader must cope with some overlap and repetition. The advantage is that each chapter can be read independently from the rest.

The second part is outlined as follows. In Chapter 1, we study the problem of localizing a transmitting radio heard by a fixed sensor network. We assume that the prior distribution on the transmitter's location is governed by a GM. In Chapter 2, we derive analytical upper and lower bounds on the MMSE when estimating $\mathbf{x}$ from the observation $\mathbf{y}$. These bounds are general and independent of the underlying application. We show that the upper and lower bounds approach each other with increasing signal-to-noise ratio. In Chapter 3 we consider a matrix design problem. Specifically, we describe a procedure to design the matrix $\mathbf{H}$, under a set of constraints, such that $\mathbf{x}$ can be estimated from $\mathbf{y}$ with as small mean square error (MSE) as possible. This problem is relevant when designing pilot signals for channel estimation, or when designing error reducing linear precoders. In Chapter 4, we also design pilot signals for channel estimation, but we assume that both the noise and the channel are Gaussian (which is a special case of Gaussian mixtures). Unlike some of the existing literature, we propose a procedure which does not assume that the covariance matrices of the channel and the noise factorize as Kronecker products.

A more elaborate summary of each problem considered is given in Section 0.8, but the detailed descriptions are deferred to the introductory part of each of the subsequent chap-

ters.

Apart from a slight change in format (from double column conference/journal format to the single column format of the thesis) the contents of the papers have not been modified. The papers have been incorporated as they were submitted/accepted. However, the author has taken the liberty to remove the bibliography following each original paper, and rather organized an overarching bibliography at the end of the thesis.

0.4 Related groups of literature

This section points to some of the related literature, but it does not offer an extensive survey. As for literature more specific to each problem considered in the thesis, a number of references are provided in each of the corresponding papers.

MMSE estimation, assuming linear models under mixed Gaussian inputs, is studied particularly within three groups of the literature. The first one focuses on state sequence estimation via noisy measurements, presuming statistics that fit the GM paradigm. Selected works include [13, 27–32]. These studies offer (approximate or exact) GM posterior state distributions, the mean always serving as the state estimate. For natural reasons, because the settings are generally non stationary, none of these works analyze the MMSE.

The second group of studies uses GM distributions to simplify processing of speech, audio and images. Selected works include [10, 15–26]. In these, one or more variates are modeled by finite GMs. This is often sufficiently accurate, and allows good practical estimators. However, none of these make an attempt at characterizing or reducing the MMSE.

The third group offers general lower bounds on the MMSE, for any kind of input distributions. Included there, are the Bayesian bounds of Cramer-Rao [33], Bobrovsky-Zakai [34], Bhattacharyya [33], Weiss-Weinstein [35], and Reuven-Messer [36]. These bounds hold for most types of joint probability densities $f(\mathbf{x}, \mathbf{y})$ of practical interest, but they rarely acquire analytical forms. Yet, these bounds are often simpler to evaluate numerically than the MMSE. That feature makes them both attractive and useful. Like the MMSE, the above mentioned bounds all become non-analytical when Gaussian mixtures serve as input to a linear model. In contrast to the usual case, however, they do *not* become simpler to evaluate numerically than the MMSE. Therefore, under such settings, do the above mentioned bounds have limited practical value.

0.5 Gaussian Mixture (GM) distributions

Gaussian mixture distributions appear in all of the papers collected. The literature often uses the equivalent term *Gaussian Mixture Model* (GMM). Gaussian mixtures are interesting for several reasons. Firstly, they can accommodate heavy tails, asymmetry, and/or

multi modality. Secondly, by judicious choice of parameters, a GM can approximate *any* distribution to desired accuracy [12], [13]. Thirdly, GM distributions are commonly assumed in practice because they simplify many problems in signal processing - notably so within speech, audio, and image processing.

Because Gaussian mixtures are central in the thesis, we proceed by formally defining a Gaussian mixture distribution. In addition, some important properties of GM distributed random vectors (to be used throughout the thesis) are collected, in a number of propositions. These show that many of the well known properties of Gaussian random vectors transfer to the Gaussian mixture case. In fact, exploiting the distribution and characteristic function of a Gaussian random vector [37], these propositions are straightforward to derive. Yet, they are not easily found in the literature. While Propositions 1 and 2 appear in similar form in [38], the author could not find the other propositions presented below anywhere else.

For this reason, and also to allow for a straightforward derivation of the MMSE estimator and the MMSE in Section 0.6, the below propositions are collected here for convenience.

In the following, $\mathbf{x}$ denotes a vector in the (finite-dimensional) real Euclidean sample space $\mathbb{X}$. We define all vectors as column vectors, and assume all sample spaces to be continuous.

Definition 1. ***Mixture distribution.***
Let $\mathcal{K}$ be a (finite or infinte) countable index set. For each $k \in \mathcal{K}$, let p_k be the probability of drawing index k from $\mathcal{K}$, and let P_k be a probability distribution (or measure) on $\mathbb{X}$. Then, the convex combination

$$P = \sum_{k \in \mathcal{K}} p_k P_k \tag{1}$$

defines a probability distribution on $\mathbb{X}$. We call (1) a mixture distribution on $\mathbb{X}$. The cardinality of $\mathcal{K}$ determines the number of components in the mixture.

In writing, we will often take the fact that k belongs to an index set as implicit, i.e., we often simply write k, instead of $k \in \mathcal{K}$.

Definition 2. ***Gaussian Mixture (GM) distribution.***
When all the component measures $\{P_k\}$ are Gaussian, we call (1) a Gaussian mixture (GM) distribution. We indicate that a random variable $\mathbf{x}$ is GM distributed by writing

$$\mathbf{x} \sim \sum_k p_k \mathcal{N}(\mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)}). \tag{2}$$

The set $\left\{p_k, \mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)}\right\}_{k \in \mathcal{K}}$ is collectively referred to as the parameters of a Gaussian mixture.

Notation (2) should be read in the distributional sense, where $\mathbf{x}$ results from an imagined two stage experiment. First, source k is activated with probability $p_k \geq 0$, $\sum_k p_k = 1$. Second, that source generates a Gaussian signal with distribution law $\mathcal{N}(\mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)})$. Note, however, that for any observed $\mathbf{x}$, we do not know which underlying Gaussian source has generated it. Also note that in the special case of $|\mathcal{K}| = 1$, then a GM distribution becomes purely Gaussian. In theory, and for much of the analysis in the included papers, $\mathcal{K}$ could well be an infinite set. However, for *implementation* of the MMSE estimator (presuming a linear model under GM inputs) *finite* GMs are necessary on both the signal and the noise. Otherwise, the MMSE estimator will be given as an infinite sum, which cannot be implemented in software. This will become clear in Section 0.6, when we recall the expression for MMSE estimator.

Proposition 1. ***Mean of a mixture.***
Suppose P_k has finite mean

$$\mathbf{u}_{\mathbf{x}}^{(k)} = \int_{\mathbf{x}\in\mathbb{X}} \mathbf{x} dP_k(\mathbf{x}).$$

Then the mixture distribution of (1) has mean

$$\mathbf{u}_{\mathbf{x}} = \sum_k p_k \mathbf{u}_{\mathbf{x}}^{(k)}.$$

Proof.

$$\begin{aligned} \mathbf{u}_{\mathbf{x}} &= \int_{\mathbf{x}\in\mathbb{X}} \mathbf{x} \sum_k p_k dP_k(\mathbf{x}) \\ &= \sum_k p_k \int_{\mathbf{x}\in\mathbb{X}} \mathbf{x} dP_k(\mathbf{x}) \\ &= \sum_k p_k \mathbf{u}_{\mathbf{x}}^{(k)}. \end{aligned}$$

□

Proposition 2. ***Covariance of a mixture.***
Suppose P_k has finite mean $\mathbf{u}_{\mathbf{x}}^{(k)}$, and covariance matrix

$$\mathbf{C}_{\mathbf{xx}}^{(k)} := \int_{\mathbf{x}\in\mathbb{X}} (\mathbf{x} - \mathbf{u}_{\mathbf{x}}^{(k)})(\mathbf{x} - \mathbf{u}_{\mathbf{x}}^{(k)})^T dP_k(\mathbf{x}).$$

Then, the covariance of the mixture distribution (1) is

$$\mathbf{C}_{\mathbf{xx}} = \sum_k p_k \left(\mathbf{C}_{\mathbf{xx}}^{(k)} + \mathbf{u}_{\mathbf{x}}^{(k)} \mathbf{u}_{\mathbf{x}}^{(k)^T} \right) - \mathbf{u}_{\mathbf{x}} \mathbf{u}_{\mathbf{x}}^T.$$

Proof. We use the fact that $\mathbf{C_{xx}} = E(\mathbf{x}\mathbf{x}^T) - E(\mathbf{x})E(\mathbf{x})^T$ by definition. Thus

$$\begin{aligned}
\mathbf{C_{xx}} &= \int_{\mathbf{x}\in\mathbb{X}} \mathbf{x}\mathbf{x}^T \sum_k p_k dP_k(\mathbf{x}) - \mathbf{u_x}\mathbf{u_x}^T \\
&= \sum_k p_k \int_{\mathbf{x}\in\mathbb{X}} \mathbf{x}\mathbf{x}^T dP_k(\mathbf{x}) - \mathbf{u_x}\mathbf{u_x}^T \\
&= \sum_k p_k \left(\mathbf{C}_{\mathbf{xx}}^{(k)} + \mathbf{u}_{\mathbf{x}}^{(k)}\mathbf{u}_{\mathbf{x}}^{(k)T}\right) - \mathbf{u_x}\mathbf{u_x}^T.
\end{aligned}$$

□

Proposition 3. ***Characteristic function of a GM distributed random vector.*** *Let* $\mathbf{x} \sim \sum_k p_k \mathcal{N}(\mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)})$*. Then the characteristic function of* $\mathbf{x}$ *is*

$$\phi(\mathbf{t}) = \sum_k p_k e^{i\mathbf{t}^T\mathbf{u}_{\mathbf{x}}^{(k)} - \frac{1}{2}\mathbf{t}^T\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{t}}.$$

for any real vector $\mathbf{t}$*.*

Proof. For any real vector $\mathbf{t}$, the characteristic function for $\mathbf{x} \sim \mathcal{N}(\mathbf{u_x}, \mathbf{C_{xx}})$ is (see e.g. [37])

$$\phi(\mathbf{t}) = \int e^{i\mathbf{t}^T\mathbf{x}} dP(\mathbf{x}) = e^{i\mathbf{t}^T\mathbf{u_x} - \frac{1}{2}\mathbf{t}^T\mathbf{C_{xx}}\mathbf{t}}$$

where $P = \mathcal{N}(\mathbf{u_x}, \mathbf{C_{xx}})$. Now, if $\mathbf{x} \sim \sum_k p_k \mathcal{N}(\mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)})$, then the characteristic function is

$$\begin{aligned}
\phi(\mathbf{t}) &= E\left(e^{i\mathbf{t}^T\mathbf{x}}\right) \\
&= \int e^{i\mathbf{t}^T\mathbf{x}} \sum_k p_k dP_k(\mathbf{x}) \\
&= \sum_k p_k \int e^{i\mathbf{t}^T\mathbf{x}} dP_k(\mathbf{x}) \\
&= \sum_k p_k e^{i\mathbf{t}^T\mathbf{u}_{\mathbf{x}}^{(k)} - \frac{1}{2}\mathbf{t}^T\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{t}}.
\end{aligned}$$

□

Proposition 4. ***Joint distribution of independent GM distributed random vectors***. *Let* $\mathbf{x} \sim \sum_k p_k \mathcal{N}(\mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)})$ *and* $\mathbf{y} \sim \sum_r q_r \mathcal{N}(\mathbf{u}_{\mathbf{y}}^{(r)}, \mathbf{C}_{\mathbf{yy}}^{(r)})$*, where* $\mathbf{x}$ *and* $\mathbf{y}$ *are mutually independent. Then* $\mathbf{y}$ *and* $\mathbf{x}$ *are jointly GM distributed as*

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} \sim \sum_{k,r} p_k q_r \mathcal{N}\left(\begin{bmatrix} \mathbf{u}_{\mathbf{y}}^{(r)} \\ \mathbf{u}_{\mathbf{x}}^{(k)} \end{bmatrix}, \begin{bmatrix} \mathbf{C}_{\mathbf{yy}}^{(r)} & 0 \\ 0 & \mathbf{C}_{\mathbf{xx}}^{(k)} \end{bmatrix}\right).$$

Proof. By Proposition 3, the characteristic functions of $\mathbf{y}$ and $\mathbf{x}$ are

$$\phi_{\mathbf{y}}(\mathbf{s}) = \sum_r q_r e^{i\mathbf{s}^T\mathbf{u}_{\mathbf{y}}^{(r)} - \frac{1}{2}\mathbf{s}^T\mathbf{C}_{\mathbf{yy}}^{(r)}\mathbf{s}}$$

and

$$\phi_{\mathbf{x}}(\mathbf{t}) = \sum_k p_k e^{i\mathbf{t}^T\mathbf{u}_{\mathbf{x}}^{(k)} - \frac{1}{2}\mathbf{t}^T\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{t}}$$

respectively. Because of the independence, the characteristic function of the joint random vector $[\mathbf{y}^T\mathbf{x}^T]^T$ is

$$\begin{aligned}
\phi_{\mathbf{y},\mathbf{x}}\left(\begin{bmatrix}\mathbf{s}\\ \mathbf{t}\end{bmatrix}\right) &= \phi_{\mathbf{y}}(\mathbf{s})\phi_{\mathbf{x}}(\mathbf{t}) \\
&= \sum_{k,r} p_k q_r e^{i\left(\mathbf{s}^T\mathbf{u}_{\mathbf{y}}^{(r)} + \mathbf{t}^T\mathbf{u}_{\mathbf{x}}^{(k)}\right) - \frac{1}{2}\left(\mathbf{s}^T\mathbf{C}_{\mathbf{yy}}^{(r)}\mathbf{s} + \mathbf{t}^T\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{t}\right)} \\
&= \sum_{k,r} p_k q_r \exp\left(i\begin{bmatrix}\mathbf{s}^T\mathbf{t}^T\end{bmatrix}\begin{bmatrix}\mathbf{u}_{\mathbf{y}}^{(r)}\\ \mathbf{u}_{\mathbf{x}}^{(k)}\end{bmatrix} - \frac{1}{2}\begin{bmatrix}\mathbf{s}^T\mathbf{t}^T\end{bmatrix}\begin{bmatrix}\mathbf{C}_{\mathbf{yy}}^{(r)} & 0\\ 0 & \mathbf{C}_{\mathbf{xx}}^{(k)}\end{bmatrix}\begin{bmatrix}\mathbf{s}\\ \mathbf{t}\end{bmatrix}\right)
\end{aligned}$$

for any real vector $[\mathbf{s}^T\mathbf{t}^T]^T$. □

Proposition 5. ***Affine transform of a GM distributed random vector.***
Let $\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{a}$*, where* $\mathbf{x} \sim \sum_k p_k \mathcal{N}(\mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)})$*,* $\mathbf{D}$ *is a deterministic matrix, and* $\mathbf{a}$ *is a deterministic vector. Then* $\mathbf{y}$ *is distributed as*

$$\mathbf{y} \sim \sum_k p_k \mathcal{N}(\mathbf{D}\mathbf{u}_{\mathbf{x}}^{(k)} + \mathbf{a}, \mathbf{D}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{D}^T).$$

Proof.

$$\begin{aligned}
\phi_{\mathbf{y}}(\mathbf{t}) &= E\left(e^{i\mathbf{t}^T(\mathbf{D}\mathbf{x}+\mathbf{a})}\right) = e^{i\mathbf{t}^T\mathbf{a}} E\left(e^{i\left(\mathbf{D}^T\mathbf{t}\right)^T\mathbf{x}}\right) \\
&= e^{i\mathbf{t}^T\mathbf{a}} \sum_k p_k e^{i\left(\mathbf{D}^T\mathbf{t}\right)^T\mathbf{u}_{\mathbf{x}}^{(k)} - \frac{1}{2}\left(\mathbf{D}^T\mathbf{t}\right)^T\mathbf{C}_{\mathbf{xx}}^{(k)}\left(\mathbf{D}^T\mathbf{t}\right)} \\
&= \sum_k p_k e^{i\mathbf{t}^T\left(\mathbf{D}\mathbf{u}_{\mathbf{x}}^{(k)} + \mathbf{a}\right) - \frac{1}{2}\mathbf{t}^T\mathbf{D}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{D}^T\mathbf{t}}.
\end{aligned}$$

□

Proposition 6. ***Marginal distribution of a GM distribution***.
Let $\mathbf{x} \sim \sum_k p_k \mathcal{N}(\mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)})$*. Partition* $\mathbf{x}$ *into two sub vectors such that*

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix},\ \mathbf{u}_{\mathbf{x}}^{(k)} = \begin{bmatrix}\mathbf{u}_{\mathbf{x}_1}^{(k)}\\ \mathbf{u}_{\mathbf{x}_2}^{(k)}\end{bmatrix},\ \mathbf{C}_{\mathbf{xx}}^{(k)} = \begin{bmatrix}\mathbf{C}_{\mathbf{x}_1\mathbf{x}_1}^{(k)} & \mathbf{C}_{\mathbf{x}_1\mathbf{x}_2}^{(k)}\\ \mathbf{C}_{\mathbf{x}_2\mathbf{x}_1}^{(k)} & \mathbf{C}_{\mathbf{x}_2\mathbf{x}_2}^{(k)}\end{bmatrix}.$$

Then the marginal distribution for $\mathbf{x}_1$ *is* $\sum_k p_k \mathcal{N}(\mathbf{u}_{\mathbf{x}_1}^{(k)}, \mathbf{C}_{\mathbf{x}_1\mathbf{x}_1}^{(k)})$.

Proof. Without loss of generality, assume that $\mathbf{x}_1$ contains the p first elements of $\mathbf{x}$. Let

$$\mathbf{D} = \begin{bmatrix} \mathbf{I}_p & 0 \\ 0 & 0 \end{bmatrix},$$

where $\mathbf{I}_p$ denotes the $p \times p$ identity matrix. Then $\mathbf{x}_1 = \mathbf{D}\mathbf{x}$, and by Proposition 5 the statement is proven. □

The covariance matrix of a Gaussian random vector can, in principle, be singular. Observe, however, that the associated characteristic function does *not* rely on the inverse of the covariance matrix (Proposition 3). The implication is that all of the above propositions also hold when one or more of the component covariance matrices are singular.

0.6 Linear models with GM inputs: MMSE estimation

A recurring system model throughout this thesis is the following linear instance:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}. \tag{3}$$

Here $\mathbf{y}$ is a vector of observations, $\mathbf{H}$ is a known matrix, and $\mathbf{x}$ and $\mathbf{n}$ are mutually independent random vectors with known Gaussian Mixture (GM) distributions:

$$\mathbf{x} \sim \sum_{k \in \mathcal{K}} p_k \mathcal{N}\big(\mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)}\big) \tag{4}$$

$$\mathbf{n} \sim \sum_{l \in \mathcal{L}} q_l \mathcal{N}\big(\mathbf{u}_{\mathbf{n}}^{(l)}, \mathbf{C}_{\mathbf{nn}}^{(l)}\big)\,. \tag{5}$$

The input distributions are chosen for their flexibility and generality. As for the linear model, it represents or approximates to first order, a multitude of systems - also beyond the context of communications and signal processing. Throughout, we are interested in problems related to the minimum mean square error (MMSE) when estimating $\mathbf{x}$, based on the observed output $\mathbf{y}$.

By definition, the MMSE estimator is given by:

$$\mathbf{u}_{\mathbf{x}|\mathbf{y}} \triangleq E\left\{\mathbf{x}|\mathbf{y}\right\} = \int \mathbf{x} f(\mathbf{x}|\mathbf{y}) d\mathbf{x}. \tag{6}$$

Here $f(\mathbf{x}|\mathbf{y})$ is the probability density function (PDF) of $\mathbf{x}$ given $\mathbf{y}$. It is often termed the *posterior* PDF. Clearly, the MMSE estimator (6) equals the posterior mean. Its associated performance measure, is

$$\begin{aligned} \text{MMSE} \triangleq E\left\{\|\mathbf{x} - \mathbf{u}_{\mathbf{x}|\mathbf{y}}\|_2^2\right\} &= \iint \|\mathbf{x} - \mathbf{u}_{\mathbf{x}|\mathbf{y}}\|_2^2 f(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \\ &= \iint \|\mathbf{x} - \mathbf{u}_{\mathbf{x}|\mathbf{y}}\|_2^2 f(\mathbf{x}|\mathbf{y}) d\mathbf{x} f(\mathbf{y}) d\mathbf{y} = \int \text{Tr}\left(\mathbf{C}_{\mathbf{x}|\mathbf{y}}\right) f(\mathbf{y}) d\mathbf{y}. \end{aligned} \tag{7}$$

Here, $\|\cdot\|_2^2$ is the squared Euclidean norm, $f(\mathbf{x},\mathbf{y})$ is the joint PDF of $\mathbf{x}$ and $\mathbf{y}$, $f(\mathbf{y})$ is the marginal PDF for $\mathbf{y}$, $\mathrm{Tr}(\cdot)$ denotes the trace operator, and $\mathbf{C}_{\mathbf{x}|\mathbf{y}}$ is the covariance matrix of the posterior.

Although the MMSE estimator for $\mathbf{x}$ under these assumptions is known (see e.g. [15, 19, 39, 40]), it tends to be presented rather briefly. For this reason, and to make the thesis more self-contained, we recall the full derivation of both the MMSE estimator (6), and the MMSE (7) here. It will become clear that integral (7), in general, is non-analytical. Hence, the MMSE of estimator (6) has no closed form analytical expression when the inputs are GMs.

In what follows, the arguments are given in terms of the characteristic function of GM distributed random vectors. As far as we know, this is not standard, but it gives a simple and compact derivation. We assume that the matrix and all vectors in (3) are real. For the complex case, the MMSE estimator and the MMSE can be derived in a straightforward manner using e.g. [41, Section 15.8].

As standing assumption, $\mathbf{x}$ and $\mathbf{n}$ are independent and GM distributed as in (4) and (5), and model (3) applies. Then, by Proposition 4, $\mathbf{x}$ and $\mathbf{n}$ are jointly GM distributed as

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{n} \end{bmatrix} \sim \sum_{k,l} p_k q_l \mathcal{N}\left(\begin{bmatrix} \mathbf{u}_{\mathbf{x}}^{(k)} \\ \mathbf{u}_{\mathbf{n}}^{(l)} \end{bmatrix}, \begin{bmatrix} \mathbf{C}_{\mathbf{xx}}^{(k)} & 0 \\ 0 & \mathbf{C}_{\mathbf{nn}}^{(l)} \end{bmatrix} \right).$$

By Proposition 5, a linear transform of a GM vector remains a GM vector. Hence, since (3) amounts to

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{H} & \mathbf{I} \\ \mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{n} \end{bmatrix},$$

the vector $[\mathbf{y}^T\ \mathbf{x}^T]^T$ is GM distributed as well:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} \sim \textstyle\sum_{k,l} p_k q_l \mathcal{N}\left(\begin{bmatrix} \mathbf{H}\mathbf{u}_{\mathbf{x}}^{(k)} + \mathbf{u}_{\mathbf{n}}^{(l)} \\ \mathbf{u}_{\mathbf{x}}^{(k)} \end{bmatrix}, \begin{bmatrix} \mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T + \mathbf{C}_{\mathbf{nn}}^{(l)} & \mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)} \\ \mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T & \mathbf{C}_{\mathbf{xx}}^{(k)} \end{bmatrix} \right).$$

We write the corresponding joint probability density function as

$$f(\mathbf{y},\mathbf{x}) = \sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y},\mathbf{x}), \tag{8}$$

where $f^{(k,l)}(\mathbf{y},\mathbf{x})$ is a Gaussian density with mean

$$\begin{bmatrix} \mathbf{H}\mathbf{u}_{\mathbf{x}}^{(k)} + \mathbf{u}_{\mathbf{n}}^{(l)} \\ \mathbf{u}_{\mathbf{x}}^{(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{\mathbf{y}}^{(k,l)} \\ \mathbf{u}_{\mathbf{x}}^{(k)} \end{bmatrix},$$

and covariance

$$\begin{bmatrix} \mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T + \mathbf{C}_{\mathbf{nn}}^{(l)} & \mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)} \\ \mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T & \mathbf{C}_{\mathbf{xx}}^{(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{\mathbf{yy}}^{(k,l)} & \mathbf{C}_{\mathbf{yx}}^{(k)} \\ \mathbf{C}_{\mathbf{xy}}^{(k)} & \mathbf{C}_{\mathbf{xx}}^{(k)} \end{bmatrix}.$$

Using Proposition 6, the marginal density for $\mathbf{y}$ is, as one might expect,

$$f(\mathbf{y}) = \sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y}). \tag{9}$$

Here $f^{(k,l)}(\mathbf{y})$ is a Gaussian density with mean $\mathbf{u}_{\mathbf{y}}^{(k,l)}$ and covariance $\mathbf{C}_{\mathbf{yy}}^{(k,l)}$. The posterior density follows from Bayes' law as

$$\begin{aligned} f(\mathbf{x}|\mathbf{y}) &= \frac{f(\mathbf{y},\mathbf{x})}{f(\mathbf{y})} = \frac{\sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y},\mathbf{x})}{\sum_{r,s} p_r q_s f^{(r,s)}(\mathbf{y})} \\ &= \frac{\sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y}) f^{(k,l)}(\mathbf{x}|\mathbf{y})}{\sum_{r,s} p_r q_s f^{(r,s)}(\mathbf{y})} \\ &= \sum_{k,l} \alpha^{(k,l)}(\mathbf{y}) f^{(k,l)}(\mathbf{x}|\mathbf{y}), \end{aligned} \tag{10}$$

where

$$\alpha^{(k,l)}(\mathbf{y}) = \frac{p_k q_l f^{(k,l)}(\mathbf{y})}{\sum_{r,s} p_r q_s f^{(r,s)}(\mathbf{y})}. \tag{11}$$

The weight, $\alpha^{(k,l)}(\mathbf{y})$, can be seen as the joint probability of $\mathbf{x}$ and $\mathbf{n}$ originating from components k and l respectively, given the observation $\mathbf{y}$. Note that these weights are non-linear in the observation $\mathbf{y}$. Clearly, $\alpha^{(k,l)}(\mathbf{y}) \geq 0$ and $\sum_{k,l} \alpha^{(k,l)}(\mathbf{y}) = 1$.

In (10), $f^{(k,l)}(\mathbf{x}|\mathbf{y})$ is a conditional density of a multivariate Gaussian density, $f^{(k,l)}(\mathbf{y},\mathbf{x})$. Consequently, $f^{(k,l)}(\mathbf{x}|\mathbf{y})$ is Gaussian (see e.g. Theorem 10.2 of [41]) with mean

$$\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)} = \mathbf{u}_{\mathbf{x}}^{(k)} + \mathbf{C}_{\mathbf{xy}}^{(k)} \left(\mathbf{C}_{\mathbf{yy}}^{(k,l)}\right)^{-1} \left(\mathbf{y} - \mathbf{u}_{\mathbf{y}}^{(k,l)}\right) \tag{12}$$

$$= \mathbf{u}_{\mathbf{x}}^{(k)} + \mathbf{C}_{\mathbf{xx}}^{(k)} \mathbf{H}^T \left(\mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)} \mathbf{H}^T + \mathbf{C}_{\mathbf{nn}}^{(l)}\right)^{-1} \left(\mathbf{y} - \mathbf{H}\mathbf{u}_{\mathbf{x}}^{(k)} - \mathbf{u}_{\mathbf{n}}^{(l)}\right), \tag{13}$$

and covariance

$$\mathbf{C}_{\mathbf{x}|\mathbf{y}}^{(k,l)} = \mathbf{C}_{\mathbf{xx}}^{(k)} - \mathbf{C}_{\mathbf{xy}}^{(k)} \left(\mathbf{C}_{\mathbf{yy}}^{(k,l)}\right)^{-1} \mathbf{C}_{\mathbf{yx}}^{(k)} \tag{14}$$

$$= \mathbf{C}_{\mathbf{xx}}^{(k)} - \mathbf{C}_{\mathbf{xx}}^{(k)} \mathbf{H}^T \left(\mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)} \mathbf{H}^T + \mathbf{C}_{\mathbf{nn}}^{(l)}\right)^{-1} \mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)}, \tag{15}$$

respectively. Clearly, by its expression, the posterior density $f(\mathbf{x}|\mathbf{y})$ of (10) is a GM. By Proposition 1 & 2 that density has mean

$$\mathbf{u}_{\mathbf{x}|\mathbf{y}} = \int \mathbf{x} f(\mathbf{x}|\mathbf{y}) d\mathbf{x} = \sum_{k,l} \alpha^{(k,l)}(\mathbf{y}) \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)}, \tag{16}$$

and covariance

$$\begin{aligned} \mathbf{C}_{\mathbf{x}|\mathbf{y}} &= \int \left(\mathbf{x} - \mathbf{u}_{\mathbf{x}|\mathbf{y}}\right) \left(\mathbf{x} - \mathbf{u}_{\mathbf{x}|\mathbf{y}}\right)^T f(\mathbf{x}|\mathbf{y}) d\mathbf{x} \\ &= \sum_{k,l} \alpha^{(k,l)}(\mathbf{y}) \left(\mathbf{C}_{\mathbf{x}|\mathbf{y}}^{(k,l)} + \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)^T}\right) - \mathbf{u}_{\mathbf{x}|\mathbf{y}} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^T. \end{aligned} \tag{17}$$

The MMSE estimator corresponds to the posterior mean in (16). Its error covariance matrix is given by (17). The trace of this matrix, averaged over all possible observations $\mathbf{y}$ is the MMSE in (7). For the special case when $|\mathcal{K}| = |\mathcal{L}| = 1$ (purely Gaussian input and purely Gaussian noise), the posterior density, $f(\mathbf{x}|\mathbf{y})$, is also Gaussian. Then the mean (16) reduces to

$$\mathbf{u}_{\mathbf{x}|\mathbf{y}} = \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(1,1)} \tag{18}$$

and the covariance (17) takes the form

$$\mathbf{C}_{\mathbf{x}|\mathbf{y}} = \mathbf{C}_{\mathbf{x}|\mathbf{y}}^{(1,1)}. \tag{19}$$

In this case, note from (18) and (13) that the MMSE estimator is linear in $\mathbf{y}$, and from (19) and (15) that the posterior covariance matrix does not depend on $\mathbf{y}$. The latter property makes it straightforward to characterize the MMSE when $f(\mathbf{x}|\mathbf{y})$ is Gaussian.

In the more general case, when $f(\mathbf{x}|\mathbf{y})$ is a multi-component GM, the MMSE estimator (16) is non-linear in the observed data $\mathbf{y}$ because of the data dependent weights $\alpha^{(k,l)}(\mathbf{y})$. Furthermore, because the posterior covariance $\mathbf{C}_{\mathbf{x}|\mathbf{y}}$ in (17) depends on the observation $\mathbf{y}$, the MMSE becomes considerably more difficult to analyze. In fact, in this case, (7) is a non-analytical integral. The implication is that, although the optimal estimator can be implemented, its performance (MMSE) cannot be exactly characterized in closed form.

0.7 How to interpret the results of the thesis

The MMSE estimator in (16) requires central processing. That is, all elements of the observation vector $\mathbf{y}$ must be known to compute the MMSE estimate. When the observation vector $\mathbf{y}$ is received by a sensor network, it could well be that each element of $\mathbf{y}$ is observed by only one sensor. In that case, (16) can only be computed if each sensor forwards its own observation to some central processing unit. Naturally, as the sensor network grows, such signaling rapidly becomes impractical to handle. One alternative is to divide the sensors into smaller groups, and let them share their observations to compute local estimates. These local estimates may in turn be combined to compute a global estimate. Such a distributed approach will, however, inevitably lead to less accurate estimation than the centralized counterpart. Moreover, distributed estimation raises several non-trivial issues of its own. For one, which and how many sensors should form a group? For another, how do local estimates best combine?

Although certainly interesting, the thesis avoids these and related questions by presuming central processing throughout. Thus, in the context of a cognitive radio assisted by a sensor network, our results may have practical or indicative value only for quite small networks. For larger networks, our results serve as a benchmark: Because we consider centralized and optimal estimation, other approaches (distributed estimation included) cannot be better.

Even if such centralized computing may be unrealistic for controlling a cognitive radio, it can more easily be justified for many of the problems within speech, audio and image processing.

0.8 Collected papers and contributions

The main body of the thesis consists of the below listed papers. A summary gives an idea about the problem at hand and the contributions of each paper.

Paper 1

Flåm J.T., Jaldén J., and Chatterjee S. ***"Gaussian Mixture Modeling for Source Localization"***, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2011.

Summary: We wish to localize a transmitting radio based on the received signal strength of a known signature signal at several sensors in known positions. Assuming a wideband and log-normal shadowing model, the received power at sensor i , in dB, is commonly modelled as

$$r_i = -10\gamma \log\left(\|\mathbf{z} - \mathbf{q}_i\|\right) + s_i = m_i(\mathbf{z}) + s_i.$$

Here, $\|\mathbf{z} - \mathbf{q}_i\|$ is the Euclidean distance from the source in unknown position $\mathbf{z}$ to sensor i in known position $\mathbf{q}_i$, γ is the path loss exponent, and $s_i \sim \mathcal{N}(0, \sigma_i^2)$ is a Gaussian random variable accounting for the shadowing between the source and sensor i. We assume then both γ and σ_i^2 are known. With M sensors in the network, the above equation can be written in vector form

$$\mathbf{r} = \mathbf{m}(\mathbf{z}) + \mathbf{s},$$

where $\mathbf{m}(\mathbf{z})$ is the vector with $m_i(\mathbf{z})$ as its i-th component, and $\mathbf{s} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ where $\mathbf{C}$ is generally non-diagonal.

The problem is to estimate the source position $\mathbf{z}$ based on the observation $\mathbf{r}$. In order to solve it, we take a Bayesian approach which does not involve numerical integration. Specifically, we approximate the probability density function of $\mathbf{m}(\mathbf{z})$ by a Gaussian mixture. The approximation allows a two stage estimation technique: Firstly, a closed form MMSE estimator, $\hat{\mathbf{m}}(\mathbf{z})$, for $\mathbf{m}(\mathbf{z})$ is derived. Secondly, an estimate of the source position, $\mathbf{z}$, is obtained by minimizing the Euclidean distance between $\mathbf{m}(\mathbf{z})$ and $\hat{\mathbf{m}}(\mathbf{z})$ using a gradient method. Numerical experiments indicate that this approach requires fewer computations than an (accurate) MMSE estimator based on numerical integration. Yet, it shows comparable accuracy. To the best of our knowledge, it has not been investigated in the context of localization before.

Paper 2

Flåm J.T., Chatterjee S., Kansanen K., and Ekman T. ***"On MMSE Estimation: A Linear Model Under Gaussian Mixture Statistics"***, IEEE Transactions on Signal Processing, Volume: 60 , Issue: 7, 2012.

Summary: Assume the system model governed by equations (3),(4) and (5) of Section 0.6. With known matrix $\mathbf{H}$, the MMSE estimator for $\mathbf{x}$ has analytical form, but in contrast to the familiar case of Gaussian inputs, the MMSE does not. Thus, while the optimal estimator is implementable, its performance is harder to assess. Moreover, in this setup, existing Bayesian lower bounds, such as those of Cramer-Rao, Bobrovsky-Zakai, Bhattacharyya, Weiss-Weinstein and Reuven-Messer become no easier to estimate than the MMSE itself. Our objective is therefore to bound the MMSE *analytically*, both from above and below, and to relate these bounds to the signal-to-noise-ratio.

We find that the MMSE is *lower* bounded by the mean square error of a 'genie-aided' estimator. This imaginary, non-implementable estimator knows precisely which source in the mixture is active, at any time, both for $\mathbf{x}$ and $\mathbf{n}$. It is therefore much better informed than the MMSE estimator. Yet, simulations indicate that the MMSE estimator is comparably accurate for SNRs above a quite modest threshold (approximately 10 dB in our numerical experiments). The *upper* bound is provided by the MSE of the linear MMSE (LMMSE) estimator. In fact, we show that the LMMSE estimator becomes *the* MMSE estimator in two extreme cases: when the SNR is either zero or infinite. However, simulations indicate that the LMMSE estimator is nearly MSE optimal, not only at these extreme points, but for a large range of SNRs. The upper and lower bounds have closed form expressions, and are straightforward to calculate. Because they approach each other with increasing SNR, the MMSE can be determined rather accurately when the SNR is moderate to high.

Paper 3

Flåm J.T., Zachariah D., Vehkaperä M. and Chatterjee S. ***"The Linear Model under Mixed Gaussian Inputs: Designing the Transfer Matrix"***, In review for IEEE Transactions on Signal Processing.

Summary: Suppose the system model defined by equations (3),(4) and (5). The problem is to design the transfer matrix $\mathbf{H}$, under a set of constraints, so as to minimize the mean square error (MSE) when estimating $\mathbf{x}$ from $\mathbf{y}$. This problem has important applications, but faces at least three hurdles. Firstly, even for a fixed $\mathbf{H}$, the minimum MSE (MMSE) has no analytical form. Secondly, the MMSE is generally not convex in $\mathbf{H}$. Thirdly, derivatives of the MMSE w.r.t. $\mathbf{H}$ are hard to obtain. This paper casts the problem as a stochastic program and invokes gradient methods. The procedure combines, iteratively, the idea of sampling and refinement.

The study is motivated by two applications in signal processing. One concerns the choice

of error-reducing precoders; the other deals with selection of pilot matrices for channel estimation. In either setting, our numerical results indicate improved estimation accuracy - markedly better than those obtained by optimal design based on standard linear estimators.

Some implications of the non-convexities of the MMSE are noteworthy, yet, to our knowledge, not well known. For example, there exist cases in which increasing the pilot power leads to worse channel estimates. This paper explains why.

Paper 4

Flåm J.T., Björnson E. and Chatterjee S. ***"Pilot Design for MIMO channel estimation: An Alternative to the Kronecker Structure Assumption"***, Accepted for IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2013.

Summary: Under the assumption of a linear communication model corrupted by Gaussian noise, this work seeks to design a power constrained pilot signal, such that the channel matrix can be estimated with minimum mean square error. The noise and the channel are modeled as independent, multivariate Gaussians, with covariance matrices $\mathbf{C_{nn}}$ and $\mathbf{C_{xx}}$ respectively. Without making limiting assumptions, the associated pilot design problem is generally non-convex.

Unlike some of the literature, we propose an approach which avoids one of these assumptions; namely that of a Kronecker structure on the covariance matrices. The Kronecker structure implies that the covariance matrices factorize as Kronecker products:

$$\mathbf{C_{xx}} = \mathbf{X}_T^T \otimes \mathbf{X}_R \text{ and } \mathbf{C_{nn}} = \mathbf{N}_T^T \otimes \mathbf{N}_R.$$

Such Kronecker factorizations allow for tractable analysis. In general, however, arbitrary covariance matrices do not factor like this. In fact, assuming a Kronecker structure imposes quite severe restrictions on the correlation. The present work offers an alternative approach.

Briefly, the pilot signal is obtained in three main steps. Firstly, we solve a relaxed, but convex, version of the original minimization problem. Secondly, its solution is projected onto the set of feasible pilot signals. Thirdly, we use the projected solution as starting point for an augmented Lagrangian method. Numerical experiments indicate that this procedure *may* produce pilot signals that are far better than those obtained under the Kronecker structure assumption.

First author papers not presented in this thesis

The following paper is, to a large extent, a conference version of Paper 3. It makes use of finite difference approximations instead of exact stochastic gradients. Hence, the Kiefer-

Wolfowitz algorithm comes to replace the more accurate Robbins-Monro procedure. The focus is limited to design of linear precoders. Because paper 3 essentially covers the same topic, but is more elaborate, the following paper has not been included.

Flåm J.T., Vehkaperä M., Zachariah D. and Tsakonas E. ***"Mean Square Error Reduction by Precoding of Mixed Gaussian Input"***, International Symposium on Information Theory and its Applications, 2012.
Abstract: Suppose a vector of observations $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$ stems from independent inputs $\mathbf{x}$ and $\mathbf{n}$, both of which are Gaussian Mixture (GM) distributed, and that $\mathbf{H}$ is a fixed and known matrix. This work focuses on the design of a *precoding* matrix, $\mathbf{F}$, such that the model modifies to $\mathbf{z} = \mathbf{H}\mathbf{F}\mathbf{x} + \mathbf{n}$. The goal is to design $\mathbf{F}$ such that the mean square error (MSE) when estimating $\mathbf{x}$ from $\mathbf{z}$ is smaller than when estimating $\mathbf{x}$ from $\mathbf{y}$. We do this under the restriction $E[(\mathbf{F}\mathbf{x})^T\mathbf{F}\mathbf{x}] \leq P_T$, that is, the precoder cannot exceed an average power constraint. Although the minimum mean square error (MMSE) estimator, for any fixed $\mathbf{F}$, has a closed form, the MMSE does not under these settings. This complicates the design of $\mathbf{F}$. We investigate the effect of two different precoders, when used in conjunction with the MMSE estimator. The first is the linear MMSE (LMMSE) precoder. This precoder will be mismatched to the MMSE estimator, unless $\mathbf{x}$ and $\mathbf{n}$ are purely Gaussian variates. We find that it may provide MMSE gains in some setting, but be harmful in others. Because the LMMSE precoder is particularly simple to obtain, it should nevertheless be considered. The second precoder we investigate, is derived as the solution to a stochastic optimization problem, where the objective is to minimize the MMSE. As such, this precoder is matched to the MMSE estimator. It is derived using the Kiefer-Wolfowitz algorithm, which moves iteratively from an initially chosen $\mathbf{F}_0$ to a local minimizer $\mathbf{F}^*$. Simulations indicate that the resulting precoder has promising performance.

The following two papers neither assume linear models, nor do they assume Gaussian mixture statistics. Therefore, I have chosen not to include them.

Flåm J.T., Kraidy G.M., and Ryan D.J. ***"Using a Sensor Network to Localize a Source under Spatially Correlated Shadowing"***, IEEE 71st Vehicular Technology Conference, 2010.
Abstract: This paper considers the use of a sensor network to estimate the position of a transmitting radio based on the received signal strength at the sensors. A generic path loss model which includes the effects of spatially correlated shadowing is assumed. A weighted likelihood (WL) estimator is proposed, which can be seen as a simplified minimum mean square error (MMSE) estimator. This estimator can be used for localizing a source in a static scenario or it can provide the initial position estimate of a tracking algorithm. The performance of the WL estimator is simulated, and robustness to erroneous assumptions about path loss exponent, shadowing variance and correlation distance is demonstrated.

Flåm J.T., Øien G., Kim A.N., and Kansanen K. ***"Sensor Requirements and Interference***

Management in Cognitive Radio", Presented at European Cooperation in the Field of Scientific and Technical Research (COST 2100), Braunschweig, Germany, 16-18 february, 2009.
Abstract: In sensor network aided cognitive radio, the goal is to find and use licensed frequencies without causing interference, with the aid of sensors deployed for that purpose. A licensed transceiver/primary user (PU) is interfered when a cognitive radio/secondary user (SU), located within a certain range, transmits on the same frequency. The actual distance between these entities can be estimated after localization by the sensor network. But, unless the localization is exact, the distance estimate is uncertain. Therefore, if the same frequency is used, there is a non-zero probability that the SU will interfere with the PU, even if power control is used in an attempt to avoid this. This interference probability is the focus of this paper. Based on uncertain localization, the probability of an SU causing interference to a PU is computed. Knowledge of this probability can, for example, be used for SU power control. We demonstrate that the interference probability not only depends on the SU transmit power, but also on the relative difference in sensitivity between the sensor and the PU.

Co-authored papers not presented in this thesis

Moussakhani B., Flåm J.T., Støa S., Balasingham I. and Ramstad, T. ***"On Localisation Accuracy inside the Human Abdomen Region"***, Journal on Wireless Sensor Systems, The Institution of Engineering and Technology, Volume: 2 , Issue: 1, 2012.
Abstract: In this work, localisation of a source within an absorbing medium is considered. By an absorbing medium, the authors mean an environment where the signal power decays exponentially with distance. The authors assume that the source is heard by nearby sensors when transmitting and its position shall be estimated based on the received signal strength (RSS) by these sensors. Under these assumptions, the focus is to determine the Cramer-Rao lower bound (CRLB). Thus, the goal is to derive the theoretical performance limit for an optimal estimator, and to study the feasibility of RSS-based localisation in an absorbing environment and specifically in human abdominal region. The authors demonstrate that the CRLB greatly depends on the shadowing conditions, and also on the relative positions of the sensors and the source. Although the obtained results are quite general, the motivating application is localisation of capsule endoscope in human abdominal region. The authors find that the RSS-based method can reach the needed accuracy for localising a capsule endoscope.

Moussakhani B., Ramstad T., Flåm, J.T. and Balasingham I. ***"On Localizing a Capsule Endoscope using Magnetic Sensors"***, 2012 Annual International Conference of the IEEE on Engineering in Medicine and Biology Society (EMBC).
Abstract: In this work, localizing a capsule endoscope within the gastrointestinal tract is addressed. It is assumed that the capsule is equipped with a magnet, and that a magnetic sensor network measures the flux from this magnet. We assume no prior knowledge on the source location, and that the measurements collected by the sensors are corrupted

by thermal Gaussian noise only. Under these assumptions, we focus on determining the Cramer-Rao Lower Bound (CRLB) for the location of the endoscope. Thus, we are not studying specific estimators, but rather the theoretical performance of an optimal one. It is demonstrated that the CRLB is a function of the distance and angle between the sensor network and the magnet. By studying the CRLB with respect to different sensor array constellations, we are able to indicate favorable constellations.

Moussakhani B., Flåm J.T., Ramstad T. and Balasingham I. ***"On the CRLB for Source Localization in a Lossy Environment"*** 2011 IEEE 12th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC).
Abstract: In this work, localization of a source within a lossy medium is considered. By a lossy medium we mean an environment where the signal power decays exponentially with distance. We assume no prior knowledge on the source location, but that the source is heard by nearby sensors when transmitting. The source position shall be estimated based on the power received by these sensors. Under these assumptions, our focus is to determine the Cramer-Rao Lower Bound (CRLB). Thus, we are not studying specific estimators, but rather the (theoretical) performance of an optimal one. We demonstrate that the CRLB greatly depends on the shadowing conditions, and also on the relative positions of the sensors and the source. This spatial variability of the CRLB is used to discuss favorable positioning of the sensors.

Moussakhani B., Flåm J.T., Ramstad T. and Balasingham I. ***"On Additive Change Detection in a Kalman Filter Based Tracking Problem"***, Submitted to Elsevier Journal on Signal Processing, January, 2013.
Abstract: This work considers detecting an additive abrupt state change in a tracking process. It is assumed that the tracking is done by a Kalman Filter and that the abrupt change takes place after the steady-state behavior of the filter is reached. The effect of the additive change on the innovation process is expressed in closed form, and we show that the optimal detection method depends on the available information, contained in the change vector. We take a Bayesian perspective and show that prior knowledge on the nature of the change can be used to significantly improve the detection performance. Specifically, we show that the performance of such a detector coincides with that of a matched filter when the variance (uncertainty) of the change tends to zero, and it coincides with that of an energy detector when the variance tends to infinity.

Part II

Papers

Paper 1

Gaussian Mixture Modeling for Source Localization

John T. Flåm, Joakim Jaldén and Saikat Chatterjee[1]

1.1 Abstract

Exploiting prior knowledge, we use Bayesian estimation to localize a source heard by a fixed sensor network. The method has two main aspects: Firstly, the probability density function (PDF) of a *function of the source location* is approximated by a Gaussian mixture model (GMM). This approximation can theoretically be made arbitrarily accurate, and allows a closed form minimum mean square error (MMSE) estimator for that function. Secondly, the source location is retrieved by minimizing the Euclidean distance between the function and its MMSE estimate using a gradient method. Our method avoids the issues of a numerical MMSE estimator but shows comparable accuracy.

1.2 Introduction

We wish to localize a transmitting radio based on the received signal strength (RSS)[2] of a known signature signal at several sensors in known positions. If the transmitted signal is sufficiently wideband, a log-normal shadowing model can be assumed. Then, the received

[1]John T. Flåm is with the Department of Electronics and Telecommunications, NTNU-Norwegian University of Science and Technology, Trondheim, Norway. Email: flam@iet.ntnu.no. Joakim Jaldén and Saikat Chatterjee are with the School of Electrical Engineering, KTH-Royal Institute of Technology, Sweden. Emails: joakim.jalden@ee.kth.se, sach@kth.se.

[2]There are other techniques for localization, such as time difference of arrival (TDOA), or triangulation by directions of arrival (DOA), but these techniques generally require more complex receivers.

power at sensor i is commonly modeled as

$$r_i = c_0 - 10\gamma \log\left(\|\mathbf{z} - \mathbf{q}_i\| / d_0\right) + s_i,$$

where all terms of the equation are in units of dB. Here, r_i is the power received by sensor i, c_0 is the average received power at a reference distance d_0 away from the source, $\|\mathbf{z} - \mathbf{q}_i\|$ is the euclidean distance from the source in unknown position $\mathbf{z} \in \mathbb{R}^D$ to sensor i in known position $\mathbf{q}_i \in \mathbb{R}^D$ (with $D = 2$ or 3), γ is the path loss exponent and $s_i \sim \mathcal{N}(0, \sigma_i^2)$ is a Gaussian random variable (RV) representing the shadowing between the source and sensor i. We assume that $c_0 = 0$ dB and $d_0 = 1$, without loss of generality, and that both γ and σ_i^2 are known. The received power then simplifies to

$$r_i = -10\gamma \log\left(\|\mathbf{z} - \mathbf{q}_i\|\right) + s_i = m_i(\mathbf{z}) + s_i. \tag{1.1}$$

With M sensors in the network, equation (1.1) can be written in vector form

$$\mathbf{r} = \mathbf{m}(\mathbf{z}) + \mathbf{s}, \tag{1.2}$$

where $\mathbf{m}(\mathbf{z})$ is the vector with $m_i(\mathbf{z})$ as its i-th component, $\mathbf{s} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$ and $\mathbf{C}$ is generally non-diagonal [42].

If the sensors and the source do not move, then neither $\mathbf{m}(\mathbf{z})$ nor the realization of $\mathbf{s}$ change, hence $\mathbf{r}$ is a constant vector[3]. Then, the problem is to estimate the source position $\mathbf{z} \in \mathbb{R}^D$ based on a single observation of $\mathbf{r} \in \mathbb{R}^M$, where typically $M > D$.

This problem is one on which there exists much related work. A good introduction to the general localization problem and different localization techniques can be found in [43], and the references therein. Particularly relevant are works on RSS-based localization with a Bayesian approach. In [44], a uniform source position density is assumed, and the resulting numerical MMSE estimator[4] is investigated. The work in [45] assumes motion on the source, and Bayesian tracking algorithms are investigated. Location estimation based on so-called fingerprinting is investigated in [46] .

We take a non-numerical, Bayesian approach to solving the described problem. Specifically, we approximate the PDF of $\mathbf{m}(\mathbf{z})$ with a Gaussian mixture model (GMM). This approximation allows a two stage estimation technique: Firstly, a closed form MMSE estimator, $\hat{\mathbf{m}}(\mathbf{z})$, for $\mathbf{m}(\mathbf{z})$ can computed. Secondly, the source position, $\mathbf{z}$, can be retrieved by minimizing the Euclidean distance between $\mathbf{m}(\mathbf{z})$ and $\hat{\mathbf{m}}(\mathbf{z})$ using a gradient search method. This estimation technique generally requires fewer computations than a numerical MMSE estimator for $\mathbf{z}$, but shows comparable accuracy. To the best of our knowledge, it has not been investigated in the context of localization before. Notation: we use $f(x)$ to denote the PDF of x, and, for simplicity, we mostly write $\mathbf{m}$ instead of $\mathbf{m}(\mathbf{z})$.

[3] Such a stationary scenario may be justified if the source is nomadic (e.g. moves to a location, switches on for a relatively long active session, then switches off and moves again).

[4] 'Numerical MMSE estimator' is short for an MMSE estimator based on numerical integration.

1.3 MMSE Estimation

This section first presents and justifies the GMM approximation of $f(\mathbf{m})$. Then we present a closed form MMSE estimator for $\mathbf{m}$. Finally, we argue that the parameters of the GMM approximation of $f(\mathbf{m})$ should be derived from a GMM approximation of $f(\mathbf{z})$.

1.3.1 GMM approximation

For arbitrary prior distributions on $\mathbf{m}$ a closed form MMSE estimator can generally not be found[5]. However, if the PDF of $\mathbf{m}$ is approximated by a K component GMM such that

$$f(\mathbf{m}) \approx \sum_{k=1}^{K} p(k) f(\mathbf{m}; \mathbf{u}_k, \mathbf{D}_k), \tag{1.3}$$

where $f(\mathbf{m}; \mathbf{u}_k, \mathbf{D}_k)$ is a Gaussian PDF with mean $\mathbf{u}_k$, covariance $\mathbf{D}_k$ and probability $p(k)$, a closed form MMSE estimator for $\mathbf{m}$ does exist. The approximation in equation (1.3) is not only mathematically convenient, it is also justifiable: Firstly, *any* PDF may be arbitrarily accurately approximated by a GMM, if the number of components in the model is large, and the mean and covariance of each component are properly chosen [47]. Secondly, in some scenarios, users tend to cluster around certain hot spots within a geographical area [48]. In that case, the PDF of $\mathbf{z}$ is multi modal, and consequently the PDF of $\mathbf{m}(\mathbf{z})$ will also be multi modal. Approximating it by a GMM is therefore reasonable.

1.3.2 MMSE estimator

For the model described by equation (1.2), and if $f(\mathbf{m})$ is approximated by a GMM of K components as in equation (1.3), then it can be shown, e.g. using [39], that the MMSE estimator for $\mathbf{m}(\mathbf{z})$ is

$$\hat{\mathbf{m}} = \frac{\sum_{k=1}^{K} v_k(\mathbf{r}) \mathbf{x}_k(\mathbf{r})}{\sum_{k=1}^{K} v_k(\mathbf{r})}. \tag{1.4}$$

Here

$$v_k(\mathbf{r}) = \frac{p(k) e^{\frac{1}{2}\left[(\mathbf{u}_k^T \mathbf{D}_k^{-1} + \mathbf{r}^T \mathbf{C}^{-1})(\mathbf{C}^{-1} + \mathbf{D}_k^{-1})^{-1}(\mathbf{D}_k^{-1}\mathbf{u}_k + \mathbf{C}^{-1}\mathbf{r})\right]}}{e^{\frac{1}{2}\mathbf{u}_k^T \mathbf{D}_k^{-1} \mathbf{u}_k} \left|(\mathbf{C} + \mathbf{D}_k)\right|^{1/2}} \tag{1.5}$$

and

$$\mathbf{x}_k(\mathbf{r}) = (\mathbf{C}^{-1} + \mathbf{D}_k^{-1})^{-1}(\mathbf{D}_k^{-1}\mathbf{u}_k + \mathbf{C}^{-1}\mathbf{r}).$$

[5] Recall that $\mathbf{m}$ is a function of a random argument, hence it is random and has a prior.

In the special case when $f(\mathbf{m})$ is approximated by a *single* Gaussian PDF, $\hat{\mathbf{m}}$ turns into the well known MMSE estimator resulting from the Bayesian linear model; see [41]. Assuming that all terms on the right hand side of equation (1.4) are known, one must retrieve a plausible $\mathbf{z}$ from $\hat{\mathbf{m}}$, to get an estimate for the source position. We address that issue in Section 1.4.

All matrix inversions in $v_k(\mathbf{r})$ and $\mathbf{x}_k(\mathbf{r})$ can be computed off line. Only the terms that involve the observation $\mathbf{r}$ must be computed on line. For $v_k(\mathbf{r})$, these include two inner products and a quadratic form. For $\mathbf{x}_k(\mathbf{r})$, it involves a matrix-vector product. Therefore, the total number of multiplications and additions needed to compute $\hat{\mathbf{m}}$ is $O(M^2K)$.

1.3.3 Determining the parameters of the GMM

The parameters $\mathbf{u}_k$, $\mathbf{D}_k$ and $p(k)$ in equation (1.3) are determined by means of training. For this purpose, we use the expectation maximization (EM) algorithm, the literature on which is rich [41], [49], [50]. Briefly, the EM algorithm relies on a sequence of independent observations drawn from the distribution we wish to approximate, and some initial estimate of the parameters. The observations are used to optimize the parameters iteratively until they converge to a local maximum of the likelihood function. This may generally require many iterations, and therefore $f(\mathbf{m})$ is best suited for GMM approximation if it is stationary. In this paper we assume that the spatial distribution of the source is stationary.

The number of components in the model, K, does not result from the algorithm. It must be chosen in advance by the designer and represents a trade off: a GMM of K components, describing an M dimensional RV, generally has $O(M^2K)$ parameters that must be estimated. Thus, increasing K leads to larger variance in the estimated parameters when the training sequence is finite. In our simple simulation scenarios, the training data form distinct clusters, and we simply choose K equal to the number of clusters. The resulting localization accuracy indicates that this works rather well, and that, in practice, the GMM approximation may be satisfactory even when K is limited. Generally, when distinct clusters are less apparent, there are several strategies on choosing K, see e.g. [51].

Assume now that a finite training sequence of $\mathbf{z}$'s (realizations of source positions) is available, and we have computed the corresponding sequence of $\mathbf{m}(\mathbf{z})$'s. If we could choose between approximating $f(\mathbf{m})$ or $f(\mathbf{z})$ by a GMM of K components, the former involves estimating a much larger number of parameters, whenever $M > D$. Therefore, one should expect that the EM algorithm approximates $f(\mathbf{z})$ more accurately than $f(\mathbf{m})$, at least on average. We shall exploit this, and approximate $f(\mathbf{m})$ *via* a GMM approximation of $f(\mathbf{z})$, as follows. Firstly, the PDF of $\mathbf{z}$ is approximated by a GMM

$$f(\mathbf{z}) \approx \sum_{k=1}^{K} p(k) f(\mathbf{z}; \mathbf{z}_k, \mathbf{S}_k), \tag{1.6}$$

where the means $\mathbf{z}_k$, covariances $\mathbf{S}_k$ and probabilities $p(k)$ are parameters resulting from

the EM algorithm with random initial values. Note that the accuracy of this approximation depends on K and the estimated parameters - it is independent of the number of sensors, M. Now, given that $\mathbf{z}$ originates from component k of the GMM in (1.6), the first order approximation of $\mathbf{m}(\mathbf{z})$ is

$$\mathbf{m}(\mathbf{z})|k \approx \mathbf{m}(\mathbf{z}_k) + \nabla\mathbf{m}(\mathbf{z}_k)(\mathbf{z} - \mathbf{z}_k), \tag{1.7}$$

where $\nabla\mathbf{m}(\mathbf{z}_k)$ denotes the Jacobian. Equation (1.7) is an affine transform of a Gaussian random variable. The result is Gaussian, even when the transform expands from low to high dimension [37], as in our case. Thus, $\mathbf{m}(\mathbf{z})|k$ is Gaussian with mean

$$\mathbf{u}_k = E\left\{\mathbf{m}(\mathbf{z})|k\right\} \approx \mathbf{m}(\mathbf{z}_k), \tag{1.8}$$

and covariance

$$\mathbf{D}_k \approx \nabla\mathbf{m}(\mathbf{z}_k)\mathbf{S}_k\nabla\mathbf{m}(\mathbf{z}_k)^T.$$

$\mathbf{D}_k$ is rank deficient, whenever $M > D$, because $\nabla\mathbf{m}(\mathbf{z}_k)$ is $M \times D$ and $\mathbf{S}_k$ is $D \times D$. This is a problem if we are interested in a PDF for $\mathbf{m}(\mathbf{z})|k$. An engineering solution, is to allow a small Gaussian noise term to the linear approximation, such that

$$\mathbf{m}(\mathbf{z})|k \approx \mathbf{m}(\mathbf{z}_k) + \nabla\mathbf{m}(\mathbf{z}_k)(\mathbf{z} - \mathbf{z}_k) + \mathbf{w},$$

where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma_w^2\mathbf{I})$. In this case we get

$$\mathbf{D}_k \approx \nabla\mathbf{m}(\mathbf{z}_k)\mathbf{S}_k\nabla\mathbf{m}(\mathbf{z}_k)^T + \sigma_w^2\mathbf{I}, \tag{1.9}$$

which is of full rank[6]. Thus $\mathbf{m}(\mathbf{z})|k$ is approximately Gaussian

$$f(\mathbf{m}(\mathbf{z})|k) \approx f(\mathbf{m}; \mathbf{u}_k, \mathbf{D}_k),$$

and the marginal PDF for $\mathbf{m}(\mathbf{z})$ is approximately a GMM

$$f(\mathbf{m}(\mathbf{z})) \approx \sum_{k=1}^{K} p(k) f(\mathbf{m}; \mathbf{u}_k, \mathbf{D}_k). \tag{1.10}$$

as in equation (1.3). The benefit of starting with equation (1.6), and linearizing $\mathbf{m}(\mathbf{z})$, is that we may now see equations (1.8) and (1.9) as reasonable initial parameters which improve the approximation in equation (1.10) when using the EM algorithm. In turn, this produces a better estimator, as we shall see in section 1.5.

[6]Our choice of σ_w^2 does of course affect the estimator, and could be a topic of investigation. In this paper, we simply choose a small value for σ_w^2 which does not introduce numerical instability when $\mathbf{D}_k$ is inverted.

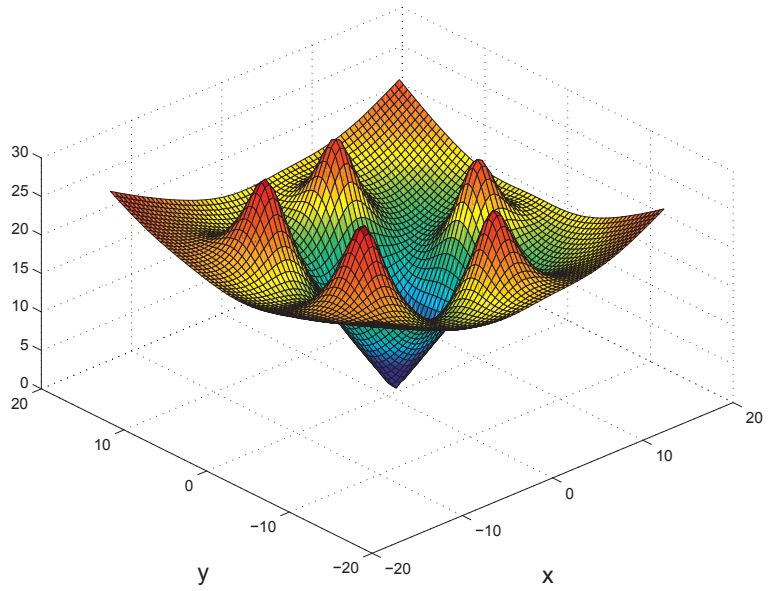

Figure 1.1: Non-convex Euclidean distance function sampled on a rectangular grid.

1.4 Retrieving the source position

Having computed $\hat{\mathbf{m}}$ (the estimate of $\mathbf{m}(\mathbf{z})$) we must retrieve a plausible source position $\mathbf{z}$. In this context, we assume that the function $\mathbf{m}(\mathbf{z})$ is one-to-one: conditions for this can be found in [52], and they are not difficult to satisfy in practice. Due to the GMM approximation and the shadowing, we cannot expect to find a $\mathbf{z}$ which satisfies $\hat{\mathbf{m}} = \mathbf{m}(\mathbf{z})$, even when $\mathbf{m}(\mathbf{z})$ is one-to-one. Instead, we must find a $\mathbf{z}$ which is satisfactory in some sense. One solution is to use that $\mathbf{z}$ which minimizes the Euclidean distance between $\mathbf{m}(\mathbf{z})$ and $\hat{\mathbf{m}}$. This distance is

$$d(\mathbf{z}|\hat{\mathbf{m}}) = \sqrt{(\hat{\mathbf{m}} - \mathbf{m}(\mathbf{z}))^T(\hat{\mathbf{m}} - \mathbf{m}(\mathbf{z}))}. \tag{1.11}$$

Figure 1.1 shows an example realization of $d(\mathbf{z}|\hat{\mathbf{m}})$; it is clearly non-convex. Here, the source has a constant and known height (1.5 m), and its true xy-coordinates are close to the origin. There are five sensors placed on a circle around the origin in the xy-plane. The saddle points are local minima of $d(\mathbf{z}|\hat{\mathbf{m}})$ where a gradient method theoretically may stop, but this is very unlikely. An analogy explains this informally: If gravity works on a ball placed randomly on the face of $d(\mathbf{z}|\hat{\mathbf{m}})$, it is unlikely to come to rest at one of the saddle points - in most cases the ball will settle at the lowest point. The same happens for a randomly started gradient method. We have tested this also for random sensor configurations, and it does not seem to change the argument. We will refer to the joint procedure of first computing $\hat{\mathbf{m}}$ and then minimizing equation (1.11) using a gradient method as a Euclidean estimator.

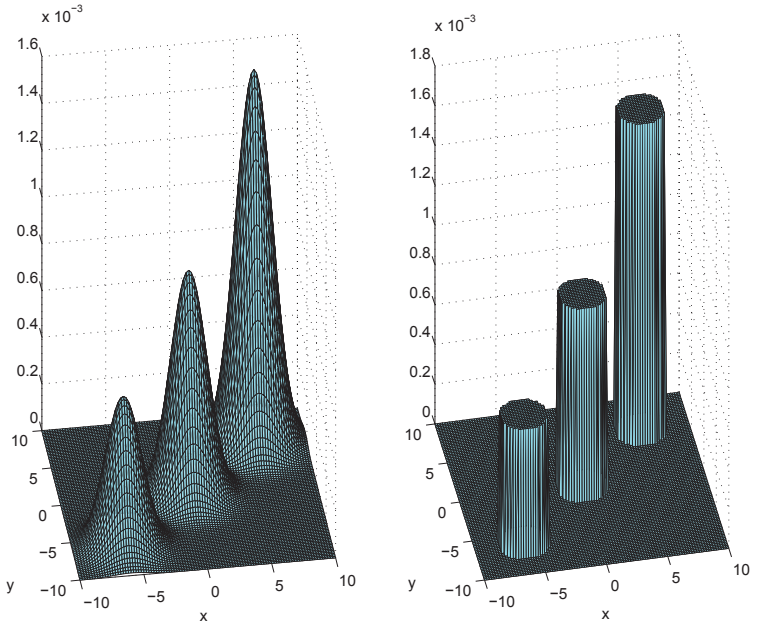

Figure 1.2: Left: Truncated GMM as source position density. Right: Cylinder shaped source position density.

1.5 Simulations

The performance of the Euclidean estimator is tested through Monte Carlo simulations. Since we wish to compare its accuracy with a numerical MMSE estimator, we simplify by assuming that the height of the source is constant and known (1.5 m). The sensors are placed in the xy-plane, on a circle with radius 10m around the origin, and we introduce spatially correlated shadowing using the empirical model presented in [42]:

$$C_{ij} = E[s_i s_j] = \sigma^2 e^{-d_{ij}/X_c}, \tag{1.12}$$

where d_{ij} is the distance between sensor i and sensor j, and X_c is the correlation distance. We set $\sigma^2 = 6$ and $X_c = 10$m.

Arguably, it is most interesting to see how our estimator deals with non Gaussian source distributions. To this end, we assume that the source transmits from a rectangular area, and that its spatial distribution within this area is a truncated GMM, shown in the left part of Figure 1.2. 3 This density has three hot spots, and since it has limited support it is *not* a GMM. We approximate it by a GMM with three components based on 25000 independent training samples. As initial estimates, we assume uniform component probabilities, three random sample positions as the means, and diagonal covariance matrices. Based on the GMM approximation for $f(\mathbf{z})$, we follow the steps in Section 1.3.3 (using

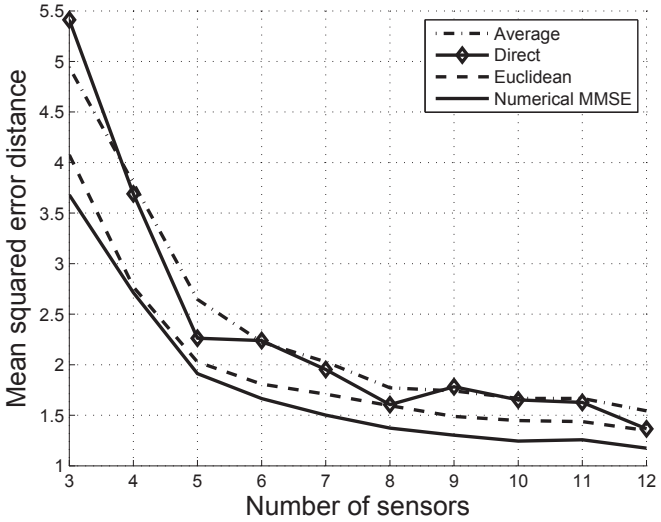

Figure 1.3: Mean squared error between estimates and the true source position versus number of receiving sensors. The source has a truncated GMM shown in left part of Figure 1.2.

$\sigma_w^2 = 0.01$) to obtain a GMM approximation for $f(\mathbf{m}(\mathbf{z}))$. The mean square error (MSE) of the resulting estimator, denoted by 'Euclidean', is shown in Figure 1.3, for an increasing number of sensors. In Figure 1.3, 'Direct' denotes the MSE of an estimator obtained by skipping the GMM approximation for $f(\mathbf{z})$, and running the EM algorithm directly to find a GMM approximation for $f(\mathbf{m}(\mathbf{z}))$. Its fluctuating MSE curve is due to the random initializations of the EM algorithm: each time a sensor is added to the network, the prior $f(\mathbf{m})$ is approximated by a GMM with randomly chosen initial parameter sets. The MSE fluctuations reflect fortunate and unfortunate approximations of the prior. The MSE curve can be smoothed by averaging the performance over multiple 'Direct' estimators (all based on different initial parameters, and hence different prior approximations). 'Average' denotes the MSE of 30 such estimators. The MSE of the 'Euclidean' estimator, on the other hand, does not fluctuate in the same way. Each time a new sensor is added to the network, the initial parameters of the GMM approximation for $f(\mathbf{m})$ are chosen, not randomly, but according to equations (1.8) and (1.9). The MSE of the 'Euclidean' estimator is clearly smaller than that of the 'Direct' estimator, and quite close to the numerical MMSE estimator. It underlines the importance of good initial parameters and that the GMM approximation of $f(\mathbf{m})$ should go via a GMM approximation of $f(\mathbf{z})$.

For the second performance test, we repeat the above experiment with a more distinctly non-Gaussian source distribution, shown in the right part of Figure 1.2. Again, we approximate this source distribution by a GMM of three components. The performance of the estimators are shown in Figure 1.4. Note that the 'Euclidean' and the numerical

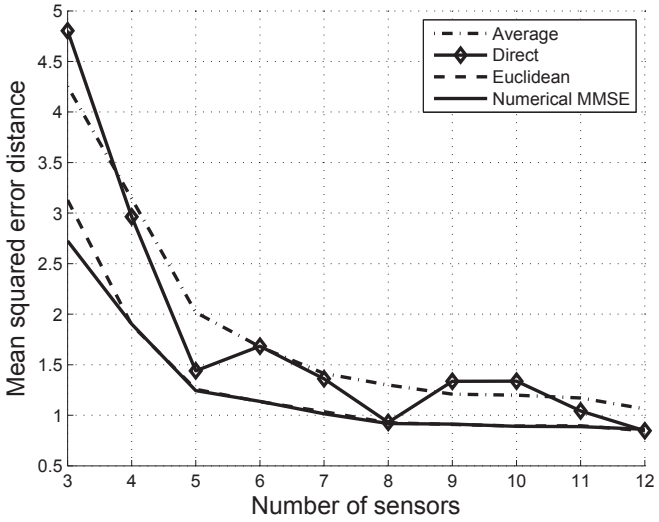

Figure 1.4: Mean squared error between estimates and the true source position versus number of receiving sensors. The source has a cylindrical density shown in right part of Figure 1.2.

MMSE estimator have almost identical MSEs, while the 'Direct' estimator shows highly varying degree of accuracy.

Disregarding any training, the complexity of these estimators is approximately the following: For the 'Euclidean' and 'Direct' estimators, computing $\hat{\mathbf{m}}$ dominates the complexity: $O(M^2K)$. For the numerical MMSE estimator, computed over U points: $O(M^2U)$. In our case, we used a rectangular grid of 1m resolution ($U = 21 \cdot 21 = 441$).

1.6 Conclusion

We have proposed a source localization technique consisting of two stages: MMSE estimation of a function of the source location followed by Euclidean distance minimization. The first stage requires training where the prior PDF is approximated by a GMM, but if the prior is stationary this training only has to be done once. Furthermore, compared to a numerical MMSE estimator, our approach need not worry about size, location and resolution of the grid of integration. In addition, it generally requires fewer computations. This estimator has been tested for two non Gaussian source distributions. In both cases the accuracy of the estimates is very close to that of a numerical MMSE estimator.

1.7 Acknowledgments

J. T. Flåm is supported by the Research Council of Norway under the NORDITE/VERDIKT programme, Project CROPS2 (Grant 181530/S10). Joakim Jaldén is supported by The Swedish Foundation for Strategic Research (grant ICA08-0046) and RICSNET. Saikat Chatterjee is supported partially by VINNOVA, RICSNET, and Eu FP7 FeedNetBack.

Paper 2

On MMSE estimation: A Linear Model under Gaussian Mixture Statistics

John T. Flåm, Saikat Chatterjee, Kimmo Kansanen, Torbjörn Ekman[1]

2.1 Abstract

In a Bayesian linear model, suppose observation $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$ stems from independent inputs $\mathbf{x}$ and $\mathbf{n}$ which are Gaussian mixture (GM) distributed. With known matrix $\mathbf{H}$, the minimum mean square error (MMSE) estimator for $\mathbf{x}$, has analytical form. However, its performance measure, the MMSE itself, has no such closed form. Because existing Bayesian MMSE bounds prove to have limited practical value under these settings, we instead seek *analytical* bounds for the MMSE, both upper and lower. This paper provides such bounds, and relates them to the signal-to-noise-ratio (SNR).

2.2 Introduction

In estimation theory, an important instance is the linear model

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

In this paper, $\mathbf{y}$ is a vector of observations, $\mathbf{H}$ is a known matrix, $\mathbf{x}$ is a vector to be estimated, and $\mathbf{n}$ is a vector of noise. We assume a Bayesian setting where the distributions on $\mathbf{x}$ and $\mathbf{n}$ are known a priori. Specifically, we posit $\mathbf{x}$ and $\mathbf{n}$ to be independent

[1]John T. Flåm, Kimmo Kansanen and Torbjörn Ekman are all with the Department of Electronics and Telecommunications, NTNU-Norwegian University of Science and Technology, Trondheim, Norway. Emails: flam.john@gmail.com, flam@iet.ntnu.no, kimmo.kansanen@iet.ntnu.no and torbjorn.ekman@iet.ntnu.no. Saikat Chatterjee is with the Communication Theory Lab, School of Electrical Engineering, KTH-Royal Institute of Technology, Sweden. Email: saikatchatt@gmail.com, sach@kth.se.

and Gaussian Mixture (GM) distributed. In order to estimate $\mathbf{x}$, we use the the minimum mean square error (MMSE) estimator

$$\mathbf{u}_{\mathbf{x}|\mathbf{y}} \triangleq E\left\{\mathbf{x}|\mathbf{y}\right\}, \tag{2.2}$$

and inquire about its performance, measured by the MMSE:

$$E\left\{\|\mathbf{x}-\mathbf{u}_{\mathbf{x}|\mathbf{y}}\|_2^2\right\}. \tag{2.3}$$

With GM inputs to (2.1), $\mathbf{u}_{\mathbf{x}|\mathbf{y}}$ becomes analytical, but in contrast to the familiar case of Gaussian inputs, the MMSE does not. Thus, while the optimal estimator is implementable, and used in practice, its performance is harder to assess. Moreover, in this setup, existing Bayesian bounds on the MMSE become no easier to estimate than the MMSE itself. Our objective is therefore to bound the MMSE *analytically*, both from above and below, and to relate these bounds to the signal-to-noise-ratio (SNR).

We find that the MMSE is *lower* bounded by the mean square error (MSE) of a 'genie-aided' estimator. This imaginary, non-implementable estimator knows precisely which source in the mixture is active, at any time, both for $\mathbf{x}$ and $\mathbf{n}$. It is therefore much better informed than the MMSE estimator. Yet, simulations indicate that the MMSE estimator is comparably accurate for SNRs above a quite modest threshold. The *upper* bound is provided by the MSE of the linear MMSE (LMMSE) estimator. In fact, we show that the LMMSE estimator becomes *the* MMSE estimator in two extreme cases: when the SNR is either zero or infinite. However, simulations indicate that the LMMSE estimator is nearly MSE optimal for a much larger range of SNR. The upper and lower bounds have closed form expressions, and are straightforward to calculate. Because they approach each other with increasing SNR, the MMSE can be determined rather accurately when the SNR is moderate to high.

The paper is organized as follows. Section 2.3 defines and motivates the input model and reviews parts of the related literature. Section 2.4 presents the main contribution: the upper and lower bounds of the MMSE. Section 2.5 proves the MMSE bounds, and analyzes them in the asymptotic cases of zero and infinite SNR. Section 2.6 simulates how the MMSE evolves, between its bounds, as a function of SNR. Section 2.7 concludes.

2.3 Input model and related work

We assume $\mathbf{x}$ and $\mathbf{n}$ in (2.1) to be independent and GM distributed:

$$\mathbf{x} \sim \sum_{k\in\mathcal{K}} p_k \mathcal{N}\left(\mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)}\right), \quad \mathbf{n} \sim \sum_{l\in\mathcal{L}} q_l \mathcal{N}\left(\mathbf{u}_{\mathbf{n}}^{(l)}, \mathbf{C}_{\mathbf{nn}}^{(l)}\right). \tag{2.4}$$

This notation should be read in the distributional sense. Thus, $\mathbf{x}$ results from a composite experiment. First, source $k \in \mathcal{K}$ is activated with probability $p_k \geq 0$, $\sum_{k\in\mathcal{K}} p_k = 1$. Second, that source generates a Gaussian signal with distribution law $\mathcal{N}(\mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)})$. The

set $\left\{p_k, \mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)}; k \in \mathcal{K}\right\}$ defines the GM *parameters* of $\mathbf{x}$. The noise $\mathbf{n}$ emerges in entirely similar, but independent manner. $\mathcal{K}$ and $\mathcal{L}$ are *finite* index sets. Their cardinalities determine the number of *components* in each mixture.

GM distributions can accommodate multi modality, asymmetry and heavy tails. In fact, with a sufficient number of components, and a judicious choice of parameters, a GM can approximate *any* distribution to desired accuracy [12], [13]. In practice, GM parameters are rarely given a priori. Most often they must be estimated; typically a non-trivial task [12, 53]. A common approach is to estimate GM parameters from training data. The expectation maximization (EM) algorithm [41, 49, 50] is well suited, and much used, for that purpose. Briefly, the algorithm relies on observations drawn from the distribution we wish to parametrize, and some initial estimate of the parameters. The observations are used to optimize the parameters, iteratively, until convergence to a local maximum of the likelihood function. Because the resulting GM parameters, and hence the GM distribution, depends on the initial estimates, the algorithm can alternatively be started from multiple initial estimates. This produces an ensemble of GM distributions, which can be averaged [54, 55]. For example, if there are $|\mathcal{M}|$ distributions in the ensemble, each with probability ξ_m, then $\mathbf{x}$ could be distributed as

$$\mathbf{x} \sim \sum_{m,k} \xi_m p_k \mathcal{N}\left(\mathbf{u}_{\mathbf{x}}^{(m,k)}, \mathbf{C}_{\mathbf{xx}}^{(m,k)}\right), \tag{2.5}$$

where $\mathbf{u}_{\mathbf{x}}^{(m,k)}$ and $\mathbf{C}_{\mathbf{xx}}^{(m,k)}$ are the k-th mean and k-th covariance, respectively, of GM distribution m. But apart from appropriate notational changes, (2.5) is a GM of the same format as (2.4). This emphasizes that notation (2.4) may also include uncertainty in the GM parameters. Our starting point is that (2.4) has resulted from such, or similar, model fitting. Above, we have compactly written $\sum_{m,k}$ for $\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}}$, a convention we will continue to use.

Model (2.1), with (2.4) as input, closely matches the image restoration problem described in e.g. [10]. There, $\mathbf{x}$ represents an image patch to be estimated and $\mathbf{H}$ is typically a non-invertible degradation operator. Since the problem usually is underdetermined, good patch models become important. Modeling $\mathbf{x}$ by a GM and the noise $\mathbf{n}$ as Gaussian has proven to give good results. However, unlike [10] which proposes a maximum a posteriori EM algorithm to estimate both the GM parameters and $\mathbf{x}$, we assume that the GM parameters already have been estimated, and concentrate on the signal $\mathbf{x}$. Moreover, we focus on MMSE estimation, the primary objective being to provide analytical bounds for the non-analytical MMSE.

Mixture distributions are common, particularly within two groups of the engineering literature. The first one focuses on state sequence estimation via noisy measurements, presuming statistics that fit the GM paradigm. Selected works include [13, 27–32]. These studies offer (approximate or exact) GM posterior state distributions, the mean always serving as the state estimate. For natural reasons, because the settings are not stationary, none of these works analyze the MMSE. The second group of studies uses GM distributions to simplify processing of speech, audio and images. Selected works include [15–26]. In

these, one or more variates are modeled by finite GMs. This is often sufficiently accurate, and allows good practical estimators. However, none of these works investigate bounds on the MMSE.

The emphasis on *analytical* bounds separates this work from those offering general lower bounds on the MMSE. Included there, are the Bayesian bounds of Cramer-Rao [33], Bobrovsky-Zakai [34], Bhattacharyya [33], Weiss-Weinstein [35], and Reuven-Messer [36]. These bounds hold for most types of joint densities $f(\mathbf{x}, \mathbf{y})$ of practical interest but they rarely acquire analytical form. Yet, for most $f(\mathbf{x}, \mathbf{y})$ they are much simpler to evaluate numerically than the MMSE. That feature makes them both attractive and useful; for interested readers, [56] proposes a general class of Bayesian lower bounds. Like the MMSE, the above mentioned bounds all become non-analytical when (2.4) is input to (2.1), but in contrast to the usual case, they do *not* become simpler to evaluate numerically than the MMSE. For this reason, the mentioned bounds have limited practical value to the current problem. Such bounds, of analytical sort, are useful because they can provide a simple way to assess estimator performance. To our knowledge, the literature offers no explicit analysis on that account. This motivates us to explore and present both upper and lower bounds.

2.4 Main result: Bounds on the MMSE

Proposition 7. *Let observation* $\mathbf{y}$ *originate from (2.1), where* $\mathbf{H}$ *is a known matrix, and* $\mathbf{x}$ *and* $\mathbf{n}$ *are independent GM variates defined by (2.4). Let* $\mathbf{u}_{\mathbf{x}|\mathbf{y}}$ *in (2.2) denote the MMSE estimator for* $\mathbf{x}$*. Its performance, measured by* $\epsilon^2 = E\left\{\|\mathbf{x} - \mathbf{u}_{\mathbf{x}|\mathbf{y}}\|_2^2\right\}$*, is bounded as follows:*

$$\begin{aligned}&\sum_{k,l} p_k q_l Tr\left(\mathbf{C}_{\mathbf{xx}}^{(k)} - \mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T\left(\mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T + \mathbf{C}_{\mathbf{nn}}^{(l)}\right)^{-1}\mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)}\right)\\ &\leq \epsilon^2 \leq Tr\left(\mathbf{C}_{\mathbf{xx}} - \mathbf{C}_{\mathbf{xx}}\mathbf{H}^T\left(\mathbf{H}\mathbf{C}_{\mathbf{xx}}\mathbf{H}^T + \mathbf{C}_{\mathbf{nn}}\right)^{-1}\mathbf{H}\mathbf{C}_{\mathbf{xx}}\right).\end{aligned}$$

Here $Tr(\cdot)$ *denotes the trace operator, and*

$$\mathbf{C}_{\mathbf{xx}} = \sum_k p_k\left(\mathbf{C}_{\mathbf{xx}}^{(k)} + \mathbf{u}_{\mathbf{x}}^{(k)}\mathbf{u}_{\mathbf{x}}^{(k)^T}\right) - \mathbf{u}_{\mathbf{x}}\mathbf{u}_{\mathbf{x}}^T, \quad \mathbf{u}_{\mathbf{x}} = \sum_k p_k \mathbf{u}_{\mathbf{x}}^{(k)}, \tag{2.6}$$

are the covariance and mean of $\mathbf{x}$*, respectively. Similarly*

$$\mathbf{C}_{\mathbf{nn}} = \sum_l q_l\left(\mathbf{C}_{\mathbf{nn}}^{(l)} + \mathbf{u}_{\mathbf{n}}^{(l)}\mathbf{u}_{\mathbf{n}}^{(l)^T}\right) - \mathbf{u}_{\mathbf{n}}\mathbf{u}_{\mathbf{n}}^T, \quad \mathbf{u}_{\mathbf{n}} = \sum_l q_l \mathbf{u}_{\mathbf{n}}^{(l)}, \tag{2.7}$$

are the covariance and mean of and $\mathbf{n}$*, respectively. Proof is given in the next section.*

2.5 Derivation of the MMSE bounds

Here, we first verify the non-analytical nature of the MMSE. Then we prove the bounds in Proposition 7. The MMSE is, by definition,

$$\begin{aligned}\epsilon^2 &\triangleq E\left\{\|\mathbf{x}-\mathbf{u}_{\mathbf{x}|\mathbf{y}}\|_2^2\right\}\\ &= \iint \left(\mathbf{x}-\mathbf{u}_{\mathbf{x}|\mathbf{y}}\right)^T\left(\mathbf{x}-\mathbf{u}_{\mathbf{x}|\mathbf{y}}\right) f(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y}\\ &= \iint \left(\mathbf{x}-\mathbf{u}_{\mathbf{x}|\mathbf{y}}\right)^T\left(\mathbf{x}-\mathbf{u}_{\mathbf{x}|\mathbf{y}}\right) f(\mathbf{x}|\mathbf{y})d\mathbf{x}f(\mathbf{y})d\mathbf{y}\\ &= \int \mathrm{Tr}\left(\mathbf{C}_{\mathbf{x}|\mathbf{y}}\right) f(\mathbf{y})d\mathbf{y} \geq 0. \end{aligned} \tag{2.8}$$

Under (2.1) and (2.4), it can be shown that $\mathbf{y}$ in (2.8) is a GM with $|\mathcal{K}|\cdot|\mathcal{L}|$ components. Its density has the form

$$f(\mathbf{y}) = \sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y}), \tag{2.9}$$

where $f^{(k,l)}(\mathbf{y})$ is a Gaussian density with mean and covariance

$$\mathbf{u}_{\mathbf{y}}^{(k,l)} = \mathbf{H}\mathbf{u}_{\mathbf{x}}^{(k)} + \mathbf{u}_{\mathbf{n}}^{(l)}, \qquad \mathbf{C}_{\mathbf{yy}}^{(k,l)} = \mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T + \mathbf{C}_{\mathbf{nn}}^{(l)}, \tag{2.10}$$

respectively. Furthermore, it can be shown that the posterior covariance matrix in (2.8) is

$$\mathbf{C}_{\mathbf{x}|\mathbf{y}} = \sum_{k,l} \alpha^{(k,l)}(\mathbf{y})\Big(\mathbf{C}_{\mathbf{x}|\mathbf{y}}^{(k,l)} + \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)}\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)^T}\Big) - \mathbf{u}_{\mathbf{x}|\mathbf{y}}\mathbf{u}_{\mathbf{x}|\mathbf{y}}^T, \tag{2.11}$$

where

$$\alpha^{(k,l)}(\mathbf{y}) = \frac{p_k q_l f^{(k,l)}(\mathbf{y})}{f(\mathbf{y})}, \tag{2.12}$$

$$\mathbf{C}_{\mathbf{x}|\mathbf{y}}^{(k,l)} = \mathbf{C}_{\mathbf{xx}}^{(k)} - \mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T\left(\mathbf{C}_{\mathbf{yy}}^{(k,l)}\right)^{-1}\mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)}, \tag{2.13}$$

$$\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)} = \mathbf{u}_{\mathbf{x}}^{(k)} + \mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T\left(\mathbf{C}_{\mathbf{yy}}^{(k,l)}\right)^{-1}\left(\mathbf{y}-\mathbf{u}_{\mathbf{y}}^{(k,l)}\right), \tag{2.14}$$

$$\mathbf{u}_{\mathbf{x}|\mathbf{y}} = \sum_{k,l} \alpha^{(k,l)}(\mathbf{y})\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)}. \tag{2.15}$$

Detailed derivations on (2.9)-(2.15) are not included here. The interested reader is instead referred to e.g. [15,19,39,40]. In (2.8), rather than dealing directly with ϵ^2, it is convenient to study the matrix

$$\begin{aligned}\mathbf{M} &= \int \mathbf{C}_{\mathbf{x}|\mathbf{y}} f(\mathbf{y})d\mathbf{y} =\\ &\sum_{k,l} p_k q_l \int\Big(\mathbf{C}_{\mathbf{x}|\mathbf{y}}^{(k,l)} + \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)}\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)^T} - \mathbf{u}_{\mathbf{x}|\mathbf{y}}\mathbf{u}_{\mathbf{x}|\mathbf{y}}^T\Big) f^{(k,l)}(\mathbf{y})d\mathbf{y}. \end{aligned} \tag{2.16}$$

Here, the last equality results by using (2.11) and (2.12). Observe that $\mathrm{Tr}(\mathbf{M}) = \epsilon^2$. Since $\mathbf{M}$ is of crucial interest, we inspect integral (2.16) term-by-term. The first term is

$$\mathbf{M}_1 = \sum_{k,l} p_k q_l \int \mathbf{C}_{\mathbf{x}|\mathbf{y}}^{(k,l)} f^{(k,l)}(\mathbf{y}) d\mathbf{y} = \sum_{k,l} p_k q_l \mathbf{C}_{\mathbf{x}|\mathbf{y}}^{(k,l)}, \tag{2.17}$$

where the last equality holds because $\mathbf{C}_{\mathbf{x}|\mathbf{y}}^{(k,l)}$ is not a function of $\mathbf{y}$, as can be seen from (2.13) and (2.10). The second term of (2.16) is

$$\mathbf{M}_2 = \sum_{k,l} p_k q_l \int \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)^T} f^{(k,l)}(\mathbf{y}) d\mathbf{y} \tag{2.18}$$

$$= \sum_{k,l} p_k q_l \left(\mathbf{u}_{\mathbf{x}}^{(k)} \mathbf{u}_{\mathbf{x}}^{(k)^T} + \mathbf{C}_{\mathbf{xx}}^{(k)} - \mathbf{C}_{\mathbf{x}|\mathbf{y}}^{(k,l)} \right). \tag{2.19}$$

The last equality derives using (2.14) and (2.13). The third term of (2.16) is

$$\mathbf{M}_3 = -\int \mathbf{u}_{\mathbf{x}|\mathbf{y}} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^T \sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y}) d\mathbf{y} \tag{2.20}$$

$$= -\sum_{k,l,r,s} p_k q_l p_r q_s \int \frac{f^{(k,l)}(\mathbf{y}) f^{(r,s)}(\mathbf{y}) \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T}}{\sum_{v,w} p_v q_w f^{(v,w)}(\mathbf{y})} d\mathbf{y}. \tag{2.21}$$

The last equality results by using (2.15), (2.12) and (2.9). As far as we can see, (2.21) cannot be solved due to the sum in the denominator of the integrand. The MMSE,

$$\epsilon^2 = \mathrm{Tr}(\mathbf{M}) = \mathrm{Tr}(\mathbf{M}_1) + \mathrm{Tr}(\mathbf{M}_2) + \mathrm{Tr}(\mathbf{M}_3) \geq 0, \tag{2.22}$$

can therefore not be determined exactly. Instead we turn to bounds.

It can be verified that the existing Bayesian bounds [33–36] all depend on integral expressions of the following form

$$\iint \frac{g(\mathbf{y}, \mathbf{x})}{f(\mathbf{y}, \mathbf{x})} d\mathbf{x}\, d\mathbf{y}. \tag{2.23}$$

With (2.4) as input to (2.1), it can be shown that $f(\mathbf{y}, \mathbf{x})$ is a GM with $|\mathcal{K}| \cdot |\mathcal{L}|$ components. The function $g(\mathbf{y}, \mathbf{x})$ depends on the bound in question, but (2.23) cannot be evaluated analytically for any of the above bounds. The main reason, is that the integrand of (2.23), like in (2.21), has a GM denominator which does not simplify by substitutions. In addition, for any of these bounds, (2.23) cannot be construed as an expectation. The latter complicates Monte Carlo approaches, and numerical estimation of the mentioned bounds is therefore not straightforward. In contrast, $\mathbf{u}_{\mathbf{x}|\mathbf{y}}$ in (2.15) is analytical, and because GMs are easy to sample from, the MMSE *can* be estimated by Monte Carlo methods. Clearly, in practice, the MMSE should only be bounded by quantities that are considerably simpler to obtain. Otherwise, one would rather estimate the MMSE directly. Therefore, we

instead seek *analytical* bounds - both upper and lower. To that end, note from equations (2.17), (2.18) and (2.20), that

$$\mathrm{Tr}(\mathbf{M}_1) = \sum_{k,l} p_k q_l \mathrm{Tr}\left(\mathbf{C}_{\mathbf{x}|\mathbf{y}}^{(k,l)}\right) \geq 0, \tag{2.24}$$

$$\mathrm{Tr}(\mathbf{M}_2) = \sum_{k,l} p_k q_l \int \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)^T} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)} f^{(k,l)}(\mathbf{y}) d\mathbf{y} \geq 0, \tag{2.25}$$

$$\mathrm{Tr}(\mathbf{M}_3) = -\sum_{k,l} p_k q_l \int \mathbf{u}_{\mathbf{x}|\mathbf{y}}^T \mathbf{u}_{\mathbf{x}|\mathbf{y}} f^{(k,l)}(\mathbf{y}) d\mathbf{y} \leq 0. \tag{2.26}$$

These inequalities follow because $\mathbf{C}_{\mathbf{x}|\mathbf{y}}^{(k,l)}$ is a covariance matrix, and $\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)^T} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)}$ and $\mathbf{u}_{\mathbf{x}|\mathbf{y}}^T \mathbf{u}_{\mathbf{x}|\mathbf{y}}$ are inner products. Observe also that $\mathrm{Tr}(\mathbf{M}_2) + \mathrm{Tr}(\mathbf{M}_3)$ is always non-negative. This can be seen by applying (2.25) and (2.26):

$$\begin{aligned}
&\mathrm{Tr}(\mathbf{M}_2) + \mathrm{Tr}(\mathbf{M}_3) \\
&= \int \sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y}) \left(\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)^T} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)} - \mathbf{u}_{\mathbf{x}|\mathbf{y}}^T \mathbf{u}_{\mathbf{x}|\mathbf{y}} \right) d\mathbf{y} \\
&= \int \sum_{k,l} \alpha^{(k,l)}(\mathbf{y}) \left\| \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)} - \mathbf{u}_{\mathbf{x}|\mathbf{y}} \right\|_2^2 f(\mathbf{y}) d\mathbf{y} \geq 0.
\end{aligned} \tag{2.27}$$

The last equality is obtained from (2.12) and (2.15). Because $\mathrm{Tr}(\mathbf{M}_3)$ is non-analytical, we shall inquire whether $\mathrm{Tr}(\mathbf{M}_1)$ and $\mathrm{Tr}(\mathbf{M}_2)$ may provide useful bounds.

2.5.1 Proving the lower bound for the MMSE

Inequalities (2.22) and (2.24)-(2.27), guarantee that

$$\mathrm{Tr}(\mathbf{M}_1) \leq \epsilon^2. \tag{2.28}$$

Now we ask whether $\mathrm{Tr}(\mathbf{M}_2)$ also could serve as a lower bound. If so, then $\mathrm{Tr}(\mathbf{M}_1) + \mathrm{Tr}(\mathbf{M}_3) \geq 0$ must be satisfied. If not, then $\mathrm{Tr}(\mathbf{M}_2)$ would be an *upper* bound for the MMSE. It can be verified (numerically) that this inequality may hold for certain GM parameters, but not for all. A larger obstacle, however, is that $\mathrm{Tr}(\mathbf{M}_1) + \mathrm{Tr}(\mathbf{M}_3)$ is almost as demanding to compute as the MMSE itself. Therefore, among $\mathrm{Tr}(\mathbf{M}_2)$ and $\mathrm{Tr}(\mathbf{M}_1)$, the latter is the only practical lower bound.

An alternative argument provides intuition for the lower bound in (2.28). Imagine that side information is available, such that for each observation $\mathbf{y}$, a genie tells precisely which Gaussian components k and l came into play. Then, for each $\mathbf{y}$, we face a Gaussian signal in Gaussian noise. Because of the genie, we no longer use the observation $\mathbf{y}$ to determine the weights in (2.12). Instead, $\alpha^{(k,l)}(\mathbf{y}) = 1$ for the correct index pair (k, l), whereas $\alpha^{(k',l')}(\mathbf{y}) = 0$ for all other index pairs. Consequently, the estimator (2.15) pinpoints the

correct posterior component mean, $\mathbf{u}_{\mathbf{x}|\mathbf{y}} = \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)}$, as the estimate for $\mathbf{x}$. From (2.27), it is then clear that $\text{Tr}(\mathbf{M}_2) + \text{Tr}(\mathbf{M}_3) = 0$, and the MSE of the genie-aided estimator is therefore $\epsilon_{genie}^2 = \text{Tr}(\mathbf{M}_1)$. One can interpret this estimator as a perfect decision device, followed by a decision dependent Gaussian signal and Gaussian noise MMSE estimator. It is not realizable in practice, but it provides a performance benchmark: Any practical estimator, has MSE of at least $\text{Tr}(\mathbf{M}_1)$. Therefore $\text{Tr}(\mathbf{M}_1)$ is a lower bound, also for the MMSE. Section 2.5.3, shows that this bound coincides with the MMSE at infinite SNR, but simulations indicate that, in practice, the lower bound may be tight already from moderate SNR levels. By appropriate substitutions, using (2.17), (2.13) and (2.10) one obtains the lower bound of Proposition 7.

2.5.2 Proving the upper bound for the MMSE

From inequalities (2.22) and (2.24)-(2.26), it is straightforward to conclude that

$$\epsilon^2 \leq \text{Tr}(\mathbf{M}_1) + \text{Tr}(\mathbf{M}_2). \tag{2.29}$$

However, this upper bound is not really informative. In order to see this, apply (2.17) and (2.19) to have

$$\begin{aligned}\text{Tr}(\mathbf{M}_1) + \text{Tr}(\mathbf{M}_2) &= \sum_k p_k \left(\text{Tr}\left(\mathbf{C}_{\mathbf{xx}}^{(k)}\right) + \mathbf{u}_{\mathbf{x}}^{(k)^T} \mathbf{u}_{\mathbf{x}}^{(k)} \right) \\ &\geq \sum_k p_k \left(\text{Tr}\left(\mathbf{C}_{\mathbf{xx}}^{(k)}\right) + \mathbf{u}_{\mathbf{x}}^{(k)^T} \mathbf{u}_{\mathbf{x}}^{(k)} \right) - \mathbf{u}_x^T \mathbf{u}_x = \text{Tr}\left(\mathbf{C}_{\mathbf{xx}}\right).\end{aligned}$$

The last equality follows from (2.6). $\text{Tr}\left(\mathbf{C}_{\mathbf{xx}}\right) = E\left\{\|\mathbf{x} - \mathbf{u}_{\mathbf{x}}\|_2^2\right\}$ is the MSE of the highly suboptimal estimator $\hat{\mathbf{x}} = \mathbf{u}_{\mathbf{x}}$. This estimator, which completely disregards the observation $\mathbf{y}$, has an MSE which is even smaller than $\text{Tr}(\mathbf{M}_1) + \text{Tr}(\mathbf{M}_2)$!

Because $\text{Tr}(\mathbf{M}_2)$, $\text{Tr}(\mathbf{M}_1) + \text{Tr}(\mathbf{M}_2)$ and $\text{Tr}\left(\mathbf{C}_{\mathbf{xx}}\right)$ all are inadequate as upper bounds, we invoke the LMMSE estimator[2]. This estimator is given by (see e.g. Theorem 12.1 of [41])

$$\hat{\mathbf{x}} = \mathbf{u}_{\mathbf{x}} + \mathbf{C}_{\mathbf{xx}}\mathbf{H}^T \left(\mathbf{H}\mathbf{C}_{\mathbf{xx}}\mathbf{H}^T + \mathbf{C}_{\mathbf{nn}}\right)^{-1} (\mathbf{y} - \mathbf{H}\mathbf{u}_{\mathbf{x}} - \mathbf{u}_{\mathbf{n}}), \tag{2.30}$$

where $\mathbf{C}_{\mathbf{xx}}$, $\mathbf{u}_{\mathbf{x}}$, $\mathbf{C}_{\mathbf{nn}}$ and $\mathbf{u}_{\mathbf{n}}$ are given in (2.6) and (2.7). Its error, $\mathbf{x} - \hat{\mathbf{x}}$, has zero mean and covariance matrix

$$\mathbf{C}_{\mathbf{xx}} - \mathbf{C}_{\mathbf{xx}}\mathbf{H}^T \left(\mathbf{H}\mathbf{C}_{\mathbf{xx}}\mathbf{H}^T + \mathbf{C}_{\mathbf{nn}}\right)^{-1} \mathbf{H}\mathbf{C}_{\mathbf{xx}}. \tag{2.31}$$

The (closed form) MSE of the LMMSE estimator is

$$\begin{aligned}\epsilon_L^2 &= E\left\{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2\right\} \\ &= \text{Tr}\left(\mathbf{C}_{\mathbf{xx}} - \mathbf{C}_{\mathbf{xx}}\mathbf{H}^T \left(\mathbf{H}\mathbf{C}_{\mathbf{xx}}\mathbf{H}^T + \mathbf{C}_{\mathbf{nn}}\right)^{-1} \mathbf{H}\mathbf{C}_{\mathbf{xx}}\right) \\ &= \text{Tr}\left(\mathbf{C}_{\mathbf{xx}}\right) - \sum_j \mathbf{g}_j^T \left(\mathbf{H}\mathbf{C}_{\mathbf{xx}}\mathbf{H}^T + \mathbf{C}_{\mathbf{nn}}\right)^{-1} \mathbf{g}_j,\end{aligned} \tag{2.32}$$

[2] Among all estimators which are linear (affine) in the observations, the LMMSE estimator obtains the smallest MSE.

where, in the last equality, $\mathbf{g}_j$ is the j-th column of $\mathbf{HC_{xx}}$. Recall that $\mathbf{C_{xx}}$ and $\mathbf{C_{nn}}$ are both positive definite and symmetric matrices. Thus, provided the LMMSE estimator in (2.30) exists, $\left(\mathbf{HC_{xx}H}^T + \mathbf{C_{nn}}\right)^{-1}$ has to be positive definite. This, and (2.32), shows that $\epsilon_L^2 \leq \mathrm{Tr}\left(\mathbf{C_{xx}}\right)$. Because no practical estimator has MSE smaller than the MMSE, by definition, we can safely conclude that

$$\epsilon^2 \leq \epsilon_L^2.$$

Note that ϵ_L^2 in (2.32) corresponds to the upper bound of Proposition 7. Clearly, if another estimator with a closed form MSE less than ϵ_L^2 exists, it can tighten this upper bound. For the present problem, we are not aware of any.

2.5.3 MMSE at zero and infinite SNR

Intuitively, one expects that the MMSE approaches its upper bound as

$$\mathrm{SNR} = E\left\{\|\mathbf{x}\|_2^2\right\} / E\left\{\|\mathbf{n}\|_2^2\right\}, \tag{2.33}$$

tends to zero. Similarly, we expect the MMSE to approach its lower bound as the SNR tends to infinity. We will demonstrate that this intuition is true for a simple but instructive example. Throughout, we assume $\mathbf{H}$ to be square and full rank. In addition, we assume the noise to be distributed as

$$\mathbf{n} \sim \sum_{l \in \mathcal{L}} q_l \mathcal{N}(a\mathbf{u}_{\mathbf{n}}^{(l)}, a^2\mathbf{C}_{\mathbf{nn}}^{(l)}), \tag{2.34}$$

where the scalar a can account for any SNR. Then (2.14) becomes

$$\mathbf{u}_{\mathbf{x|y}}^{(k,l)} = \mathbf{u}_{\mathbf{x}}^{(k)} + \mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T\left(\mathbf{HC}_{\mathbf{xx}}^{(k)}\mathbf{H}^T + a^2\mathbf{C}_{\mathbf{nn}}^{(l)}\right)^{-1}\left(\mathbf{y} - \mathbf{Hu}_{\mathbf{x}}^{(k)} - a\mathbf{u}_{\mathbf{n}}^{(l)}\right) \tag{2.35}$$

$$= \mathbf{u}_{\mathbf{x}}^{(k)} + \left(\left(\mathbf{C}_{\mathbf{xx}}^{(k)}\right)^{-1} + \mathbf{H}^T\frac{1}{a^2}\left(\mathbf{C}_{\mathbf{nn}}^{(l)}\right)^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^T\frac{1}{a^2}\left(\mathbf{C}_{\mathbf{nn}}^{(l)}\right)^{-1}\left(\mathbf{y} - \mathbf{Hu}_{\mathbf{x}}^{(k)} - a\mathbf{u}_{\mathbf{n}}^{(l)}\right), \tag{2.36}$$

where (2.36) is merely an equivalent but useful form (see e.g. equation (10.32) of [41]). Similarly, the LMMSE estimator in (2.30) becomes

$$\hat{\mathbf{x}} = \mathbf{u}_{\mathbf{x}} + \mathbf{C}_{\mathbf{xx}}\mathbf{H}^T\left(\mathbf{HC}_{\mathbf{xx}}\mathbf{H}^T + a^2\mathbf{C}_{\mathbf{nn}}\right)^{-1}\left(\mathbf{y} - \mathbf{Hu}_{\mathbf{x}} - a\mathbf{u}_{\mathbf{n}}\right) \tag{2.37}$$

$$= \mathbf{u}_{\mathbf{x}} + \left(\mathbf{C}_{\mathbf{xx}}^{-1} + \mathbf{H}^T\frac{\mathbf{C}_{\mathbf{nn}}^{-1}}{a^2}\mathbf{H}\right)^{-1}\mathbf{H}^T\frac{\mathbf{C}_{\mathbf{nn}}^{-1}}{a^2}\left(\mathbf{y} - \mathbf{Hu}_{\mathbf{x}} - a\mathbf{u}_{\mathbf{n}}\right), \tag{2.38}$$

where (2.38) is also an equivalent form. Given (2.34), then (2.13) becomes

$$\mathbf{C}_{\mathbf{x|y}}^{(k,l)} = \mathbf{C}_{\mathbf{xx}}^{(k)} - \mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T\left(\mathbf{HC}_{\mathbf{xx}}^{(k)}\mathbf{H}^T + a^2\mathbf{C}_{\mathbf{nn}}^{(l)}\right)^{-1}\mathbf{HC}_{\mathbf{xx}}^{(k)}. \tag{2.39}$$

Zero SNR

We drive the SNR to zero by $a \to \infty$. Then (2.36) reduces to $\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)} = \mathbf{u}_{\mathbf{x}}^{(k)}$. Thus, the MMSE estimate for $\mathbf{x}$ is

$$\mathbf{u}_{\mathbf{x}|\mathbf{y}} = \frac{\sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y}) \mathbf{u}_{\mathbf{x}}^{(k)}}{\sum_{r,s} p_r q_s f^{(r,s)}(\mathbf{y})} = \sum_k p_k \mathbf{u}_{\mathbf{x}}^{(k)} = \mathbf{u}_{\mathbf{x}}. \tag{2.40}$$

Here, the second equality holds because $f^{(k,l)}(\mathbf{y})$ is a function of the covariance matrix, $\mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T + a^2\mathbf{C}_{\mathbf{nn}}^{(l)}$, and when $a \to \infty$, $f^{(k,l)}(\mathbf{y})$ approaches a uniform distribution with infinite support. Hence, $f^{(k,l)}(\mathbf{y})$ approaches a constant which is independent of $\mathbf{y}$, k and l. Note from (2.40), that the MMSE estimator discards the data and only uses prior information. This is expected at zero SNR.

Now we compare with the LMMSE estimator. When $a \to \infty$, (2.38) reduces to $\hat{\mathbf{x}} = \mathbf{u}_{\mathbf{x}} = \sum_k p_k \mathbf{u}_{\mathbf{x}}^{(k)}$. But this is equal to (2.40), which is the MMSE estimate. Thus, the LMMSE estimator becomes *the* MMSE estimator, at zero SNR. As argued in Section 2.5.2 the MMSE becomes $\text{Tr}(\mathbf{C}_{\mathbf{xx}})$ in this case.

Infinite SNR

We drive the SNR to infinity by $a \to 0$. Then, because $\mathbf{H}$ is square and full rank, (2.35) reads $\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)} = \mathbf{H}^{-1}\mathbf{y}$. Using this in (2.15), we find that

$$\mathbf{u}_{\mathbf{x}|\mathbf{y}} = \mathbf{H}^{-1}\mathbf{y}. \tag{2.41}$$

From (2.27), it can then be seen that $\text{Tr}(\mathbf{M}_2) + \text{Tr}(\mathbf{M}_3) = 0$. Thus, at infinite SNR, the MMSE in (2.22) becomes $\text{Tr}(\mathbf{M}_1)$. Evaluating (2.39) when $a \to 0$, and plugging the result into (2.24), it is straight forward to verify that MMSE becomes zero at infinite SNR. Note from (2.41), that the estimator discards all prior knowledge and completely trusts the data. This is expected at infinitely high SNR.

To conclude this inquiry, we again compare with the LMMSE estimator. When $a \to 0$, (2.37) reduces to $\hat{\mathbf{x}} = \mathbf{H}^{-1}\mathbf{y}$. But this is the same as the MMSE estimator in (2.41). Thus, the LMMSE estimator becomes *the* MMSE estimator, also at infinite SNR.

In summary, at both zero and infinite SNR, the MMSE becomes analytical and equals the MSE of the LMMSE estimator. At infinite SNR, the MMSE also equals the MSE of the genie-aided estimator. Therefore, the LMMSE upper bound and the genie lower bound must coincide as the SNR tends to infinity.

2.6 Numerical Results

Here we simulate the MMSE, and calculate the bounds, all as functions of SNR. The MMSE can be estimated as the sample mean of $\left\|\mathbf{x} - \mathbf{u}_{\mathbf{x}|\mathbf{y}}\right\|_2^2$ from a series of observa-

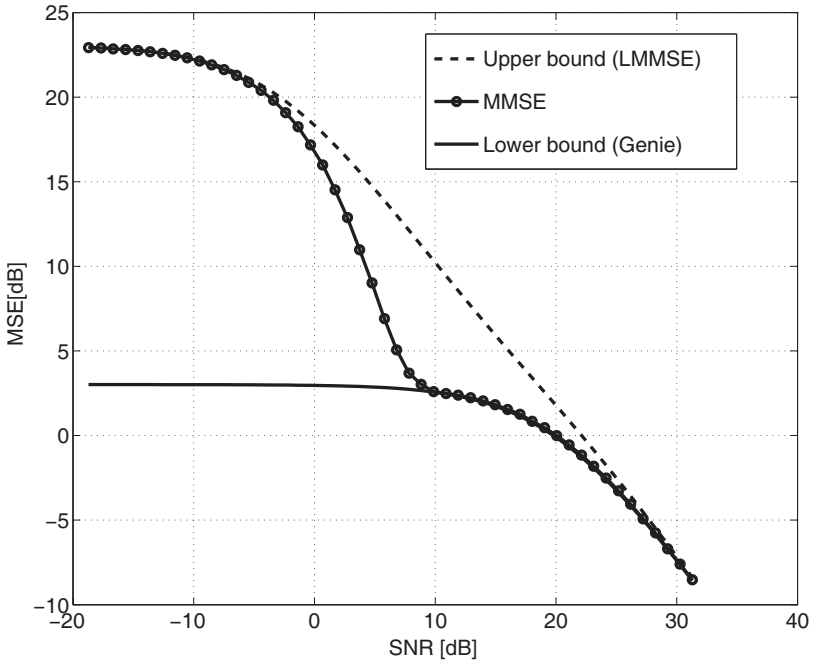

Figure 2.1: The MMSE and its analytical upper and lower bounds.

tions. However, the computational burden grows rapidly with the number of mixture components and the dimensions of $\mathbf{H}$. We have simulated several scenarios, all of which show the same trend. Therefore we only consider a low-dimensional system here, and impose inputs (2.4) such that $|\mathcal{K}| \cdot |\mathcal{L}|$ is moderate. Specifically, we assume

- $\mathbf{H} = \mathbf{I}$, with $\mathbf{I}$ being 2×2 identity matrix.
- $\mathbf{x}$ is GM distributed ($|\mathcal{K}| = 2$) with parameters

$$p_1{=}p_2{=}\frac{1}{2},\ \ \mathbf{u}_{\mathbf{x}}^{(1)}{=}\begin{bmatrix}10\\10\end{bmatrix},\ \ \mathbf{u}_{\mathbf{x}}^{(2)}{=}\begin{bmatrix}-10\\-10\end{bmatrix},\ \ \mathbf{C}_{\mathbf{xx}}^{(1)}{=}\mathbf{C}_{\mathbf{xx}}^{(2)}{=}\mathbf{I}$$

- $\mathbf{n}$ is also GM distributed ($|\mathcal{L}| = 2$), with parameters

$$q_1{=}q_2{=}\frac{1}{2},\ \ \mathbf{u}_{\mathbf{n}}^{(1)}{=}\mathbf{u}_{\mathbf{n}}^{(2)}{=}\begin{bmatrix}0\\0\end{bmatrix},\ \ \mathbf{C}_{\mathbf{nn}}^{(1)}{=}\frac{a}{2}\mathbf{I},\ \ \mathbf{C}_{\mathbf{nn}}^{(2)}{=}a\mathbf{I}$$

- The scalar a is adjusted to produce different SNR values.

Figure 2.1 shows the genie lower bound, the estimated MMSE, and the upper bound, all in dB, versus increasing SNR. The MMSE is estimated from 10^5 independent $\mathbf{y}$'s for each

SNR value. Figure 2.1 is in line our previous findings: At very low and very high SNR, the MMSE coincides with the MSE of the LMMSE estimator. Also, the bounds approach each other with increasing SNR. Interestingly, for the current setup, the MMSE estimator performs as if helped by a genie for SNR>10 dB. Thus, in practice, the MMSE estimator pinpoints the correct Gaussian component (k, l), and places nearly all weight in $\alpha^{(k,l)}(\mathbf{y})$, already from modest SNR.

From Figure 2.1 it is apparent that the MMSE estimator outperforms the LMMSE estimator, especially at intermediate SNR levels. For designers, it could therefore be of interest to know the approximate computational cost of this MSE gain. It can be determined as follows. Note from (2.14) that $\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)}$ depends on the (say $N \times M$) matrix $\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T\mathbf{C}_{\mathbf{yy}}^{-(k,l)}$. When all parameters are known, this matrix can be computed offline. Defining $|\mathcal{K}| \cdot |\mathcal{L}| = S$, it can be verified that the *online* cost for computing (2.15) is in the order of $\mathcal{O}(SNM) + \mathcal{O}(SM^2)$ multiplications and additions. The corresponding cost for the LMMSE estimate in (2.30) is only $\mathcal{O}(NM)$. Roughly, the LMMSE estimator's complexity depends on the dimensions of the system. In contrast, the MMSE estimator's complexity depends on the dimensions multiplied by the number of mixture components, S.

For the interested reader, and in order to facilitate reproducible research, the MATLAB code can be downloaded from: http://sites.google.com/site/johntorjusflaam.

2.7 Conclusion

Much motivation for this paper derives from the applicability and generality of GMs. Assuming GM inputs to a Bayesian linear model, we show that the MMSE does not come in analytical form. Existing Bayesian lower bounds are, however, not attractive in this setup. Instead the MMSE can be bounded analytically - from above and below. The LMMSE estimator yields an upper bound, and a genie aided MMSE estimator offers a lower one. The genie-aided estimator consists of a perfect decision device, followed by a decision dependent Gaussian signal and Gaussian noise MMSE estimator. We have shown that the upper and lower bounds approach each other with increasing SNR. We have also studied the MMSE in the extreme cases of zero and infinite SNR. In both of these cases, the MMSE becomes analytical and corresponds to the MSE of the LMMSE estimator. A numerical example displays the behavior of these bounds as a function of SNR. It indicates that, in practice, the MMSE estimator has performance comparable to the genie-aided estimator already from quite modest SNR.

2.8 Acknowledgments

John T. Flåm is supported by the Research Council of Norway under the NORDITE/VERDIKT program, Project CROPS2 (Grant 181530/S10). Saikat Chatterjee is funded in part by VINNOVA, the Strategic Research Area project RICSNET. Kimmo Kansanen has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement nr 216076 (FP7-SENDORA).

Paper 3

The Linear Model under Mixed Gaussian Inputs: Designing the Transfer Matrix

John T. Flåm, Dave Zachariah, Mikko Vehkaperä and Saikat Chatterjee[1]

3.1 Abstract

Suppose a linear model $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$, where inputs $\mathbf{x}, \mathbf{n}$ are independent Gaussian mixtures. The problem is to design the transfer matrix $\mathbf{H}$ so as to minimize the mean square error (MSE) when estimating $\mathbf{x}$ from $\mathbf{y}$. This problem has important applications, but faces at least three hurdles. Firstly, even for a fixed $\mathbf{H}$, the minimum MSE (MMSE) has no analytical form. Secondly, the MMSE is generally not convex in $\mathbf{H}$. Thirdly, derivatives of the MMSE w.r.t. $\mathbf{H}$ are hard to obtain. This paper casts the problem as a stochastic program and invokes gradient methods.

The study is motivated by two applications in signal processing. One concerns the choice of error-reducing precoders; the other deals with selection of pilot matrices for channel estimation. In either setting, our numerical results indicate improved estimation accuracy - markedly better than those obtained by optimal design based on standard linear estimators.

Some implications of the non-convexities of the MMSE are noteworthy, yet, to our knowledge, not well known. For example, there are cases in which more pilot power is detrimental for channel estimation. This paper explains why.

[1]John T. Flåm is with the Department of Electronics and Telecommunications, NTNU-Norwegian University of Science and Technology, Trondheim, Norway. Email: flam@iet.ntnu.no. Dave Zachariah, Mikko Vehkaperä and Saikat Chatterjee are with the School of Electrical Engineering, KTH-Royal Institute of Technology, Sweden. Emails: davez@kth.se, mikkov@kth.se, sach@kth.se.

3.2 Problem statement

Consider the following linear system

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}. \tag{3.1}$$

Here $\mathbf{y}$ is a vector of observations, and $\mathbf{x}$ and $\mathbf{n}$ are mutually independent random vectors with known Gaussian Mixture (GM) distributions:

$$\mathbf{x} \sim \sum_{k\in\mathcal{K}} p_k \mathcal{N}\big(\mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)}\big) \tag{3.2}$$

$$\mathbf{n} \sim \sum_{l\in\mathcal{L}} q_l \mathcal{N}\big(\mathbf{u}_{\mathbf{n}}^{(l)}, \mathbf{C}_{\mathbf{nn}}^{(l)}\big)\,. \tag{3.3}$$

In this work, we assume that $\mathbf{H}$ is a transfer matrix that we are at liberty to *design*, typically under some constraints. Specifically, our objective is to design $\mathbf{H}$ such that $\mathbf{x}$ can be estimated from $\mathbf{y}$ with minimum mean square error (MMSE). The MMSE, for a *fixed* $\mathbf{H}$, is by definition [41]

$$\begin{aligned}\text{MMSE} &\triangleq E\left\{\|\mathbf{x} - \mathbf{u}_{\mathbf{x}|\mathbf{y}}\|_2^2\right\} \\ &= \iint \|\mathbf{x} - \mathbf{u}_{\mathbf{x}|\mathbf{y}}\|_2^2 f(\mathbf{x},\mathbf{y}) d\mathbf{x} d\mathbf{y}.\end{aligned} \tag{3.4}$$

Here, $\|\cdot\|_2$ denotes the 2-norm, $f(\mathbf{x},\mathbf{y})$ is the joint probability density function (PDF) of $(\mathbf{x},\mathbf{y})$,

$$\mathbf{u}_{\mathbf{x}|\mathbf{y}} \triangleq E\left\{\mathbf{x}|\mathbf{y}\right\} = \int \mathbf{x} f(\mathbf{x}|\mathbf{y}) d\mathbf{x} \tag{3.5}$$

is the MMSE estimator, and $f(\mathbf{x}|\mathbf{y})$ is the PDF of $\mathbf{x}$ given $\mathbf{y}$. The MMSE in equation (3.4) depends on $\mathbf{H}$ both through $\mathbf{u}_{\mathbf{x}|\mathbf{y}}$ and $f(\mathbf{x},\mathbf{y})$. Our objective is to solve the following optimization problem

$$\min_{\mathbf{H}\in\mathbb{H}} \ \text{MMSE}, \tag{3.6}$$

where $\mathbb{H}$ denotes a prescribed set of matrices that $\mathbf{H}$ must belong to.

The next section introduces two applications where this matrix design problem appears. Section 3.4 illustrates some of the basic properties of that problem, through a simple example. Section 3.5 spells out problem (3.6) in full detail. Section 3.6 argues that a sampling based approach, involving stochastic gradients, is a viable option towards a solution. It also reviews how the Robbins-Monro method [57, 58] applies. Numerical results are provided in Section 3.7. Section 3.8 concludes. A large part of the detailed analysis, concerning stochastic gradients, is deferred to the appendix.

3.3 Background and Motivation

Problem (3.6) appears in various applications of interest. Sections 3.3.1 and 3.3.2 present two of these, which are of particular interest to the signal processing community. Section 3.3.3 introduces and motivates the use of GM input distributions. Section 3.3.4 discusses the general properties of problem (3.6), and list some of the contributions in this paper.

3.3.1 Linear precoder design

Consider a linear system model

$$\mathbf{y} = \underbrace{\mathbf{BF}}_{\mathbf{H}}\mathbf{x} + \mathbf{n}, \tag{3.7}$$

where $\mathbf{B}$ is a known matrix, and $\mathbf{F}$ is a precoder matrix to be designed such that the mean square error (MSE) when estimating $\mathbf{x}$ from $\mathbf{y}$ becomes as small as possible. The vector $\mathbf{n}$ is random noise. If $\mathbf{x}$ and $\mathbf{n}$ are independent and GM distributed, this is a matrix design problem as described in Section 3.2, where $\mathbf{F}$ is the design parameter. A typical constraint is to require that $\mathbf{Fx}$ cannot exceed a certain average power, i.e. $E\left\|\mathbf{Fx}\right\|_2^2 \leq \gamma$. Then, in terms of the minimization problem given by (3.6), the feasible set is defined by

$$\mathbb{H} = \left\{\mathbf{H} = \mathbf{BF} \text{ where } E\left\|\mathbf{Fx}\right\|_2^2 \leq \gamma\right\}.$$

A linear model with known transfer matrix $\mathbf{H}$ and GM distributed inputs is frequently assumed within speech and image processing. In these applications, the signal of interest often exhibits multi modal behavior. That feature can be reasonably explained by assuming an underlying GM distribution. The noise is often modeled as Gaussian, which is a special case of a GM. Conveniently, with (3.2) and (3.3) as inputs, the MMSE estimator in (3.5) has a closed analytical form for any given $\mathbf{H}$. Selected works exploiting this include [10, 19–22]. However, none of these works study MSE reducing precoders. These have the potential to significantly improve the estimation accuracy, and should therefore be of interest.

3.3.2 Pilot signal design

Consider a multiple-input-multiple-output (MIMO) communication model

$$\mathbf{z} = \mathbf{As} + \mathbf{n}, \tag{3.8}$$

where $\mathbf{A}$ is a random channel matrix that we wish to estimate with as small MSE as possible, and $\mathbf{s}$ is a pilot signal to be designed for that purpose. As before, $\mathbf{n}$ is random noise. In order to estimate $\mathbf{A}$ with some confidence, we must transmit as least as many pilot vectors as there are columns in $\mathbf{A}$. In addition we must assume that the realization

of $\mathbf{A}$ does not change before all pilots have been transmitted. This assumption typically holds in flat, block-fading MIMO systems [59–61]. With multiple transmitted pilots, model (3.8) can be written in matrix form as

$$\mathbf{Z} = \mathbf{AS} + \mathbf{N}. \tag{3.9}$$

If $\mathbf{A}$ is $m \times n$, then this model can be vectorized into (Thm. 2, Ch. 2, [62])

$$\underbrace{\text{vec}(\mathbf{Z})}_{\mathbf{y}} = \underbrace{\left(\mathbf{S}^T \otimes \mathbf{I}_m\right)}_{\mathbf{H}} \underbrace{\text{vec}(\mathbf{A})}_{\mathbf{x}} + \underbrace{\text{vec}(\mathbf{N})}_{\mathbf{n}}. \tag{3.10}$$

Here $\mathbf{I}_m$ denotes the $m \times m$ identity matrix, the $\text{vec}(\cdot)$ operator stacks the columns of a matrix into a column vector, and $\otimes$ denotes the Kronecker product. Assuming that the channel ($\mathbf{x}$) and noise ($\mathbf{n}$) are independent and GM distributed, this is again a design problem as described in Section 3.2, where the pilot matrix $\mathbf{S}$ is the design parameter. A natural constraint, is to impose power limitations on the transmitted pilots, i.e. $\|\mathbf{S}\|_2^2 \leq \gamma$. Again, in terms of the minimization problem given by (3.6), the feasible set is then defined by

$$\mathbb{H} = \left\{\mathbf{H} = \mathbf{S}^T \otimes \mathbf{I}_m \text{ where } \|\mathbf{S}\|_2^2 \leq \gamma\right\}.$$

In (3.10), a GM distribution on the channel $\mathbf{x}$ can account for multiple fading situations. This can be useful, for example if the source is assumed to transmit from multiple locations. Then, the commonly used Rice distribution is unlikely to accurately capture the channel statistics associated with all transmit locations (especially so in urban areas). In fact, in [63] it has been experimentally observed and reported that different transmit locations are indeed associated with different channel statistics. A GM distributed channel, with multiple modes, has the potential to capture this. As for the noise, one may either assume that $\mathbf{n}$ is pure background noise, or that $\mathbf{n}$ represents noise *and* interference. In the former case a Gaussian distribution may be justifiable, whereas in the latter a GM distribution may be more suitable [64, 65].

Note that the assumption that a channel realization can originate from several underlying distributions is not novel. For instance, all studies assuming channels governed by a Markov Model make this assumption, see e.g. [66, 67] and the references therein. A GM is a special case of an Hidden Markov model, where subsequent observations are independent, rather than governed by a Markov process. In spite of this, to the best of our knowledge, pilot optimization for estimating channels governed by a GM distribution has not been considered in the literature.

3.3.3 Gaussian Mixture distributions

While aimed at minimizing the MSE, most optimization studies on linear precoders [68, 69] or pilot signals [59–61, 70–72] utilize only the first and second moments of the input

distributions. Commonly, the underlying motivation is that a *linear* MMSE (LMMSE) estimator is employed. The LMMSE estimator[2] only relies on first and second order statistics, which conveniently tends to simplify the associated matrix design problem. In fact, the desired matrix can often be obtained as the solution of a convex optimization problem. It is known, however, that the LMMSE estimator is optimal *only* for the special case when the random signals $\mathbf{x}$ and $\mathbf{n}$ are both Gaussian. For all other cases, the LMMSE estimator is suboptimal.

In practice, purely Gaussian inputs are rare. In general, the input distributions may be asymmetric, heavy tailed and/or multi modal. A type of distribution that can accommodate all of these cases is the *Gaussian Mixture* (GM) distribution. In fact, a GM can in theory represent *any* distribution with arbitrary accuracy [12], [13]. Therefore, in this work, we assume that the inputs are GM distributed as in (3.2) and (3.3). Notation (3.2) should be read in the distributional sense, where $\mathbf{x}$ results from an imagined composite experiment. First, source $k \in \mathcal{K}$ is activated with probability $p_k \geq 0$, $\sum_{k\in\mathcal{K}} p_k = 1$. Second, that source generates a Gaussian signal with distribution law $\mathcal{N}(\mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)})$. For any realized $\mathbf{x}$, however, the underlying index k is not observable. The noise $\mathbf{n}$ emerges in an entirely similar, but independent manner. $\mathcal{K}$ and $\mathcal{L}$ are index sets. In theory, it suffices that these sets are countable, but in practice they must be finite. Clearly when $\mathcal{K}$ and $\mathcal{L}$ are singletons, one falls back to the familiar case of Gaussian inputs.

The mixture parameters, e.g. $(p_k, \mathbf{u}_{\mathbf{x}}^{(k)}, \mathbf{C}_{\mathbf{xx}}^{(k)})_{k\in\mathcal{K}}$, are rarely given a priori. Most often they must be estimated, which is generally a non-trivial task [12, 53]. A common approach is to estimate the GM parameters from training data. The expectation maximization (EM) algorithm is well suited, and much used, for that purpose [41, 49, 50]. Briefly, the algorithm relies on observations drawn from the distribution we wish to parametrize, and some initial estimate of the parameters. The observations are used to update the parameters, iteratively, until convergence to a local maximum of the likelihood function. Because the resulting GM parameters depend on the initial estimates, the algorithm can alternatively be started from multiple initial estimates. This produces multiple sets of GM parameters, and each set can be assigned probabilities based on the training data. Our starting point is that the distributions in (3.2) and (3.3) have resulted from such, or similar model fitting.

3.3.4 Problem properties and contributions

Regardless of the underlying application, solving optimization problem (3.6) is not straightforward. In particular, three hurdles stand out. Firstly, with (3.2) and (3.3) as inputs to (3.1), the MMSE in (3.4) has no analytical closed form [73]. Thus, the effect of *any* matrix $\mathbf{H}$, in terms of MMSE, cannot be evaluated exactly. Secondly, as we shall see in Section 3.4, the MMSE is generally not convex in $\mathbf{H}$. Thirdly, as will be argued in the appendix, the first and second order derivatives of the MMSE w.r.t $\mathbf{H}$ cannot be calculated exactly,

[2] Among all estimators which are linear (affine) in the observations, the LMMSE estimator obtains the smallest MSE.

and accurate approximations are hard to obtain. For these reasons, and in order to make progress, we cast the problem as a stochastic program and invoke the Robbins-Monro algorithm [57, 58]. Very briefly our approach goes as follows: We draw samples and use these to compute stochastic gradients of the MMSE. These feed into an iterative gradient method that involves projection. The contributions of the paper are several:

- As always, for greater accuracy, its preferable to use gradients instead of finite difference approximations. For this reason the paper spells out a procedure for exact realization of stochastic gradients, as given in Section 3.10.1 of the appendix. Accordingly, the Robbins-Monro algorithm comes to replace the Kiefer-Wolfowitz procedure.
- In the design phase, we exploit the known input statistics and update $\mathbf{H}$ based on samples of the inputs $(\mathbf{x}, \mathbf{n})$, instead of output $\mathbf{y}$. This yields a closed form stochastic gradient, and we prove in the initial part of the appendix that it is unbiased.
- Numerical experiments indicate that our method has far better accuracy than methods which proceed via linear estimators. The main reason is that the optimal estimator, used here, is non-linear.
- It turns out that the non-convexities of the MMSE may have practical implications that deserve being better known. Specifically, in channel estimation, it can be harmful to increase the power of the pilot signal. This paper offers an explanation.

Clearly, in many practical problems, the quantities in (3.1) are complex-valued. Throughout this paper, however, they will all be assumed real. For the analysis, this assumption introduces no loss of generality. This can be seen by considering the case when the quantities are indeed complex. Then (3.1) can be written as

$$\mathbf{y}_r + j\mathbf{y}_i = (\mathbf{H}_r + j\mathbf{H}_i)\,(\mathbf{x}_r + j\mathbf{x}_i) + \mathbf{n}_r + j\mathbf{n}_i,$$

where subscripts r and i denote real and imaginary parts respectively, and $j = \sqrt{-1}$. Note that a complex GM distribution on $\mathbf{x}$ implies that $\mathbf{x}_r$ and $\mathbf{x}_i$ are *jointly* GM distributed. That is, the real vector $\left[\begin{array}{cc} \mathbf{x}_r^T & \mathbf{x}_i^T \end{array}\right]^T$ has a GM distribution. The same argument goes of course for the real vector $\left[\begin{array}{cc} \mathbf{n}_r^T & \mathbf{n}_i^T \end{array}\right]^T$. Separating the real and imaginary parts of (3.11), the system of equations can be written as

$$\left[\begin{array}{c} \mathbf{y}_r \\ j\mathbf{y}_i \end{array}\right] = \left[\begin{array}{cc} \mathbf{H}_r & -\mathbf{H}_i \\ j\mathbf{H}_i & j\mathbf{H}_r \end{array}\right] \left[\begin{array}{c} \mathbf{x}_r \\ \mathbf{x}_i \end{array}\right] + \left[\begin{array}{c} \mathbf{n}_r \\ j\mathbf{n}_i \end{array}\right]. \tag{3.11}$$

Left-multiplying this equation by $\left[\begin{array}{cc} \mathbf{I} & 0 \\ 0 & -j\mathbf{I} \end{array}\right]$ produces

$$\underbrace{\left[\begin{array}{c} \mathbf{y}_r \\ \mathbf{y}_i \end{array}\right]}_{\tilde{\mathbf{y}}} = \underbrace{\left[\begin{array}{cc} \mathbf{H}_r & -\mathbf{H}_i \\ \mathbf{H}_i & \mathbf{H}_r \end{array}\right]}_{\tilde{\mathbf{H}}} \underbrace{\left[\begin{array}{c} \mathbf{x}_r \\ \mathbf{x}_i \end{array}\right]}_{\tilde{\mathbf{x}}} + \underbrace{\left[\begin{array}{c} \mathbf{n}_r \\ \mathbf{n}_i \end{array}\right]}_{\tilde{\mathbf{n}}}. \tag{3.12}$$

Clearly, there is a one-to-one correspondence between (3.11) and (3.12). The latter model consists exclusively of real quantities, and has similar form as (3.1). The only difference is that $\widetilde{\mathbf{H}}$ now has a specific structure. In terms of designing $\widetilde{\mathbf{H}}$, any candidate matrix must therefore obey this structure. But this can be enforced through the set of matrices that $\widetilde{\mathbf{H}}$ must belong to. Thus, we are essentially back to the same problem as (3.6) with all quantities being real.

Model (3.1) with GM inputs (3.2) and (3.3) is quite generic, and we have indicated two signal processing applications where the matrix design problem appears. The solution to that problem is, however, essentially application independent. It should therefore be of interest also to readers with different applications in mind. To the best of our knowledge, it has not been pursued in the literature.

3.4 An illustrative example

Here we study a special instance of the matrix design problem where the MMSE for *all* $\mathbf{H} \in \mathbb{H}$ can be plotted. In general this is of course not possible, but the following simple example reveals some fundamental properties of the problem. Assume that we wish to design a precoder, as in (3.7), where $\mathbf{B} = \mathbf{I}_2$ and $\mathbf{F}$ is restricted to be an orthogonal matrix. Thus, the norm of precoded signal is equal to that of the non-precoded signal. Orthogonal (unitary) precoders are reported e.g. in [74, 75]. From an implementation viewpoint, orthogonal precoding implies that the signal of interest needs no amplification - it is only *rotated*. Because $\mathbf{B} = \mathbf{I}_2$, equation (3.7) simplifies to $\mathbf{y} = \mathbf{F}\mathbf{x} + \mathbf{n}$. Further, let $\mathbf{x}$ and $\mathbf{n}$ be independent and identically GM distributed as

$$\frac{1}{2}\mathcal{N}\left(\alpha\mathbf{e}_x, \mathbf{I}_2\right) + \frac{1}{2}\mathcal{N}\left(-\alpha\mathbf{e}_x, \mathbf{I}_2\right),$$

where α is a scalar and $\mathbf{e}_x$ is the unit vector along the x-axis. Assume initially that $\mathbf{F} = \mathbf{I}_2$, which corresponds to no rotation. In this case, Figure 3.1(a) illustrates the densities of $\mathbf{F}\mathbf{x}$ (full circles) and $\mathbf{n}$ (dashed circles), when seen from above. They are identical and sit on top of each other. Now, with a precoder that rotates $\mathbf{x}$ by $\pi/2$, the densities of $\mathbf{F}\mathbf{x}$ and $\mathbf{n}$

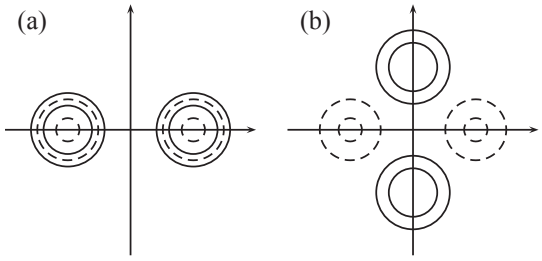

Figure 3.1: (a): Densities without any rotation. (b): The effect of rotating $\mathbf{x}$ by $\pi/2$.

will look like in Figure 3.1(b). The latter configuration is preferable from an estimation

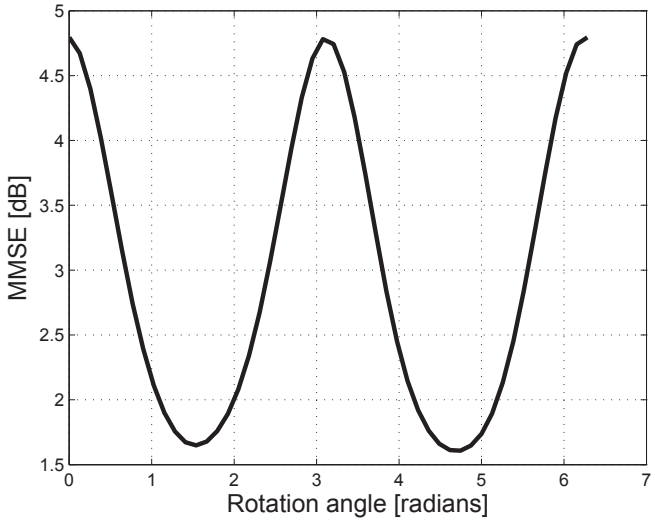

Figure 3.2: MMSE versus rotation angle.

viewpoint. This is clear from Figure 3.2, where the MMSE is displayed as a function of all rotation angles between 0 and 2π (with $\alpha = 2$). As can be seen, a significant gain can be obtained by rotating $\pi/2$ (or by $3\pi/2$). This gain is *not* due to a particularly favorable signal-to-noise-ratio SNR $= E\left\|\mathbf{Fx}\right\|_2^2 / E\left\|\mathbf{n}\right\|_2^2$; because $\mathbf{F}$ is orthogonal, the SNR remains equal for all rotation angles. The MMSE gain is instead due to a rotation producing a signal which *tends* to be orthogonal to the noise, as Figure 3.1(b) indicates.

The above example is a special case of the matrix design problem, where $\mathbf{H}$ in (3.1) is restricted to be orthogonal. It is clear that $\mathbf{H}$ plays a decisive role in how accurately $\mathbf{x}$ can be estimated. An almost equally important observation, however, is that the MMSE is not convex in $\mathbf{H}$. Hence, in general, we cannot expect that first order optimality (zero gradient) implies a *global* minimum. When studying the channel estimation problem in Section 3.7.2, we will see an implication of this non-convexity, which is perhaps not well known: In certain cases the MMSE of the channel estimator does not decrease with increasing pilot power. On the contrary, the MMSE may in fact increase.

In the next section we rewrite the original minimization problem into an equivalent but more compact maximization problem. Then, in Section, 3.6 we present a stochastic optimization approach which provides a solution.

3.5 An equivalent maximization problem

The applications in Section 3.3.1 and 3.3.2 represent special cases of the matrix design problem; in each case, $\mathbf{H}$ depends on another matrix of interest. In general, however, $\mathbf{H}$ can stand on its own and need not be a function of some other argument. Our objective is to propose a solution to the general matrix design problem, not only special cases of it. From here onwards, we therefore assume that $\mathbf{H}$ is the matrix of interest and make no assumptions about functional dependencies. We will present a framework for solving this general matrix design problem. Special cases of it can be handled by the exact same framework simply by using appropriate substitutions for $\mathbf{H}$. The substitutions for the applications in Section 3.3.1 and 3.3.2, are provided in Section 3.10.1 of the appendix.

In this section we propose a more convenient formulation of the general matrix design problem than given in (3.6). To that end we first rewrite expression (3.4). Using the results of [73], it follows that for model (3.1), under independent GM inputs (3.2) and (3.3), and a fixed $\mathbf{H}$, the MMSE can be written as

$$E\left\{\left\|\mathbf{x}-E\left\{\mathbf{x}|\mathbf{y}\right\}\right\|_2^2\right\} = \sum_k p_k\left(\text{tr}\left(\mathbf{C}_{\mathbf{xx}}^{(k)}\right)+\left\|\mathbf{u}_{\mathbf{x}}^{(k)}\right\|_2^2\right)-\int\left\|\mathbf{u}_{\mathbf{x}|\mathbf{y}}\right\|_2^2 f(\mathbf{y})d\mathbf{y}. \tag{3.13}$$

In (3.13), $\text{tr}(\cdot)$ denotes the trace operator and $f(\mathbf{y})$ is a (GM) probability density function

$$f(\mathbf{y})=\sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y}), \tag{3.14}$$

where

$$f^{(k,l)}(\mathbf{y})=\frac{e^{-\frac{1}{2}\left(\mathbf{y}-\mathbf{u}_{\mathbf{y}}^{(k,l)}\right)^T\mathbf{C}_{\mathbf{yy}}^{-(k,l)}\left(\mathbf{y}-\mathbf{u}_{\mathbf{y}}^{(k,l)}\right)}}{(2\pi)^{\frac{M}{2}}\left|\mathbf{C}_{\mathbf{yy}}^{(k,l)}\right|^{\frac{1}{2}}}, \tag{3.15}$$

$$\mathbf{u}_{\mathbf{y}}^{(k,l)}=\mathbf{H}\mathbf{u}_{\mathbf{x}}^{(k)}+\mathbf{u}_{\mathbf{n}}^{(l)}, \tag{3.16}$$

$$\mathbf{C}_{\mathbf{yy}}^{(k,l)}=\mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T+\mathbf{C}_{\mathbf{nn}}^{(l)}. \tag{3.17}$$

In (3.15), $(\cdot)^T$ denotes transposition, $|\cdot|$ denotes the determinant, $\mathbf{C}_{\mathbf{yy}}^{-(k,l)}$ is short for $(\mathbf{C}_{\mathbf{yy}}^{(k,l)})^{-1}$ and M is the length of $\mathbf{y}$. The MMSE estimator $\mathbf{u}_{\mathbf{x}|\mathbf{y}}$ in (3.13) can be written as

$$\mathbf{u}_{\mathbf{x}|\mathbf{y}}=\frac{\sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y})\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)}}{f(\mathbf{y})}, \tag{3.18}$$

where

$$\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)}=\mathbf{u}_{\mathbf{x}}^{(k)}+\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T\mathbf{C}_{\mathbf{yy}}^{-(k,l)}\left(\mathbf{y}-\mathbf{u}_{\mathbf{y}}^{(k,l)}\right). \tag{3.19}$$

In what follows, it is convenient to define

$$G(\mathbf{H}, \mathbf{y}) \triangleq \left\|\mathbf{u}_{\mathbf{x}|\mathbf{y}}\right\|_2^2. \tag{3.20}$$

This notation emphasizes that the squared norm of the MMSE estimate depends on both $\mathbf{H}$ and the observation $\mathbf{y}$. In (3.13), only the integral depends on $\mathbf{H}$. Exploiting this, and using (3.20), the minimizer of the MMSE is that $\mathbf{H}$ which maximizes

$$\int G(\mathbf{H}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y} = E\left[G(\mathbf{H}, \mathbf{y})\right] =: g(\mathbf{H}), \tag{3.21}$$

subject to

$$\mathbf{H} \in \mathbb{H}. \tag{3.22}$$

The integral in (3.21) cannot be evaluated analytically, even for a fixed and known $\mathbf{H}$ [73]. Moreover, as the example in Section 3.4 illustrated, the MMSE is generally not convex in $\mathbf{H}$, which implies that $g(\mathbf{H})$ is generally not concave. Hence, any optimization method that merely aims at first order optimality, does in general not produce a global maximizer for $g(\mathbf{H})$. Finally, as argued in Section 3.10.2 of the appendix, neither first or second order derivatives of $g(\mathbf{H})$ w.r.t. $\mathbf{H}$ can be computed exactly, and accurate approximations of these are hard to obtain.

3.6 The Robbins-Monro solution

The above observations suggest that a sampling based approach is the only viable option. The problem of maximizing a non-analytical expectation $E\left[G(\mathbf{H}, \mathbf{y})\right]$, over a parameter $\mathbf{H}$, falls under the umbrella of stochastic optimization. In particular, for our problem, the Robbins-Monro algorithm [57, 58], can be used to move iteratively from a judicially chosen initial matrix $\mathbf{H}_0$ to a local maximizer $\mathbf{H}^*$. The philosophy is to update the current matrix $\mathbf{H}$ using the gradient of the MMSE. Since that gradient cannot be calculated exactly, however, one instead relies on a stochastic approximation. Translated to our problem, the idea is briefly as follows. Although (3.21) cannot be computed analytically, it can be estimated from a set of generated sample vectors

$$\left\{\mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \mathbf{n}_t\right\}_{t=1}^N, \tag{3.23}$$

as

$$g(\mathbf{H}) \approx \frac{1}{N}\sum_{t=1}^N G(\mathbf{H}, \mathbf{y}_t) = \frac{1}{N}\sum_{t=1}^N \left\|\mathbf{u}_{\mathbf{x}|\mathbf{y_t}}\right\|_2^2. \tag{3.24}$$

In (3.23), $\mathbf{x}_t$ and $\mathbf{n}_t$ are independent realizations of the signal and noise, as generated by (3.2) and (3.3) respectively. The derivative of (3.24) w.r.t. $\mathbf{H}$ represents an approximation of $\frac{\partial g(\mathbf{H})}{\partial \mathbf{H}}$, which can be used to update the current $\mathbf{H}$. Each update is then projected onto the feasible set $\mathbb{H}$. This is the core idea of the much celebrated Robbins-Monro algorithm [57]. In our context, the algorithm can be outlined as follows.

- Let the current matrix be $\mathbf{H}_r$.
- Draw at random $(\mathbf{x}, \mathbf{n})$ and compute $\mathbf{y} = \mathbf{H}_r\mathbf{x} + \mathbf{n}$.
- Calculate the update direction (stochastic gradient) as

$$\mathbf{L}_r = \frac{\partial G(\mathbf{H}_r, \mathbf{y})}{\partial \mathbf{H}_r}, \tag{3.25}$$

  and

$$\mathbf{W}_r = \mathbf{H}_r + \epsilon_r \mathbf{L}_r, \tag{3.26}$$

  where $\{\epsilon_r\}_{r=1}^{\infty}$ is an infinite sequence of step sizes satisfying $\epsilon_r > 0$, $\epsilon_r \to 0$ and $\sum_{r=1}^{\infty} \epsilon_r = \infty$.
- Update the matrix as

$$\mathbf{H}_{r+1} = \pi_{\mathbb{H}}\left(\mathbf{W}_r\right), \tag{3.27}$$

  where $\pi_{\mathbb{H}}(\cdot)$ represents the projection onto the set of permissible matrices $\mathbb{H}$.
- Repeat all steps until convergence.

3.6.1 Remarks on the Robbins-Monro algorithm

The above algorithm is presented in terms of $\mathbf{H}$. When deriving a linear precoder, or a pilot matrix, we are more interested in the matrix $\mathbf{F}$ in (3.7), or $\mathbf{S}$ in (3.10). By invoking Lemma 1 in Section 3.10.1 of the appendix, and the substitutions explained there, the above algorithm can be used in both cases. These substitutions guarantee that the underlying structure on $\mathbf{H}$ is taken into account, and they are straightforward to implement.

Recall that the input statistics (3.2), (3.3) are assumed known. Therefore, in a design phase, it is reasonable to assume that the inputs $(\mathbf{x}, \mathbf{n})$ can be sampled to *compute* $\mathbf{y}$, as indicated in the second step of the algorithm. The alternative would be to sample $\mathbf{y}$ directly, leaving the underlying $\mathbf{x}$ and $\mathbf{n}$ unknown. The first approach is preferred because it guarantees that (3.25) becomes an unbiased estimate of $\frac{\partial g(\mathbf{H})}{\partial \mathbf{H}}$, whereas the alternative does not. This important point is fully explained in the initial part of the appendix. In general, the Robbins-Monro procedure does not *require* observing the input realizations $(\mathbf{x}, \mathbf{n})$. The algorithm converges also when only outputs $\mathbf{y}$ are available. For this reason we write (3.25) in terms of $\mathbf{y}$, but for our implementation we will assume that the underlying inputs $(\mathbf{x}, \mathbf{n})$ are fully known.

Because the stochastic gradient $\frac{\partial G(\mathbf{H},\mathbf{y})}{\partial \mathbf{H}}$ in (3.25) is central in the algorithm, its closed form expression is derived using Lemma 1 in Section 3.10.1 of the appendix. Clearly, $\frac{\partial G(\mathbf{H},\mathbf{y})}{\partial \mathbf{H}}$ is random because it relies on a random realization of $\mathbf{y}$. It can be seen as an estimator of $\frac{\partial g(\mathbf{H})}{\partial \mathbf{H}}$ based on a *single* random observation vector $\mathbf{y}$. Depending on the

input, the stochastic gradient may point in various directions; also in directions opposite to $\frac{\partial g(\mathbf{H})}{\partial \mathbf{H}}$. In order to increase the probability of a beneficial update, one can alternatively compute the stochastic gradient as an average based on multiple $\mathbf{y}$'s, as suggested in (3.24). Then (3.25) would modify to

$$\mathbf{L}_r = \frac{1}{N} \sum_{t=1}^{N} \frac{\partial G(\mathbf{H}_r, \mathbf{y}_t)}{\partial \mathbf{H}_r}.$$

In our implementation of the algorithm, however, we do not do this. In fact, it was recognized by Robbins and Monro, that choosing N large is generally inefficient. The reason is that $\mathbf{H}_r$ is only intermediate in the calculations, and as argued in the appendix, regardless of the value of N, the stochastic gradient can be chosen such that it coincides with $\frac{\partial g(\mathbf{H})}{\partial \mathbf{H}}$ *in expectation*.

The Robbins-Monro procedure does not rely on the existence of $\frac{\partial G(\mathbf{H},\mathbf{y})}{\partial \mathbf{H}}$ at all points. If this derivative is discontinuous, one can instead use any of its sub-gradients; all of which are defined. Consequently, if the local maximum towards which the algorithm converges has a discontinuous derivative, then the algorithm will oscillate around this point. Due to the decaying step sizes, however, the oscillations will eventually become infinitesimal, and for all practical purposes, the system comes to rest.

For unconstrained problems, when there are no restrictions on $\mathbf{H}$, the projection $\pi_{\mathbb{H}}(\cdot)$ in (3.27) can be left out. The solution for an unconstrained problem is an $\mathbf{H}$ with infinite norm, magnifying the signal of interest beyond boundaries, and marginalizing the effect of the noise. In practice, this translates to transmit amplifiers that consume infinite power and that cannot be realized. Unconstrained problems are therefore hardly of any practical interest. In the more realistic constrained case, the matrix $\mathbf{W}_r$ in (3.26) may not reside in the feasible set $\mathbb{H}$. In such cases, we seek its projection in $\mathbb{H}$. A common projection is to choose that $\mathbf{H} \in \mathbb{H}$ which minimizes $\|\mathbf{H} - \mathbf{W}_r\|_2$. In words, one selects that matrix in $\mathbb{H}$ which has minimum Euclidean distance to $\mathbf{W}_r$ as the update. If $\mathbf{W}_r$ is already in $\mathbb{H}$, the solution to the projection is clearly $\mathbf{H} = \mathbf{W}_r$.

Convergence towards a local optimum is guaranteed [57, 58] only as $r \to \infty$. Therefore, in theory, the algorithm must run forever in order to converge. The engineering solution, which tends to work well in practice, is to terminate the algorithm whenever $\|\mathbf{H}_{r+1} - \mathbf{H}_r\|_2 < \gamma$, where γ is a chosen threshold, or simply after a predefined number of iterations. Still, the running time may be non-negligible: If $\mathbf{H}$ is an $m \times n$ matrix, and we define $|\mathcal{K}| \cdot |\mathcal{L}| = s$, then it can be verified that the computational complexity per iteration of the Robbins-Monro algorithm is in the order of $\mathcal{O}(sm^3) + \mathcal{O}(snm^2)$ multiplications and additions. Therefore the Robins-Monro procedure is best suited for problems with limited dimensions, and when the transfer matrix can be computed once and for all. The latter would imply stationary input signals.

In general, for other problems than considered here, it may happen that the functional form of $G(\mathbf{H}, \mathbf{y})$ is unknown, even when its output can be observed for any $\mathbf{H}$ and $\mathbf{y}$. In this case, $\frac{\partial G(\mathbf{H},\mathbf{y})}{\partial \mathbf{H}}$ cannot be computed. Instead one may replace it by a *finite difference*

approximation [76]. In some cases, this may also be preferable even when the derivative can be computed; especially so if computing $\frac{\partial G(\mathbf{H},\mathbf{y})}{\partial \mathbf{H}}$ requires much more effort. When finite difference approximation are used, the procedure is known as the Kiefer-Wolfowitz algorithm [58, 76]. If the derivative can be computed, however, the Kiefer-Wolfowitz algorithm is associated with more uncertainty (larger variance) than the Robbins-Monro procedure. For the interested reader, the present paper extends [77], which considers Kiefer-Wolfowitz precoding.

3.7 Numerical results

In this section we will study two specific examples. One is on linear precoding, the other is on pilot design for channel estimation. In conformance with much of the literature, we will use the *normalized* MSE (NMSE) as performance measure[3]. This is defined as

$$\text{NMSE} = \frac{E\left\{\|\mathbf{x} - \mathbf{u}_{\mathbf{x}|\mathbf{y}}\|_2^2\right\}}{E\left\{\|\mathbf{x}\|_2^2\right\}}.$$

3.7.1 Precoder design

Here we study the performance of a Robbins-Monro precoder. As in the simple example of Section 3.4, we restrict the precoder to be orthogonal. Imposing the precoder to be an orthogonal matrix makes the projection in the Robbins-Monro algorithm particularly simple, as we shall see. For the current example we choose the following parameters.

- $\mathbf{B} = \mathbf{I}_2$.
- $\mathbf{x}$ is GM distributed with parameters

$$\begin{aligned}
&p_k = 1/4, \text{ for } k = 1...4 \\
&\mathbf{u}_{\mathbf{x}}^{(1)} = \begin{bmatrix} -10 \\ 10 \end{bmatrix}, \mathbf{u}_{\mathbf{x}}^{(2)} = \begin{bmatrix} 10 \\ -10 \end{bmatrix}, \\
&\mathbf{u}_{\mathbf{x}}^{(3)} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \mathbf{u}_{\mathbf{x}}^{(4)} = \begin{bmatrix} -10 \\ -10 \end{bmatrix}, \\
&\mathbf{C}_{\mathbf{xx}}^{(1)} = \mathbf{C}_{\mathbf{xx}}^{(2)} = \mathbf{C}_{\mathbf{xx}}^{(3)} = \mathbf{C}_{\mathbf{xx}}^{(4)} = \frac{1}{10}\mathbf{I}_2.
\end{aligned}$$

- The noise is Gaussian and distributed as

$$\mathbf{n} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, a \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}\right). \tag{3.28}$$

[3] Assuming $\mathbf{x}$ to be a zero mean signal, the NMSE is never larger than 1 (zero dB). The reason is that the MMSE estimator, $\mathbf{u}_{\mathbf{x}|\mathbf{y}}$, will coincide with the prior mean of $\mathbf{x}$ only when the SNR tends to zero. Hence, the prior mean is a worst case estimate of $\mathbf{x}$, and the NMSE describes the relative gain over the worst case estimate.

where a is a scalar that can account for any chosen SNR=tr $(\mathbf{C_{xx}})$ /tr $(\mathbf{C_{nn}})$, and $\mathbf{C_{xx}}$ and $\mathbf{C_{nn}}$ are the covariance matrices of $\mathbf{x}$ and $\mathbf{n}$ respectively.

- We use $\mathbf{F}_0 = \mathbf{I_2}$ as the initial guess in the Robbins-Monro algorithm.
- As stopping criterion we use: $\|\mathbf{F}_{r+1} - \mathbf{F}_r\|_2 < 10^{-4}$.

The Robbins-Monro algorithm described in Section 3.6 is employed using Lemma 1 in Section 3.10.1 of the appendix, and the substitution described there. For the projection in (3.27), we choose the *nearest orthogonal matrix*. This projection is the solution to the following optimization problem.

$$\mathbf{F}_{r+1} = \arg\min_{\mathbf{F}\in\mathbb{O}} \|\mathbf{F} - \mathbf{W}_r\|_2^2$$

where $\mathbb{O}$ is the set of orthogonal matrices. The solution is particularly simple, and exploits the singular value decomposition:

$$\mathbf{W}_r = \mathbf{U}\mathbf{D}\mathbf{V}^T \Rightarrow \mathbf{F}_{r+1} = \mathbf{U}\mathbf{V}^T.$$

In Figure 3.3, the two lower curves display the NMSE with and without precoding, associated with the MMSE estimator, for increasing SNR levels. As can be seen, Robbins-Monro precoding provides a significant NMSE gain, especially at SNR levels between 0 and 10dB. The upper curve shows the NMSE of the LMMSE estimator when its corresponding optimal precoder, implemented as in [69], is used. In order to make a 'fair' comparison with the MMSE estimator, we require for the LMMSE estimator that its precoder satisfies $E\|\mathbf{F}\mathbf{x}\|_2^2 = E\|\mathbf{x}\|_2^2$. Observe that this implies that the LMMSE precoder is not restricted to be an orthogonal matrix. Still, we see that the LMMSE estimator is highly suboptimal at intermediate SNR levels. In fact, it is even worse than doing MMSE estimation without any precoding. The reason is that the LMMSE estimator is linear in the data, and that it only makes use of first and second order statistics. Even though the LMMSE estimator is helped by a precoder, that precoder is inherently matched to a highly suboptimal estimator. The optimal MMSE estimator (3.18), on the other hand, is non-linear in the data and makes use of the full knowledge of the underlying distributions.

From [73], we know that the LMMSE estimator becomes *the* MMSE estimator as SNR $\to 0$ and as SNR $\to \infty$. The short explanation is that, at SNR $= 0$, both estimators do not trust the observed data. Instead they rely on the prior knowledge and always select the mean of $\mathbf{x}$ as their estimate. At SNR $= \infty$, the situation is reverse: Both estimators will fully trust the observed data, and disregard the prior knowledge. Thus, in the case of a non-singular transfer matrix, both estimators will simply use its inverse to recover $\mathbf{x}$. In terms of precoders, the implication is that the optimal LMMSE precoder will be optimal also for the MMSE estimator in these two extreme cases. For intermediate SNR levels, however, Figure 3.3 reminds us that the LMMSE estimator may be quite far from optimal. It is nevertheless widely used, because it is straightforward to implement.

The above example indicates that our method generates a reasonable precoder, for a *particular* setup. Admittedly, there exists other examples for which the gain is much less

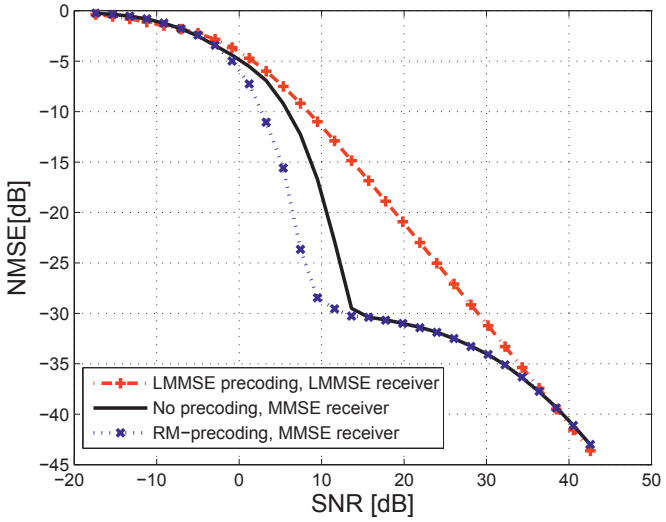

Figure 3.3: The NMSE with and without precoding.

significant. However, in all simulations we have carried out, the clear tendency is that a Robbins-Monro precoder/ MMSE receiver outperforms the LMMSE precoder/LMMSE receiver, at intermediate SNR levels.

3.7.2 Pilot design for channel estimation

Also for the channel estimation problem, the Robbins-Monro pilot matrix/ MMSE receiver outperforms the LMMSE pilot matrix/LMMSE receiver at intermediate SNR levels. In the next example we choose parameters in order to highlight this, and one additional property. That property is a direct consequence of the non-convex nature of the MMSE. We believe it to be of interest, but not well known. The starting point is the channel estimation problem in (3.9), where we assume that all matrices are 2×2. In the corresponding vectorized model

$$\underbrace{\text{vec}(\mathbf{Z})}_{\mathbf{y}} = \underbrace{\left(\mathbf{S}^T \otimes \mathbf{I}_2\right)}_{\mathbf{H}} \underbrace{\text{vec}(\mathbf{A})}_{\mathbf{x}} + \underbrace{\text{vec}(\mathbf{N})}_{\mathbf{n}}, \tag{3.29}$$

we assume the following parameters.

- The vectorized channel, $\mathbf{x}$, is distributed as $\mathcal{N}(\mathbf{0}, \mathbf{I}_4)$.

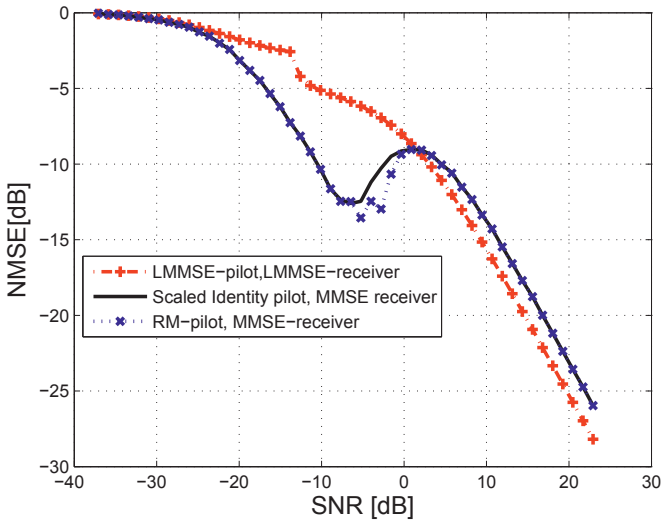

Figure 3.4: The NMSE as a function of pilot power.

- The vectorized noise, $\mathbf{n}$, is GM distributed with parameters

$$\begin{aligned} &q_l = 1/2, \text{ for } l = 1, 2, \\ &\mathbf{u}_{\mathbf{n}}^{(1)} = -\mathbf{u}_{\mathbf{n}}^{(2)} = 5 \left[\begin{array}{cccc} 1 & 1 & 1 & 1 \end{array} \right]^T, \\ &\mathbf{C}_{\mathbf{nn}}^{(1)} = \mathbf{C}_{\mathbf{nn}}^{(2)} = \frac{1}{2}\mathbf{I}_4 \end{aligned} \tag{3.30}$$

- As constraint we impose that $\|\mathbf{S}\|_2^2 = \alpha$, where α is a positive scalar that can account for any chosen pilot power, and therefore also any SNR=$\|\mathbf{S}\|_2^2 / \text{tr}\,(\mathbf{C}_{\mathbf{nn}})$.
- Stopping criterion: $\|\mathbf{S}_{r+1} - \mathbf{S}_r\|_2 < 10^{-4}$.

Again, the Robbins-Monro algorithm described in Section 3.6 is employed using Lemma 1 in Section 3.10.1 of the appendix, and the substitution described there. As starting point, we set $\mathbf{S}$ equal to a scaled identity matrix satisfying the power constraint. During the iterations we rely on the following simple projection: If the candidate pilot matrix has power $\|\mathbf{S}\|_2^2 = \gamma$, then $\mathbf{S} \to \sqrt{\frac{\alpha}{\gamma}}\mathbf{S}$. Thus, if the pilot matrix does not use the entire power budget, the magnitude of all its elements are increased. Similarly, if pilot matrix has too large power, the magnitude of its elements are decreased.

Figure 3.4 shows the estimation accuracy for increasing SNR (increasing values of α). It can bee seen that our method outperforms the commonly used LMMSE channel estimator/LMMSE pilot matrix, at intermediate SNRs. The latter scheme is implemented as in [60], and is much worse than transmitting a scaled identity pilot matrix and using the MMSE estimator. Again, the explanation is that LMMSE estimator only makes use of

lower order statistics, whereas the MMSE estimator incorporates the full knowledge of the problem. In this setup, not even an LMMSE pilot matrix can compensate for that. As the SNR tends to infinity, however, it is known that the LMMSE estimator becomes optimal [73]. The performance gap between our approach and the LMMSE estimator at high SNR therefore indicates that a scaled identity pilot matrix is poor starting point for the Robbins-Monro algorithm in this setup. One way to alleviate this problem, could of course be to run the algorithm from multiple starting points.

The most striking observation in Figure 3.4, however, is perhaps that the channel estimates may become *worse* by increasing the pilot power! This is not an artifact of the Robbins-Monro algorithm; the same tendency is seen when a scaled identity (satisfying the same power constraint) is used as the pilot matrix.

3.7.3 Increased pilot power $\neq$ improved channel estimates

We believe that the above phenomenon is not well known, and that it deserves to be explained. In order to visualize what happens, we will consider an example of smaller dimensions, but with similar properties as in the previous subsection. Specifically, we will assume that the unknown channel matrix $\mathbf{A}$ is 2×1, and that the pilot signal is just a scalar, $\mathbf{s} = a$. Then, using (3.10) it follows that $\mathbf{H} = a\mathbf{I}_2$. Thus, in this setup, we do not *optimize* anything, we only study the NMSE as function of increasing values for a. We assume the following parameters:

- $\mathbf{H} = a\mathbf{I}_2$, where a is a scalar that we can vary.
- The signal (channel) $\mathbf{x}$ is distributed as $\mathcal{N}(\mathbf{0}, \mathbf{I}_2)$.
- The noise $\mathbf{n}$ is GM distributed with parameters

$$\begin{aligned} &q_l = 1/2, \text{ for } l = 1, 2, \\ &\mathbf{u}_{\mathbf{n}}^{(1)} = -\mathbf{u}_{\mathbf{n}}^{(2)} = 5 \begin{bmatrix} 1 & 1 \end{bmatrix}^T, \\ &\mathbf{C}_{\mathbf{nn}}^{(1)} = \mathbf{C}_{\mathbf{nn}}^{(2)} = \frac{1}{2}\mathbf{I}_2 \end{aligned} \tag{3.31}$$

In Figure 3.5, the NMSE is plotted as a function of increasing values for the scalar a. We observe the same tendency as in Figure 3.4: For increasing values of a (corresponding to increasing pilot power in Figure 3.4), the NMSE may increase. In Figure 3.5, we also plot the NMSE that would be obtained by a genie aided estimator [73]. Briefly, the genie aided estimator knows from which underlying Gaussian source the noise $\mathbf{n}$ originates for each observation $\mathbf{y}$. Accordingly it can always produce the MMSE estimate corresponding to a purely Gaussian model. The genie aided estimator can of course not be implemented in practice, but because it is much better informed than the MMSE estimator, it provides a lower bound on the NMSE. Yet, from Figure 3.5 we see that for $a < 3.45$ dB, the MMSE estimator is able to pin-point the correct noise component. A plausible explanation is the following. For small a, almost all realizations of $\mathbf{H}\mathbf{x} = a\mathbf{I}_2\mathbf{x}$ are close to the

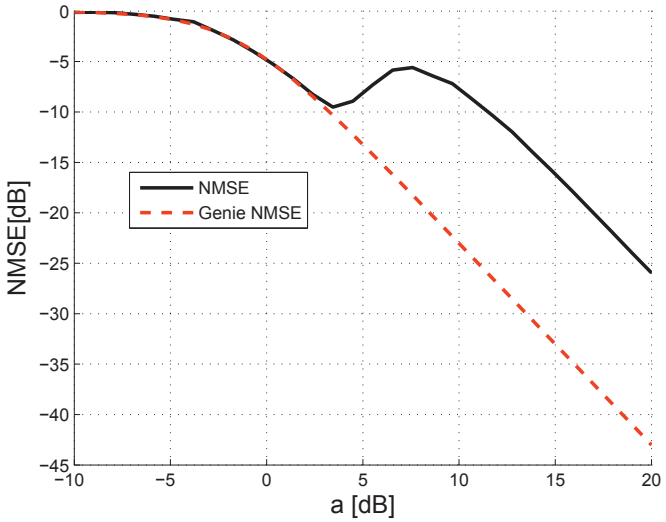

Figure 3.5: The NMSE as a function of the scalar a.

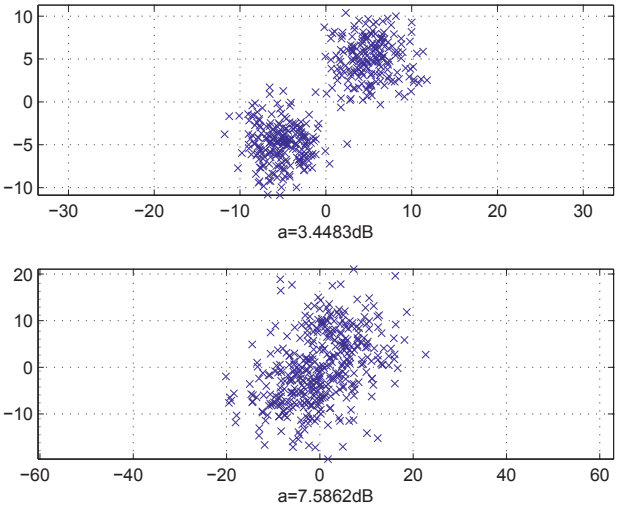

Figure 3.6: Sampled observations $\mathbf{y}$ for $a = 3.45$ dB and $a = 7.6$ dB.

origin. Thus, observations $\mathbf{y}$ tend to appear in two distinct clusters; one cluster centered at each noise component. As a consequence, the active noise component can essentially always be identified. As a grows, $\mathbf{Hx} = a\mathbf{I}_2\mathbf{x}$ take values in an increasingly larger area, and the cluster borders approach each other. The value $a = 3.45$ dB is the largest value for a where the clusters can still be 'perfectly' separated. This value corresponds to the local minimum in Figure 3.5. Because we are considering 2-dimensional random vectors, we can actually visualize these clusters. The upper part of Figure 3.6 shows how 400 independent $\mathbf{y}$'s form two nearby, but separable, clusters generated at $a = 3.45$ dB. When a grows beyond this level, the receiver faces a larger identification problem: it is harder to tell which noise component was active. The lower part of Figure 3.6 shows 400 independent $\mathbf{y}$'s generated at $a = 7.6$ dB. This value corresponds to the local maximum in Figure 3.5. Here the clusters largely overlap. As a continues to grow, however, the average magnitude of a noise contribution becomes so small compared to the average magnitude of $a\mathbf{Ix}$, that near perfect recovery of $\mathbf{x}$ eventually becomes possible.

Translated to the channel estimation problem in Figure 3.4, the interpretation is that there is a continuous range where increasing the pilot power is harmful. From Figure 3.4, one observes that, unless one can spend an additional 15 dB (approximately) on pilot power, one will not improve from the local minimum solution.

3.8 Conclusion

We have provided a framework for solving the matrix design problem of the linear model under Gaussian mixture statistics. The study is motivated by two applications in signal processing. One concerns the choice of error-reducing precoders; the other deals with selection of pilot matrices for channel estimation. In either setting we use the Robbins-Monro procedure to arrive at a solution. Our numerical results indicate improved estimation accuracy at intermediate SNR levels; markedly better than those obtained by optimal design based on the LMMSE estimator.

Although the Robbins-Monro algorithm in theory only converges asymptotically, in practice we see that a hard stopping criterion may work well. The algorithm is still computationally demanding, and therefore best suited under stationary settings and for problems with limited dimensions.

We have explored an interesting implication of the non-convexity of the MMSE; namely a case where spending more pilot power gives worse channel estimates. This phenomenon is not linked to the stochastic optimization procedure. It can be observed without optimizing $\mathbf{H}$ at all, and we have offered a plausible explanation.

3.9 Acknowledgments

John T. Flåm is supported by the Research Council of Norway under the NORDITE/VERDIKT program, Project CROPS2 (Grant 181530/S10).

3.10 Appendix

This appendix explains how to derive a closed form expression for the gradient direction $\frac{\partial G(\mathbf{H},\mathbf{y})}{\partial \mathbf{H}}$ in (3.25), where $G\left(\mathbf{H},\mathbf{y}\right)$ is defined through (3.18)-(3.20). To that end, it is worth observing that when optimizing $\mathbf{H}$ it is beneficial if the designer can draw samples directly from the inputs $\mathbf{x}$ and $\mathbf{n}$, and not only the output $\mathbf{y}$. In order to see why, assume in what follows that the order of derivation and integration can be interchanged such that

$$\begin{aligned}\frac{\partial g(\mathbf{H})}{\partial \mathbf{H}} &= \frac{\partial}{\partial \mathbf{H}}\left(\int G(\mathbf{H},\mathbf{y})f(\mathbf{y})d\mathbf{y}\right)\\ &= \int \frac{\partial}{\partial \mathbf{H}}\left[G\left(\mathbf{H},\mathbf{y}\right)f(\mathbf{y})d\mathbf{y}\right].\end{aligned} \tag{3.32}$$

Such a change of order is justified for our problem, and it can be verified by invoking Lebesgue's Dominated Convergence Theorem. Now, if we can only observe outputs $\mathbf{y}$, we have

$$E\left(\frac{\partial G\left(\mathbf{H},\mathbf{y}\right)}{\partial \mathbf{H}}\right) = \int \frac{\partial G\left(\mathbf{H},\mathbf{y}\right)}{\partial \mathbf{H}} f(\mathbf{y})d\mathbf{y} \neq \frac{\partial g(\mathbf{H})}{\partial \mathbf{H}}.$$

Hence, in this case, the update direction $\frac{\partial G(\mathbf{H},\mathbf{y})}{\partial \mathbf{H}}$ is *not* an unbiased estimator of the desired gradient $\frac{\partial g(\mathbf{H})}{\partial \mathbf{H}}$.

In contrast, assume that we can draw inputs $(\mathbf{x},\mathbf{n})$, and define

$$\left\|\mathbf{u}_{\mathbf{x}|\mathbf{H}\mathbf{x}+\mathbf{n}}\right\|_2^2 = \breve{G}\left(\mathbf{H}\mathbf{x}+\mathbf{n}\right),$$

then

$$\begin{aligned}E\left(\frac{\partial \breve{G}\left(\mathbf{H}\mathbf{x}+\mathbf{n}\right)}{\partial \mathbf{H}}\right) &= \iint \frac{\partial \breve{G}\left(\mathbf{H}\mathbf{x}+\mathbf{n}\right)}{\partial \mathbf{H}} f(\mathbf{x})d\mathbf{x}f(\mathbf{n})d\mathbf{n}\\ = \iint \frac{\partial}{\partial \mathbf{H}}\left[\breve{G}\left(\mathbf{H}\mathbf{x}+\mathbf{n}\right)f(\mathbf{x})d\mathbf{x}f(\mathbf{n})d\mathbf{n}\right] &= \frac{\partial g(\mathbf{H})}{\partial \mathbf{H}}.\end{aligned}$$

Here, the second equality holds because $f(\mathbf{x})d\mathbf{x}f(\mathbf{n})d\mathbf{n}$ is independent of $\mathbf{H}$. Hence, $\frac{\partial \breve{G}(\mathbf{H}\mathbf{x}+\mathbf{n})}{\partial \mathbf{H}}$ *is* an unbiased estimator of $\frac{\partial g(\mathbf{H})}{\partial \mathbf{H}}$, which is of course desirable. Because it is beneficial to sample $\mathbf{x}$ and $\mathbf{n}$, rather than just $\mathbf{y}$, we will assume here that the designer can do this. In practice, this implies that the optimization of $\mathbf{H}$ is done off line, as preparation for the subsequent estimation.

In what follows, we will derive a closed form expression for $\frac{\partial \breve{G}(\mathbf{Hx}+\mathbf{n})}{\partial \mathbf{H}}$, and use this as the update direction in (3.25). Although we assume knowledge of $(\mathbf{x}, \mathbf{n})$ for each observed $\mathbf{y}$, we will write $\mathbf{y}$ instead of $\mathbf{Hx}+\mathbf{n}$, and $\frac{\partial G(\mathbf{H},\mathbf{y})}{\partial \mathbf{H}}$ instead of $\frac{\partial \breve{G}(\mathbf{Hx}+\mathbf{n})}{\partial \mathbf{H}}$, simply to save space.

Using (3.18), $\frac{\partial G(\mathbf{H},\mathbf{y})}{\partial \mathbf{H}}$ can be written as

$$\sum_{k,l,r,s} p_k q_l p_r q_s \frac{\partial}{\partial \mathbf{H}} \left(\frac{f^{(k,l)}(\mathbf{y}) f^{(r,s)}(\mathbf{y}) \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)^T} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)}}{\left(\sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y})\right)^2} \right). \tag{3.33}$$

In order to compute (3.33), we make use of the following theorem [62].

Theorem 1. *For a scalar function, $\phi(\mathbf{H})$, of a matrix argument, the differential has the form*

$$d(\phi) = tr\left(\mathbf{Q}^T d(\mathbf{H})\right) = vec(\mathbf{Q})^T vec(d\mathbf{H}),$$
$$where \quad \mathbf{Q} = \frac{\partial \phi}{\partial \mathbf{H}}.$$

In our case, we take $\phi(\mathbf{H})$ to be the expression in the large parenthesis of (3.33). We will identify its differential, and exploit Theorem 1 in order to obtain the derivative. For that purpose, it is convenient to define

$$f^{k,l,r,s} = f^{(k,l)}(\mathbf{y}) f^{(r,s)}(\mathbf{y}), \tag{3.34}$$
$$z^{k,l,r,s} = \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)^T} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)}, \tag{3.35}$$
$$t = \sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y}). \tag{3.36}$$

Using these, the derivative in (3.33), can then be compactly written as $\frac{\partial}{\partial \mathbf{H}} \left(\frac{f^{k,l,r,s} z^{k,l,r,s}}{t^2} \right)$. Using the chain rule, the differential of this fraction is

$$d(\phi) = d\left(\frac{f^{k,l,r,s} z^{k,l,r,s}}{t^2}\right) = -\frac{2 f^{k,l,r,s} z^{k,l,r,s} d(t)}{t^3} + \frac{d(f^{k,l,r,s}) z^{k,l,r,s} + d(z^{k,l,r,s}) f^{k,l,r,s}}{t^2}. \tag{3.37}$$

Thus, we must identify the differentials $d(f^{k,l,r,s})$, $d(z^{k,l,r,s})$ and $d(t)$, which we do in next. **Notation:** we will in the remainder of this appendix use $\langle \cdot \rangle$ to compactly denote the trace operator.

Computing $d(f^{k,l,r,s})$

$$d(f^{k,l,r,s}) = d\left(f^{(k,l)}(\mathbf{y}) f^{(r,s)}(\mathbf{y})\right) = d\left(f^{(k,l)}(\mathbf{y})\right) f^{(r,s)}(\mathbf{y}) + f^{(k,l)}(\mathbf{y}) d\left(f^{(r,s)}(\mathbf{y})\right). \tag{3.38}$$

The differential $d\left(f^{(k,l)}(\mathbf{y})\right)$, is a differential of a Gaussian probability density function. In our case it depends on the indexes (k,l), but in order to enhance readability, we will disregard these indexes in what follows. Thus, for now, we will use equations (3.14)-(3.19) with all indexes removed, and reincorporate the indexes when needed. In addition, we will for now disregard the constant factor $(2\pi)^{-\frac{M}{2}}$ in (3.15). Hence, instead of considering the differential $d\left(f^{(k,l)}(\mathbf{y})\right)$, we therefore consider $d\left(|\mathbf{C_{yy}}|^{-\frac{1}{2}}\, g(\mathbf{y})\right)$, where $g(\mathbf{y}) = e^{-\frac{1}{2}(\mathbf{y}-\mathbf{u_y})^T\mathbf{C_{yy}^{-1}}(\mathbf{y}-\mathbf{u_y})}$. This can be written as

$$\begin{aligned}
&d\left(|\mathbf{C_{yy}}|^{-\frac{1}{2}}\, g(\mathbf{y})\right) \\
&= d\left(|\mathbf{C_{yy}}|^{-\frac{1}{2}}\right) g(\mathbf{y}) + |\mathbf{C_{yy}}|^{-\frac{1}{2}}\, d\left(g(\mathbf{y})\right) \\
&= -\frac{g(\mathbf{y})}{2}\,|\mathbf{C_{yy}}|^{-\frac{3}{2}}\, d\left(|\mathbf{C_{yy}}|\right) \\
&\quad - \frac{g(\mathbf{y})}{2}\,|\mathbf{C_{yy}}|^{-\frac{1}{2}}\, d\left((\mathbf{y}-\mathbf{u_y})^T\,\mathbf{C_{yy}^{-1}}\,(\mathbf{y}-\mathbf{u_y})\right)
\end{aligned} \tag{3.39}$$

In the second equality we have used the chain rule, and exploited that $g(\mathbf{y})$ is an exponential function. The first differential in (3.39), provided $\mathbf{C_{yy}}$ is full rank, is (Theorem 1, ch. 8, of [62])

$$\begin{aligned}
d\left(|\mathbf{C_{yy}}|\right) &= |\mathbf{C_{yy}}|\left\langle \mathbf{C_{yy}^{-1}} d\left(\mathbf{C_{yy}}\right)\right\rangle && (3.40)\\
&= |\mathbf{C_{yy}}|\left\langle \mathbf{C_{yy}^{-1}} d\left(\mathbf{H}\mathbf{C_{xx}}\mathbf{H}^T + \mathbf{C_{nn}}\right)\right\rangle \\
&= |\mathbf{C_{yy}}|\left\langle \mathbf{C_{yy}^{-1}}\left(d(\mathbf{H})\mathbf{C_{xx}}\mathbf{H}^T + \mathbf{H}\mathbf{C_{xx}} d\left(\mathbf{H}^T\right)\right)\right\rangle && (3.41)\\
&= |\mathbf{C_{yy}}|\left\langle \mathbf{C_{xx}}\mathbf{H}^T\mathbf{C_{yy}^{-1}} d(\mathbf{H}) + d(\mathbf{H})\mathbf{C_{xx}}\mathbf{H}^T\mathbf{C_{yy}^{-1}}\right\rangle && (3.42)\\
&= |\mathbf{C_{yy}}|\left\langle \mathbf{C_{xx}}\mathbf{H}^T\mathbf{C_{yy}^{-1}} d(\mathbf{H}) + \mathbf{C_{xx}}\mathbf{H}^T\mathbf{C_{yy}^{-1}} d(\mathbf{H})\right\rangle && (3.43)\\
&= 2\,|\mathbf{C_{yy}}|\left\langle \mathbf{C_{xx}}\mathbf{H}^T\mathbf{C_{yy}^{-1}} d(\mathbf{H})\right\rangle \\
&= 2\,|\mathbf{C_{yy}}|\left\langle \left(\mathbf{C_{yy}^{-1}}\mathbf{H}\mathbf{C_{xx}}\right)^T d(\mathbf{H})\right\rangle . && (3.44)
\end{aligned}$$

In (3.42), we have rotated the first trace (done a cyclic permutation of the matrix product), and transposed the second trace. Because $\mathbf{C_{xx}}$ and $\mathbf{C_{yy}^{-1}}$ are symmetric, they are not affected by transposition. Moreover, $d(\mathbf{H}^T) = (d(\mathbf{H}))^T$. The trace operator is invariant to such rotations and transposition, and therefore these operations are justified. In (3.43) we have rotated the second term. Such rotations and transpositions will be frequently employed throughout. Introducing $\mathbf{w} = \mathbf{y} - \mathbf{u_y}$, the second differential of (3.39) can be written

$$\begin{aligned}
&d\left(\mathbf{w}^T\mathbf{C_{yy}^{-1}}\mathbf{w}\right) = d\left(\left\langle \mathbf{w}^T\mathbf{C_{yy}^{-1}}\mathbf{w}\right\rangle\right) \\
&= \left\langle \mathbf{w}^T d\left(\mathbf{C_{yy}^{-1}}\right)\mathbf{w}\right\rangle + 2\left\langle \mathbf{w}^T\mathbf{C_{yy}^{-1}} d\left(\mathbf{w}\right)\right\rangle .
\end{aligned} \tag{3.45}$$

The first term of (3.45) is

$$\left\langle \mathbf{w}^T d\left(\mathbf{C}_{\mathbf{yy}}^{-1}\right)\mathbf{w}\right\rangle \tag{3.46}$$

$$= -\left\langle \mathbf{w}^T\mathbf{C}_{\mathbf{yy}}^{-1} d\left(\mathbf{C}_{\mathbf{yy}}\right)\mathbf{C}_{\mathbf{yy}}^{-1}\mathbf{w}\right\rangle \tag{3.47}$$

$$= -\left\langle \mathbf{w}^T\mathbf{C}_{\mathbf{yy}}^{-1}\left(d(\mathbf{H})\mathbf{C}_{\mathbf{xx}}\mathbf{H}^T + \mathbf{H}\mathbf{C}_{\mathbf{xx}} d\left(\mathbf{H}^T\right)\right)\mathbf{C}_{\mathbf{yy}}^{-1}\mathbf{w}\right\rangle$$

$$= -\left\langle \mathbf{C}_{\mathbf{yy}}^{-1}\mathbf{w}\mathbf{w}^T\mathbf{C}_{\mathbf{yy}}^{-1}\left(d(\mathbf{H})\mathbf{C}_{\mathbf{xx}}\mathbf{H}^T + \mathbf{H}\mathbf{C}_{\mathbf{xx}} d\left(\mathbf{H}^T\right)\right)\right\rangle. \tag{3.48}$$

Equation (3.47) results from Theorem 3, ch. 8, of [62]. Observe that $\mathbf{C}_{\mathbf{yy}}^{-1}\mathbf{w}\mathbf{w}^T\mathbf{C}_{\mathbf{yy}}^{-1}$ in (3.48) is a symmetric matrix, playing the same role as $\mathbf{C}_{\mathbf{yy}}^{-1}$ in (3.41). Therefore, we can utilize (3.44) and conclude that

$$\left\langle \mathbf{w}^T d\left(\mathbf{C}_{\mathbf{yy}}^{-1}\right)\mathbf{w}\right\rangle = -2\left\langle \left(\mathbf{C}_{\mathbf{yy}}^{-1}\mathbf{w}\mathbf{w}^T\mathbf{C}_{\mathbf{yy}}^{-1}\mathbf{H}\mathbf{C}_{\mathbf{xx}}\right)^T d(\mathbf{H})\right\rangle. \tag{3.49}$$

Recall that $\mathbf{w} = \mathbf{H}(\mathbf{x}-\mathbf{u}_{\mathbf{x}}) + \mathbf{n} - \mathbf{u}_{\mathbf{n}}$. The second term of (3.45) can therefore be written as

$$\begin{aligned}
&2\left\langle \mathbf{w}^T\mathbf{C}_{\mathbf{yy}}^{-1} d\left(\mathbf{w}\right)\right\rangle \\
&= 2\left\langle \mathbf{w}^T\mathbf{C}_{\mathbf{yy}}^{-1} d\left(\mathbf{H}\right)\left(\mathbf{x}-\mathbf{u}_{\mathbf{x}}\right)\right\rangle \\
&= 2\left\langle \left(\mathbf{x}-\mathbf{u}_{\mathbf{x}}\right)\mathbf{w}^T\mathbf{C}_{\mathbf{yy}}^{-1} d\left(\mathbf{H}\right)\right\rangle \\
&= 2\left\langle \left(\mathbf{C}_{\mathbf{yy}}^{-1}\mathbf{w}\left(\mathbf{x}-\mathbf{u}_{\mathbf{x}}\right)^T\right)^T d\left(\mathbf{H}\right)\right\rangle.
\end{aligned} \tag{3.50}$$

Using (3.44),(3.49) and (3.50), and inserting into (3.39), we find that

$$\begin{aligned}
&d\left(\left|\mathbf{C}_{\mathbf{yy}}\right|^{-\frac{1}{2}} g(\mathbf{y})\right) \\
&= -g(\mathbf{y})\left|\mathbf{C}_{\mathbf{yy}}\right|^{-\frac{1}{2}}\left\langle \left(\mathbf{C}_{\mathbf{yy}}^{-1}\mathbf{H}\mathbf{C}_{\mathbf{xx}}\right)^T d(\mathbf{H})\right\rangle \\
&\quad + g(\mathbf{y})\left|\mathbf{C}_{\mathbf{yy}}\right|^{-\frac{1}{2}}\left\langle \left(\mathbf{C}_{\mathbf{yy}}^{-1}\mathbf{w}\mathbf{w}^T\mathbf{C}_{\mathbf{yy}}^{-1}\mathbf{H}\mathbf{C}_{\mathbf{xx}}\right)^T d(\mathbf{H})\right\rangle \\
&\quad - g(\mathbf{y})\left|\mathbf{C}_{\mathbf{yy}}\right|^{-\frac{1}{2}}\left\langle \left(\mathbf{C}_{\mathbf{yy}}^{-1}\mathbf{w}\left(\mathbf{x}-\mathbf{u}_{\mathbf{x}}\right)^T\right)^T d\left(\mathbf{H}\right)\right\rangle.
\end{aligned} \tag{3.51}$$

If we define

$$\begin{aligned}
\mathbf{R}^{(k,l)} = &\mathbf{C}_{\mathbf{yy}}^{-(k,l)}\mathbf{w}^{(k,l)}\left(\mathbf{x}-\mathbf{u}_{\mathbf{x}}^{(k)}\right)^T \\
&+ \mathbf{C}_{\mathbf{yy}}^{-(k,l)}\left(\mathbf{I} - \mathbf{w}^{(k,l)}\mathbf{w}^{(k,l)^T}\mathbf{C}_{\mathbf{yy}}^{-(k,l)}\right)\mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)},
\end{aligned}$$

where $\mathbf{w}^{(k,l)} = \mathbf{y} - \mathbf{u}_{\mathbf{y}}^{(k,l)}$, and reincorporate the constant factor $(2\pi)^{-\frac{M}{2}}$, we now find that

$$d\left(f^{(k,l)}(\mathbf{y})\right) = -f^{(k,l)}(\mathbf{y})\left\langle \left(\mathbf{R}^{(k,l)}\right)^T d\left(\mathbf{H}\right)\right\rangle. \tag{3.52}$$

Accordingly, (3.38) becomes

$$d\left(f^{k,l,r,s}\right) = -\left\langle f^{k,l,r,s}\left(\mathbf{R}^{(k,l)} + \mathbf{R}^{(r,s)}\right)^T d(\mathbf{H})\right\rangle. \tag{3.53}$$

Computing $d(z^{k,l,r,s})$

$$\begin{aligned} d(z^{k,l,r,s}) =& d\left(\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)^T}\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)}\right) \\ &= \left\langle d\left(\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)^T}\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)}\right)\right\rangle \\ &= \left\langle \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T} d\left(\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)}\right) + \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)^T} d\left(\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)}\right)\right\rangle. \end{aligned} \tag{3.54}$$

Apart from a rearrangement of the indexes, equation (3.54) contains two similar terms. Hence it suffices to compute one of them. Recalling that $\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)}$ is defined by (3.19), we focus on the differential

$$\begin{aligned} &\left\langle \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T} d\left(\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(k,l)}\right)\right\rangle \\ &= \left\langle \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T} d\left(\mathbf{u}_{\mathbf{x}}^{(k)} + \mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T\mathbf{C}_{\mathbf{yy}}^{-(k,l)}\mathbf{w}^{(k,l)}\right)\right\rangle \\ &= \left\langle \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T}\mathbf{C}_{\mathbf{xx}}^{(k)} d\left(\mathbf{H}^T\right)\mathbf{C}_{\mathbf{yy}}^{-(k,l)}\mathbf{w}^{(k,l)}\right\rangle && (3.55) \\ &\quad + \left\langle \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T d\left(\mathbf{C}_{\mathbf{yy}}^{-(k,l)}\right)\mathbf{w}^{(k,l)}\right\rangle && (3.56) \\ &\quad + \left\langle \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T\mathbf{C}_{\mathbf{yy}}^{-(k,l)} d\left(\mathbf{w}^{(k,l)}\right)\right\rangle. && (3.57) \end{aligned}$$

We will resolve this term by term. The first term, (3.55), reads

$$\begin{aligned} &\left\langle \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T}\mathbf{C}_{\mathbf{xx}}^{(k)} d\left(\mathbf{H}^T\right)\mathbf{C}_{\mathbf{yy}}^{-(k,l)}\mathbf{w}^{(k,l)}\right\rangle \\ &= \left\langle \left(\mathbf{C}_{\mathbf{yy}}^{-(k,l)}\mathbf{w}^{(k,l)}\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T}\mathbf{C}_{\mathbf{xx}}^{(k)}\right)^T d\left(\mathbf{H}\right)\right\rangle. \end{aligned} \tag{3.58}$$

The second term, (3.56), can be written as

$$\begin{aligned} &\left\langle \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T d\left(\mathbf{C}_{\mathbf{yy}}^{-(k,l)}\right)\mathbf{w}^{(k,l)}\right\rangle \\ &= -\left\langle \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T\mathbf{C}_{\mathbf{yy}}^{-(k,l)} d\left(\mathbf{C}_{\mathbf{yy}}^{(k,l)}\right)\mathbf{C}_{\mathbf{yy}}^{-(k,l)}\mathbf{w}^{(k,l)}\right\rangle \\ &= -\left\langle \underbrace{\mathbf{C}_{\mathbf{yy}}^{-(k,l)}\mathbf{w}^{(k,l)}\mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T}\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T\mathbf{C}_{\mathbf{yy}}^{-(k,l)}}_{\mathbf{C}^{(k,l,r,s)}} d\left(\mathbf{C}_{\mathbf{yy}}^{(k,l)}\right)\right\rangle \\ &= -\left\langle \mathbf{C}^{(k,l,r,s)}\left(d\left(\mathbf{H}\right)\mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T + \mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)} d\left(\mathbf{H}^T\right)\right)\right\rangle \\ &= -\left\langle \mathbf{C}_{\mathbf{xx}}^{(k)}\mathbf{H}^T\left(\mathbf{C}^{(k,l,r,s)} + \mathbf{C}^{(k,l,r,s)^T}\right) d(\mathbf{H})\right\rangle \\ &= -\left\langle \left(\left(\mathbf{C}^{(k,l,r,s)} + \mathbf{C}^{(k,l,r,s)^T}\right)\mathbf{H}\mathbf{C}_{\mathbf{xx}}^{(k)}\right)^T d(\mathbf{H})\right\rangle. \end{aligned} \tag{3.59}$$

The third term, (3.57), reads

$$\begin{aligned}
&\left\langle \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T} \mathbf{C}_{\mathbf{xx}}^{(k)} \mathbf{H}^T \mathbf{C}_{\mathbf{yy}}^{-(k,l)} d\left(\mathbf{w}^{(k,l)}\right) \right\rangle \\
&= \left\langle \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T} \mathbf{C}_{\mathbf{xx}}^{(k)} \mathbf{H}^T \mathbf{C}_{\mathbf{yy}}^{-(k,l)} d\left(\mathbf{H}\right) \left(\mathbf{x} - \mathbf{u}_{\mathbf{x}}^{(k)}\right) \right\rangle \\
&= \left\langle \left(\mathbf{C}_{\mathbf{yy}}^{-(k,l)} \mathbf{H} \mathbf{C}_{\mathbf{xx}}^{(k)} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)} \left(\mathbf{x} - \mathbf{u}_{\mathbf{x}}^{(k)}\right)^T \right)^T d\left(\mathbf{H}\right) \right\rangle . \qquad (3.60)
\end{aligned}$$

Using (3.58),(3.59) and (3.60) we now define

$$\begin{aligned}
\mathbf{D}^{(k,l,r,s)} = &\mathbf{C}_{\mathbf{yy}}^{-(k,l)} \mathbf{w}^{(k,l)} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)^T} \mathbf{C}_{\mathbf{xx}}^{(k)} \\
&- \left(\mathbf{C}^{(k,l,r,s)} + \mathbf{C}^{(k,l,r,s)^T} \right) \mathbf{H} \mathbf{C}_{\mathbf{xx}}^{(k)} \\
&+ \mathbf{C}_{\mathbf{yy}}^{-(k,l)} \mathbf{H} \mathbf{C}_{\mathbf{xx}}^{(k)} \mathbf{u}_{\mathbf{x}|\mathbf{y}}^{(r,s)} \left(\mathbf{x} - \mathbf{u}_{\mathbf{x}}^{(k)}\right)^T .
\end{aligned}$$

Due to its two similar terms, the differential in (3.54) can then be written

$$d(z^{k,l,r,s}) = \left\langle \left(\mathbf{D}^{(k,l,r,s)} + \mathbf{D}^{(r,s,k,l)} \right)^T d(\mathbf{H}) \right\rangle . \qquad (3.61)$$

Computing $d(t)$

$$\begin{aligned}
d(t) &= d\left(\sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y}) \right) = \sum_{k,l} p_k q_l d\left(f^{(k,l)}(\mathbf{y}) \right) \\
&= -\sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y}) \left\langle \left(\mathbf{R}^{(k,l)} \right)^T d(\mathbf{H}) \right\rangle . \qquad (3.62)
\end{aligned}$$

The last equation results immediately by employing (3.52).

3.10.1 Computing the derivative

In order to obtain the derivative, we first formulate the following lemma:
Lemma 1. *Utilizing (3.53), (3.61) and (3.62), expression (3.37), which is the differential of the objective function in the matrix design problem, can be written as*

$$\begin{aligned}
d(\phi) &= d\left(\frac{f^{k,l,r,s} z^{k,l,r,s}}{t^2} \right) \\
&= -\frac{\left\langle f^{k,l,r,s} \left(\mathbf{R}^{(k,l)} + \mathbf{R}^{(r,s)} \right)^T d(\mathbf{H}) \right\rangle z^{k,l,r,s}}{t^2} \\
&+ \frac{\left\langle \left(\mathbf{D}^{(k,l,r,s)} + \mathbf{D}^{(r,s,k,l)} \right)^T d(\mathbf{H}) \right\rangle f^{k,l,r,s}}{t^2} \\
&+ \frac{2 f^{k,l,r,s} z^{k,l,r,s} \sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y}) \left\langle \left(\mathbf{R}^{(k,l)} \right)^T d(\mathbf{H}) \right\rangle}{t^3} . \qquad (3.63)
\end{aligned}$$

Now it is a matter of applying Theorem 1 on (3.63) in order to identify the derivative. This is generally straightforward. In case of a precoder design problem, one makes the following substitutions: $\mathbf{H} = \mathbf{BF}$ and $d(\mathbf{H}) = \mathbf{B}d(\mathbf{F})$ throughout. In case of the pilot design problem (3.10), $\mathbf{H} = \mathbf{S}^T \otimes \mathbf{I}_m$. In addition, assuming that $\mathbf{S}$ is $n \times r$, we have

$$\begin{aligned}\text{vec}(d\mathbf{H}) &= \text{vec}\left(d(\mathbf{S}^T) \otimes \mathbf{I}_m\right) \\ &= (\mathbf{I}_n \otimes \mathbf{K}_{mr} \otimes \mathbf{I}_m)\,(\mathbf{I}_{rn} \otimes \text{vec}(\mathbf{I}_m))\,d(\text{vec}(\mathbf{S}^T)).\end{aligned}$$

In the latter equality $\mathbf{K}_{mr}$ is the Magnus and Neudecker commutation matrix [62].

If $\mathbf{H}$ in (3.63) is *not* a function of some other argument, identifying the derivative from (3.63) becomes particularly simple. Compactly defining $p_k q_l p_r q_s f^{k,l,r,s} = h^{k,l,r,s}$, and observing that $\sum_{k,l,r,s} h^{k,l,r,s} = t^2$, we find from equations (3.33), (3.63), and Theorem 1 that the stochastic gradient is given by

$$\begin{aligned}&\frac{\partial \breve{G}\left(\mathbf{Hx}+\mathbf{n}\right)}{\partial \mathbf{H}} = \\ &-\frac{\sum_{k,l,r,s} h^{k,l,r,s}\left(\mathbf{R}^{(k,l)}+\mathbf{R}^{(r,s)}\right) z^{k,l,r,s}}{\sum_{k,l,r,s} h^{k,l,r,s}} \\ &+\frac{\sum_{k,l,r,s} h^{k,l,r,s}\left(\mathbf{D}^{(k,l,r,s)}+\mathbf{D}^{(r,s,k,l)}\right)}{\sum_{k,l,r,s} h^{k,l,r,s}} \\ &+\frac{2\sum_{k,l,r,s} h^{k,l,r,s} z^{k,l,r,s} \sum_{k,l} p_k q_l f^{(k,l)}(\mathbf{y})\mathbf{R}^{(k,l)}}{\sum_{i,j} p_i q_j f^{(i,j)}(\mathbf{y}) \sum_{k,l,r,s} h^{k,l,r,s}}.\end{aligned} \tag{3.64}$$

3.10.2 First and second order derivatives of the objective function

When trying to compute

$$\iint \frac{\partial \breve{G}\left(\mathbf{Hx}+\mathbf{n}\right)}{\partial \mathbf{H}} f(\mathbf{x}) f(\mathbf{n}) d\mathbf{x} d\mathbf{n}, \tag{3.65}$$

the mixture densities in the denominators of (3.64) will not simplify by substitutions. An entirely similar argument provides the reason for why (3.21) cannot be computed analytically in the first place [73]. Hence, (3.65) cannot be computed analytically, and a closed form derivative of (3.21) w.r.t $\mathbf{H}$ does not exist. A similar argument will hold also for the second order derivative.

Paper 4

Pilot Design for MIMO channel estimation: An Alternative to the Kronecker Structure Assumption

John Flåm, Emil Björnson and Saikat Chatterjee[1]

4.1 Abstract

This work seeks to design a pilot signal under a power constraint, such that the channel can be estimated with minimum mean square error. The procedure we propose does not assume Kronecker structure on the underlying covariance matrices, and the pilot signal is obtained in three main steps. Firstly, we solve a relaxed convex version of the original minimization problem. Secondly, its solution is projected onto the feasible set. Thirdly we use the projected solution as starting point for an augmented Lagrangian method. Numerical experiments indicate that this procedure may produce pilot signals that are far better than those obtained under the Kronecker structure assumption.

4.2 Problem statement

Consider the following multiple-input-multiple-output (MIMO) communication model

$$\mathbf{z} = \mathbf{H}\mathbf{s} + \mathbf{w}. \tag{4.1}$$

[1]John T. Flåm is with the Department of Electronics and Telecommunications, NTNU-Norwegian University of Science and Technology, Trondheim, Norway. Email: flam@iet.ntnu.no. Emil Björnson was with the School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden. He is currently with the Alcatel-Lucent Chair on Flexible Radio, SUPELEC, Gif-sur-Yvette, France.

Here $\mathbf{z}$ is the observed output, $\mathbf{w}$ is random noise, $\mathbf{H}$ is a random channel matrix that we wish to estimate, and $\mathbf{s}$ is a pilot vector to be designed for that purpose. In order to estimate $\mathbf{H}$ with some confidence, we should typically send at least as many pilot vectors at there are columns in $\mathbf{H}$, although this is not strictly necessary when the columns are strongly correlated [61]. In order to utilize the channel estimate for subsequent data transmission, we also assume that the time needed to transmit the pilot signal is only a fraction of the coherence time. This assumption typically holds in flat, block-fading MIMO systems [59–61]. With p transmitted pilots, model (4.1) can be written in matrix form as

$$\mathbf{Z} = \mathbf{HS} + \mathbf{W}. \tag{4.2}$$

We assume that $\mathbf{H} \in \mathbb{C}^{m\times n}$ and that the *pilot matrix* $\mathbf{S} \in \mathbb{C}^{n\times p}$. Vectorizing equation (4.2) gives [78, Lemma 2.11]

$$\underbrace{\operatorname{vec}(\mathbf{Z})}_{\mathbf{y}} = \underbrace{\left(\mathbf{S}^T \otimes \mathbf{I}_m\right)}_{\mathbf{G}} \underbrace{\operatorname{vec}(\mathbf{H})}_{\mathbf{x}} + \underbrace{\operatorname{vec}(\mathbf{W})}_{\mathbf{n}}, \tag{4.3}$$

where $\mathbf{I}_m$ denotes the $m \times m$ identity matrix, the $\operatorname{vec}(\cdot)$ operator stacks the columns of a matrix into a column vector, $\otimes$ denotes the Kronecker product, and $(\cdot)^T$ denotes transposition. In this work, we assume a Bayesian setting where a priori knowledge is available. Specifically, we assume that the vectorized channel $\mathbf{x}$ and vectorized noise $\mathbf{n}$ are independent and circular symmetric complex Gaussian distributed as

$$\mathbf{x} \sim \mathcal{CN}\left(\mathbf{u}_\mathbf{x}, \mathbf{C}_{\mathbf{xx}}\right) \tag{4.4}$$

$$\mathbf{n} \sim \mathcal{CN}\left(\mathbf{u}_\mathbf{n}, \mathbf{C}_{\mathbf{nn}}\right). \tag{4.5}$$

The estimator for $\mathbf{x}$ with minimum mean square error (MMSE), is the mean of the posterior distribution, which is given by [61]

$$\mathbf{u}_\mathbf{x} + \left(\mathbf{C}_{\mathbf{xx}}^{-1} + \mathbf{G}^H\mathbf{C}_{\mathbf{nn}}^{-1}\mathbf{G}\right)^{-1}\mathbf{G}^H\mathbf{C}_{\mathbf{nn}}^{-1}\left(\mathbf{y} - \mathbf{G}\mathbf{u}_\mathbf{x} - \mathbf{u}_\mathbf{n}\right).$$

Here, $(\cdot)^H$ denotes the complex conjugate transpose. The MMSE associated with this estimator is given by the trace of the posterior covariance matrix,

$$\operatorname{Tr}\left(\left(\mathbf{C}_{\mathbf{xx}}^{-1} + \mathbf{G}^H\mathbf{C}_{\mathbf{nn}}^{-1}\mathbf{G}\right)^{-1}\right), \tag{4.6}$$

where $\operatorname{Tr}(\cdot)$ denotes the trace operator. When designing $\mathbf{S}$ in (4.3), our objective is to estimate $\mathbf{x}$ from the observation $\mathbf{y}$ with as small MSE as possible. As constraint, we will impose a total power limitation on the transmitted pilots. Utilizing (4.6), and $\mathbf{G} = \mathbf{S}^T \otimes \mathbf{I}_m$, this optimization problem can be formulated as

$$\min_{\mathbf{S}} \ \operatorname{Tr}\left(\left(\mathbf{C}_{\mathbf{xx}}^{-1} + \left(\mathbf{S}^T \otimes \mathbf{I}_m\right)^H \mathbf{C}_{\mathbf{nn}}^{-1}\left(\mathbf{S}^T \otimes \mathbf{I}_m\right)\right)^{-1}\right) \tag{4.7}$$

$$\text{s.t. } \|\mathbf{S}\|_2^2 \triangleq \operatorname{Tr}\left(\mathbf{S}^H\mathbf{S}\right) \leq \sigma. \tag{4.8}$$

The objective function in (4.7) is MMSE, for a *given* $\mathbf{S}$. The constraint in (4.8) represents an upper bound on the squared Frobenius norm of $\mathbf{S}$.

4.3 Background and Motivation

The literature on pilot design for MIMO channel estimation is rich, because (4.7)-(4.8) is a non convex problem and therefore difficult to optimize without making limiting assumptions. This work offers an alternative approach to those works that assume *Kronecker structure* on $\mathbf{C_{xx}}$ and $\mathbf{C_{nn}}$. The Kronecker structure assumption is the assumption that the covariance matrices in (4.4) and (4.5) factorize as Kronecker products [61]:

$$\mathbf{C_{xx}} = \mathbf{X}_T^T \otimes \mathbf{X}_R \text{ and } \mathbf{C_{nn}} = \mathbf{N}_T^T \otimes \mathbf{N}_R. \tag{4.9}$$

Here, $\mathbf{X}_R$ is the spatial covariance matrix at the receiver, and $\mathbf{X}_T$ is at the transmitter. Similarly, $\mathbf{N}_T$ is the temporal noise covariance matrix, and $\mathbf{N}_R$ is the spatial noise covariance matrix.

Such Kronecker factorizations allow for tractable analysis. Moreover, exploiting the Weichselberger channel model [79], it has been demonstrated in [61] that this assumption may provide good pilots even when the Kronecker structure does not hold. In general, however, assuming Kronecker structure imposes quite severe restrictions on the spatial correlation of the MIMO channel [80]. The main reason is that arbitrary covariance matrices do generally not factorize like this. Therefore, the present work avoids this assumption, and offers an alternative approach.

For smooth optimization problems, as described by (4.7), (4.8), we can arrive at a solution that is at least first order optimal (zero gradient), from an arbitrary initial starting point [81]. The challenge is that our problem is generally not convex in $\mathbf{S}$, and the number of local minima may be large. Therefore, we propose a procedure that most often provides a better starting point than a random one. From this starting point, we proceed iteratively towards a local minimum. Briefly the idea goes as follows. First, we solve a relaxed convex version of the original optimization problem. Next, we project that solution onto the nearest candidate in the feasible set. Finally we move iteratively from the projected solution towards a local minimizer by employing an augmented Lagrangian method. These three steps are described in the next three sections, respectively.

Before proceeding, we mention that our approach does not always produce the best pilot matrix. In some scenarios, the pilot matrix resulting from the Kronecker structure assumption, e.g. [61], may prove better. This should not be considered a problem; we merely provide the designer with an alternative pilot matrix. Equipped with alternatives, the designer can compute the MMSE associated with each alternative, and simply choose the best one. This is valuable, especially when the channel and noise processes are stationary.

4.4 A relaxed convex problem

It is not difficult to generate examples showing that the problem defined by (4.7)-(4.8) is generally not convex in $\mathbf{S}$. Fig. 4.1 illustrates one case, when $\mathbf{S} \in \mathbb{R}^{2\times 2}$, $\|\mathbf{S}\|_2^2 \leq 4$ and

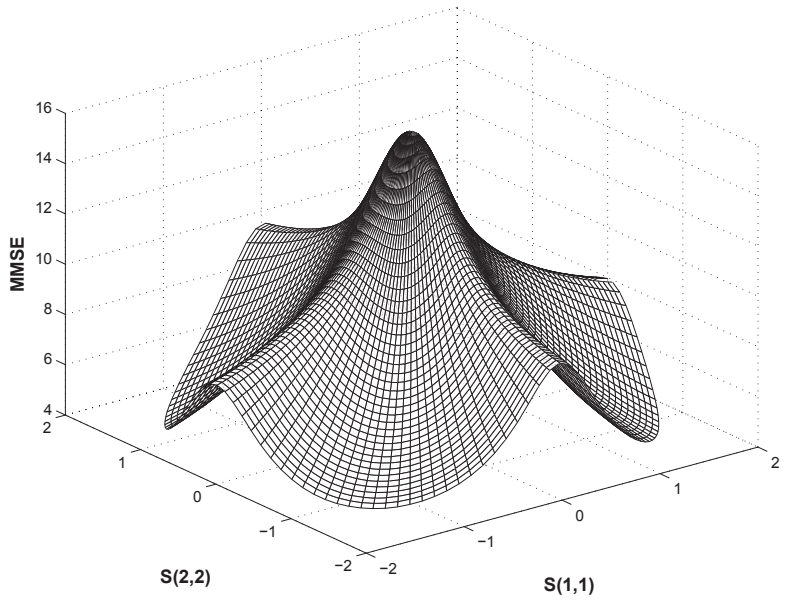

Figure 4.1: Example of MMSE when $\mathbf{S} \in \mathbb{R}^{2\times 2}$, $\|\mathbf{S}\|_2^2 \leq 4$ and $\mathbf{S}$ is restricted to be a diagonal matrix.

$\mathbf{S}$ is restricted to be diagonal. The implication is that we must generally contend with a *local* minimizer. Such a minimizer tends to depend critically on the starting point. This section therefore derives a starting point which in many cases is better than an arbitrary one.

Note from (4.3) that a power constraint on the pilots $\|\mathbf{S}\|_2^2 \leq \sigma$, transfers into a corresponding power constraint $\|\mathbf{G}\|_2^2 \leq \gamma = m\sigma$. If we now consider the latter constraint only, and disregard the fact that any feasible $\mathbf{G} \in \mathbb{C}^{pm\times nm}$ must factorize as $\mathbf{S}^T \otimes \mathbf{I}_m$, we may formulate the following relaxed optimization problem

$$\min_{\mathbf{S}} \ \mathrm{Tr}\left(\left(\mathbf{C}_{\mathbf{xx}}^{-1} + \mathbf{G}^H\mathbf{C}_{\mathbf{nn}}^{-1}\mathbf{G}\right)^{-1}\right) \tag{4.10}$$

$$\text{s.t.} \ \ \mathrm{Tr}\left(\mathbf{G}^H\mathbf{G}\right) \leq \gamma. \tag{4.11}$$

This problem has a convex structure, which will become clear shortly. Its solution, which can be efficiently obtained, must then be projected on to the set of feasible $\mathbf{G}$:s, defined by

$$\mathbb{G} := \left\{\mathbf{G} = \mathbf{S}^T \otimes \mathbf{I}_m, \text{ where } \|\mathbf{S}\|_2^2 \leq \sigma\right\}. \tag{4.12}$$

Finally the result of the projection is treated as a starting point, and updated iteratively towards a local minimum. Note that this approach, in contrast to [61], allows for completely arbitrary covariance matrices.

The remainder of this section presents the solution for the problem (4.10)-(4.11), where $\mathbf{G}$ can have arbitrary structure. Because $\mathbf{G}$ is $pm \times nm$, it follows that $\mathbf{C_{xx}}$ is $nm \times nm$ and that $\mathbf{C_{nn}}$ is $pm \times pm$. We introduce the following singular value decompositions (SVD)

$$\mathbf{C_{xx}} = \mathbf{U_x}\Sigma_\mathbf{x}\mathbf{U_x}^H, \quad \mathbf{C_{nn}^{-1}} = \mathbf{U_n}\Sigma_\mathbf{n}^{-1}\mathbf{U_n}^H, \tag{4.13}$$

with

$$\Sigma_\mathbf{x}(1,1) \geq \Sigma_\mathbf{x}(2,2) \geq \cdots \geq \Sigma_\mathbf{x}(nm,nm) > 0, \tag{4.14}$$

$$\Sigma_\mathbf{n}^{-1}(1,1) \geq \Sigma_\mathbf{n}^{-1}(2,2) \geq \cdots \geq \Sigma_\mathbf{n}^{-1}(pm,pm) > 0\,, \tag{4.15}$$

Throughout, $\mathbf{B}(i,j)$ will denote the element on the i-th row and j-th column of matrix $\mathbf{B}$. In order to rewrite the optimization problem (4.10)-(4.11) in a more convenient form, we now assume that

$$\mathbf{G} = \mathbf{U_n}\mathbf{F}\mathbf{U_x}^H. \tag{4.16}$$

Observe that this introduces no restrictions on $\mathbf{G}$: If $\mathbf{F}$ can be any $pm \times nm$ matrix, then so can $\mathbf{G}$, because both $\mathbf{U_n}$ and $\mathbf{U_x}$ are unitary. Exploiting (4.16), (4.13) and (4.7), it is straightforward to verify that the optimization simplifies to

$$\min_\mathbf{F} \text{Tr}\left(\left(\Sigma_\mathbf{x}^{-1} + \mathbf{F}^H\Sigma_\mathbf{n}^{-1}\mathbf{F}\right)^{-1}\right) \tag{4.17}$$

$$\text{s.t.} \quad \text{Tr}\left(\mathbf{F}^H\mathbf{F}\right) \leq \gamma. \tag{4.18}$$

Applying [61, Lemma 1], it can be shown that the optimal $\mathbf{F}$ is diagonal[2]. If we define the compact notation $\mathbf{F}^H(i,i)\mathbf{F}(i,i) = f_i^2$, $\Sigma_\mathbf{x}^{-1}(i,i) = \sigma_x^{-1}(i)$ and $\Sigma_\mathbf{n}^{-1}(i,i) = \sigma_n^{-1}(i)$ the optimization problem can therefore be written as

$$\min_\mathbf{F} \sum_{i=1}^{nm} \frac{1}{\sigma_x^{-1}(i) + f_i^2\sigma_n^{-1}(i)} \tag{4.19}$$

$$\text{s.t.} \quad \sum_{i=1}^{nm} f_i^2 \leq \gamma. \tag{4.20}$$

This is clearly a convex problem in f_i^2, for which we know the KKT conditions define the optimal solution. The solution can be derived as

$$f_i^2 = \left(0, \sqrt{\frac{\sigma_n(i)}{\alpha}} - \frac{\sigma_n(i)}{\sigma_x(i)}\right)^+, \tag{4.21}$$

where $(0,q)^+$ denotes the maximum of 0 and q, and $\alpha > 0$ is a Lagrange multiplier chosen such that

$$\gamma = \sum_{i=1}^{nm} \left(0, \sqrt{\frac{\sigma_n(i)}{\alpha}} - \frac{\sigma_n(i)}{\sigma_x(i)}\right)^+. \tag{4.22}$$

[2]In fact, it can also be shown that the optimal $\mathbf{F}$ is such that $\mathbf{F}^H\Sigma_\mathbf{n}^{-1}\mathbf{F}$ has decreasingly ordered diagonal elements. We do not have to rely on that property at this point, because it follows naturally.

Observe that both the objective function and the constraint depend on $\mathbf{F}$ only via the squared elements $f_i^2 = \mathbf{F}^H(i,i)\mathbf{F}(i,i)$. This implies that the optimal solution for $\mathbf{F}$ is not unique: any $\mathbf{F}$ satisfying (4.21) and (4.22) is optimal, and we may for instance choose $\mathbf{F}$ to be purely real. Inserting such an $\mathbf{F}$ into (4.16), produces an optimal $\mathbf{G}$ matrix.

Note finally that the solution (4.21) satisfies the constraint (4.20) with equality. The explanation for this is straightforward: The objective function $\mathrm{Tr}\,(\mathbf{W}^{-1})$, as in (4.17), is strictly convex in the eigenvalues of any positive definite matrix $\mathbf{W}$. Hence, for a matrix $\mathbf{F}$ that does not fulfill (4.18) with equality, we can always reduce (4.17) by updating $\mathbf{F}$ to $\eta\mathbf{F}$, where $\eta > 1$ without violating the constraint. The implication is that we need not consider the interior of the constraint region, only its boundary. An entirely similar argument goes of course for the original problem (4.7), and therefore we can conclude that a solution should satisfy (4.8) with equality.

4.5 Projecting onto the feasible set

We cannot expect that an optimal matrix $\mathbf{G}$, as given in the previous section, factorizes as required by (4.12). Moreover, we are actually interested in the underlying $\mathbf{S}$. Since we know that a solution will spend all the available power, a natural approach is to select the $\mathbf{S}$ which solves

$$\min_{\mathbf{S}} \left\|\mathbf{G} - \left(\mathbf{S}^T \otimes \mathbf{I}_m\right)\right\|_2^2 \tag{4.23}$$

$$\text{s.t.} \qquad \mathrm{Tr}\left(\mathbf{S}^H\mathbf{S}\right) = \sigma. \tag{4.24}$$

From the definition of the Kronecker product we have

$$\mathbf{S}^T \otimes \mathbf{I}_m = \begin{bmatrix} \mathbf{S}(1,1)\mathbf{I}_m & \cdots & \mathbf{S}(n,1)\mathbf{I}_m \\ \vdots & \ddots & \vdots \\ \mathbf{S}(1,p)\mathbf{I}_m & \cdots & \mathbf{S}(n,p)\mathbf{I}_m \end{bmatrix}.$$

If we partition $\mathbf{G}$ into a similar block structure, such that

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{1,1} & \cdots & \mathbf{G}_{n,1} \\ \vdots & \ddots & \vdots \\ \mathbf{G}_{1,p} & \cdots & \mathbf{G}_{n,p} \end{bmatrix},$$

where each block $\mathbf{G}_{i,j}$ is $m \times m$, it can be verified that

$$\left\|\mathbf{G} - \left(\mathbf{S}^T \otimes \mathbf{I}_m\right)\right\|_2^2 = \sum_{i=1}^{n}\sum_{j=1}^{p} \left\|\mathbf{G}_{i,j} - \mathbf{S}(i,j)\mathbf{I}_m\right\|_2^2. \tag{4.25}$$

As for the constraint, it can be written

$$\mathrm{Tr}\left(\mathbf{S}^H\mathbf{S}\right) = \sum_{i=1}^{n}\sum_{j=1}^{p} \mathbf{S}(i,j)\mathbf{S}^*(i,j) = \sigma, \tag{4.26}$$

where $(\cdot)^*$ denotes complex conjugation. Utilizing (4.25)-(4.26), the projection is therefore the solution to

$$\min_{\mathbf{S}} \sum_{i=1}^{n} \sum_{j=1}^{p} \|\mathbf{G}_{i,j} - \mathbf{S}(i,j)\mathbf{I}_m\|_2^2$$
$$\text{s.t.} \sum_{i=1}^{n} \sum_{j=1}^{p} \mathbf{S}(i,j)\mathbf{S}^*(i,j) = \sigma.$$

This is a convex problem with convex constraints. The solution can be derived as

$$\mathbf{S}(i,j) = \frac{\text{Tr}\,(\mathbf{G}_{i,j})}{m+\beta}, \tag{4.27}$$

where β is a Lagrange multiplier chosen such that

$$\sum_{i=1}^{n} \sum_{j=1}^{p} \frac{\text{Tr}\,(\mathbf{G}_{i,j})^* \,\text{Tr}\,(\mathbf{G}_{i,j})}{(m+\beta)^2} = \sigma. \tag{4.28}$$

Observe that if $\beta = 0$ satisfies (4.28), we see from (4.27) that $\mathbf{S}(i,j)$ becomes the mean of the diagonal elements of block $\mathbf{G}_{i,j}$.

4.6 Updating to a local minimum

The result of the projection of (4.27)-(4.28) produces a feasible pilot matrix, but that pilot matrix is in general not even a first order optimal solution to the original problem (4.7)-(4.8). Therefore it should be treated as a starting point for subsequent optimization. To that end, we will move from this starting point towards a local optimum using an augmented Lagrangian method. The latter is also known as the *method of multipliers*. The core idea is to replace a constrained problem by a sequence of unconstrained problems. A good introduction to this method, along with algorithms for its implementation, can be found in [81]. Therefore we do not present the full details of the method here, but rather focus on some key ingredients.

Let the objective function in (4.7) be denoted by $g(\mathbf{S})$. Because we know that a solution will spend all the available power, we substitute the inequality contraint (4.8) by the equality constraint $c(\mathbf{S}) = \sigma - \text{Tr}\left(\mathbf{S}^H\mathbf{S}\right) = 0$. The augmented Lagrangian function is then given by

$$\mathcal{L}\left(\mathbf{S}, \lambda, \mu\right) = g(\mathbf{S}) - \lambda c(\mathbf{S}) + \frac{1}{2\mu} c^2(\mathbf{S}), \tag{4.29}$$

where λ is a Lagrange multiplier and μ is a penalty parameter. The derivative of this function with respect to $\mathbf{S}$ is

$$\nabla_{\mathbf{s}} \mathcal{L}\left(\mathbf{S}, \lambda, \mu\right) = \nabla_{\mathbf{s}} g(\mathbf{S}) - \left(\lambda - \frac{c(\mathbf{S})}{\mu}\right) \nabla_{\mathbf{s}} c(\mathbf{S}). \tag{4.30}$$

Because $\nabla_{\mathbf{s}} g(\mathbf{S})$ and $\nabla_{\mathbf{s}} c(\mathbf{S})$ are key elements in the augmented Lagrangian method, we derive them next.

4.6.1 The gradient of the objective and the constraint

The objective function can be expressed as $\mathrm{Tr}\left(\mathbf{W}^{-1}\right)$, where

$$\mathbf{W} = \mathbf{C}_{\mathbf{xx}}^{-1} + \left(\mathbf{S}^T \otimes \mathbf{I}_m\right)^H \mathbf{C}_{\mathbf{nn}}^{-1} \left(\mathbf{S}^T \otimes \mathbf{I}_m\right). \tag{4.31}$$

In order to find the derivative of the objective function w.r.t. $\mathbf{S}$, it is convenient to take the approach suggested in [78]. That is, to first identify the differential, and then use this to obtain the derivative. Without displaying the preceding steps here, the gradient of the objective function w.r.t. $\mathbf{S}$, expressed as a $1 \times np$ row vector is:

$$-\mathrm{vec}^T\left(\mathbf{S}^* \otimes \mathbf{I}_m\right) \left(\mathbf{C}_{\mathbf{nn}}^{-1} \otimes \mathbf{W}^{-1}\mathbf{W}^{-1}\right) \left(\mathbf{I}_p \otimes \mathbf{R}\right), \tag{4.32}$$

where

$$\mathbf{R} = \left(\mathbf{K}_{m,n} \otimes \mathbf{I}_m\right) \left(\mathbf{I}_n \otimes \mathrm{vec}\left(\mathbf{I}_m\right)\right), \tag{4.33}$$

and $\mathbf{K}_{m,n}$ is the *commutation* matrix [78, Definition 2.9]. In order to obtain $\nabla_{\mathbf{s}} g(\mathbf{S})$, we split the row vector (4.32) into p equally long sub vectors and take these as the columns of $\nabla_{\mathbf{s}} g(\mathbf{S})$. The derivative of the constraint w.r.t. $\mathbf{S}$ is simply

$$\nabla c_{\mathbf{s}}(\mathbf{S}) = \mathbf{S}^*. \tag{4.34}$$

For space reasons, we do not present the full algorithmic framework of the augmented Lagrangian method here. Instead we refer the reader to [81, Framework 17.3]. With the gradients given in (4.32) and (4.34), the algorithm is straightforward to implement.

4.7 Numerical results

This section compares experimentally the performance of our method with that of [61, Heuristic 1], for a particular class of noise and channel covariance matrices, which we describe shortly. The augmented Lagrangian method is implemented as described in [81, Framework 17.3], using the following parameters:

$$\mu_k = \tau_k = \frac{1}{k}.$$

As initial values, we select $\lambda_0 = \mu_0 = \tau_0 = 1$. We assume a case where all matrices in (4.2) are 2×2. Consequently, $\mathbf{C}_{\mathbf{xx}}$ and $\mathbf{C}_{\mathbf{nn}}$ are 4×4. We study the average MMSE over 500 different scenarios, each in which $\mathbf{C}_{\mathbf{xx}}$ and $\mathbf{C}_{\mathbf{nn}}$ are generated randomly. For each scenario, the covariance matrices are generated as follows. Let $\mathbf{A}$ be a realization of a

	$\mathbf{S}$	$\mathbf{S}_{rand}$	$\mathbf{S}_{kron}$
Winner rate	0.5080	0.3160	0.1760
Average normalized MMSE	0.0657	0.0674	0.0748

Table 4.1: Performance for a specific class of covariance matrices.

4×4 matrix with i.i.d. elements $\mathcal{N}(0,1)$. Compute $\mathbf{C_{xx}} = \text{abs}(\mathbf{A}^T)\text{abs}(\mathbf{A})$, where the abs$(\cdot)$ operator turns the sign of the negative elements. $\mathbf{C_{nn}}$ is generated independently in the same manner. Observe that these matrices are symmetric, and positive definite with probability one. We assume that $\sigma = 4$ in (4.8).

Table 4.1 summarizes the results. $\mathbf{S}$, $\mathbf{S}_{kron}$ and $\mathbf{S}_{rand}$ denote the pilot matrices that result from our method, from [61, Heuristic 1], and from an augmented Lagrangian method with *random* starting point, respectively. The 'winner rate' represents the share of scenarios where a method outperforms the two other methods. The normalized MMSE is defined as

$$\frac{\text{Tr}\left(\left(\mathbf{C}_{\mathbf{xx}}^{-1} + \left(\mathbf{S}^T \otimes \mathbf{I}_m\right)^H \mathbf{C}_{\mathbf{nn}}^{-1} \left(\mathbf{S}^T \otimes \mathbf{I}_m\right)\right)^{-1}\right)}{\text{Tr}\left(\mathbf{C}_{\mathbf{xx}}\right)}.$$

This is just the standard MMSE normalized with the power of the channel that we wish to estimate.

Table 4.1 indicates that, for covariance matrices generated as described, the proposed method is better than that of [61, Heuristic 1] on average. In fact, for this setup, an augmented Lagrangian method with random starting point is also better than [61, Heuristic 1] on average. These observations underline that pilot matrices based on the Kronecker structure assumption should be used with some care, and that other alternatives could be worth exploring. Observe also that our method is better than using a random starting point on average. The above example focuses on the average performance over a class of scenarios. For a single scenario, and especially if the settings are stationary, one can evaluate several alternative pilot matrices and select the best one. We have observed, in such cases, that the difference between the different methods may be substantial.

An attractive feature of the Kronecker structure assumption, is that it allows for a closed form solution. It may not always turn out to be the best solution, but it can be derived with very limited complexity. In contrast, the augmented Lagrangian method is based on an iterative algorithm. The speed of convergence depends on the initial values and on how one updates the parameters, but it will invariably require much more computations. Under stationary or slowly varying statistics, that effort may still pay off.

4.8 Conclusion

We have described a procedure which obtains a pilot matrix for MIMO channel estimation when the structure on the underlying covariance matrices is arbitrary. In particular, we do not rely on Kronecker structure. The procedure is based on a convex relaxation of the original problem. Its solution is projected onto the feasible set, and used as starting point for an augmented Lagrangian method. Numerical experiments indicate that this procedure may produce pilot signals that are better than those obtained under the Kronecker structure assumption.

Bibliography

[1] "Report of the Spectrum Efficiency Working Group." FCC, Tech. Rep., Nov. 2002.

[2] S. Sankaranarayanan, P. Papadimitratos, and A. Mishra, "Enhancing Wireless Spectrum Utilization with a Cellular ad-hoc Overlay Architecture," in *Military Communications Conference, 2005. MILCOM 2005. IEEE*, oct. 2005, pp. 405 –412 Vol. 1.

[3] V. Valenta, R. Marsalek, G. Baudoin, M. Villegas, M. Suarez, and F. Robert, "Survey on Spectrum Utilization in Europe: Measurements, Analyses and Observations," in *Cognitive Radio Oriented Wireless Networks Communications (CROWNCOM), 2010 Proceedings of the Fifth International Conference on*, june 2010, pp. 1 –5.

[4] M. Islam, C. Koh, S. Oh, X. Qing, Y. Lai, C. Wang, Y.-C. Liang, B. Toh, F. Chin, G. Tan, and W. Toh, "Spectrum Survey in Singapore: Occupancy Measurements and Analyses," in *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on*, may 2008, pp. 1 –7.

[5] I. Mitola, J. and J. Maguire, G.Q., "Cognitive Radio: Making Software Radios more Personal," *Personal Communications, IEEE*, vol. 6, no. 4, pp. 13 –18, aug 1999.

[6] G. Karlsson, S. Lindfors, M. Skoglund, and G. Øien, "CROSS-LAYER OPTIMIZATION IN SHORT-RANGE WIRELESS SENSOR NETWORKS, PART 2 (CROPS2) - APPLICATION TO SPECTRUM SENSING," *Project Proposal*, june 2007.

[7] Y.-C. Liang, K.-C. Chen, G. Li, and P. Mahonen, "Cognitive Radio Networking and Communications: An Overview," *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 7, pp. 3386 –3407, sept. 2011.

[8] A. Sahai, N. Hoven, and R. Tandra, "Some Fundamental Limits on Cognitive Radio," in *the 42sd Allerton Conference on Communication, Control and Computing*, 2004.

[9] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. New York, NY, USA: Cambridge University Press, 2005.

[10] G. Yu, G. Sapiro, and S. Mallat, "Solving Inverse Problems With Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity," *Image Processing, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2011.

[11] Y. Ephraim and H. Van Trees, " Signal Subspace Approach for Speech Enhancement," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 4, pp. 251 –266, jul 1995.

[12] J. Q. Li and A. R. Barron, "Mixture Density Estimation," in *In Advances in Neural Information Processing Systems 12*. MIT Press, 1999, pp. 279–285.

[13] H. Sorenson and D. Alspach, "Recursive Bayesian Estimation Using Gaussian Sums," *Automatica*, vol. 7, no. 4, pp. 465 – 479, 1971.

[14] H. El Ghannudi, L. Clavier, N. Azzaoui, F. Septier, and P.-A. Rolland, "α-stable Interference Modeling and Cauchy Receiver for an IR-UWB Ad Hoc Network," *Communications, IEEE Transactions on*, vol. 58, no. 6, pp. 1748 –1757, june 2010.

[15] M. Afify, X. Cui, and Y. Gao, "Stereo-Based Stochastic Mapping for Robust Speech Recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 7, pp. 1325 –1334, sept. 2009.

[16] X. Cui, M. Afify, and Y. Gao, "MMSE-based Stereo Feature Stochastic Mapping for Noise Robust Speech Recognition," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-april 4 2008, pp. 4077 –4080.

[17] L. Buera, E. Lleida, A. Miguel, A. Ortega, and O. Saz, "Cepstral Vector Normalization Based on Stereo Data for Robust Speech Recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 3, pp. 1098 –1113, march 2007.

[18] J. Samuelsson and P. Hedelin, "Recursive Coding of Spectrum Parameters," *Speech and Audio Processing, IEEE Transactions on*, vol. 9, no. 5, pp. 492 –503, jul 2001.

[19] D. Persson and T. Eriksson, "Mixture Model- and Least Squares-Based Packet Video Error Concealment," *Image Processing, IEEE Transactions on*, vol. 18, no. 5, pp. 1048 –1054, may 2009.

[20] A. Kundu, S. Chatterjee, and T. Sreenivas, "Subspace Based Speech Enhancement Using Gaussian Mixture Model," in *Interspeech 2008*, Brisbane, Australia, september 2008, pp. 395 –398.

[21] ——, "Speech Enhancement using Intra-Frame Dependency in DCT Domain," in *16th European Signal Processing Conference (EUSIPCO 2008)*, Lausanne, Switzerland, August 25-29 2008.

[22] A. Subramaniam, W. Gardner, and B. Rao, "Low-Complexity Source Coding Using Gaussian Mixture Models, Lattice Vector Quantization, and Recursive Coding

with Application to Speech Spectrum Quantization," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 2, pp. 524 – 532, march 2006.

[23] L. Deng, J. Droppo, and A. Acero, "Estimating Cepstrum of Speech Under the Presence of Noise Using a Joint Prior of Static and Dynamic Features," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 3, pp. 218 – 233, may 2004.

[24] V. Stouten, H. V. hamme, and P. Wambacq, "Model-based Feature Enhancement with Uncertainty Decoding for Noise Robust ASR," *Speech Communication*, vol. 48, no. 11, pp. 1502 – 1514, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639306000057

[25] W. Kim and J. H. Hansen, "Feature Compensation in the Cepstral Domain employing Model Combination," *Speech Communication*, vol. 51, no. 2, pp. 83 – 96, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639308000927

[26] J. Gonzandlez, A. Peinado, A. Gandmez, and J. Carmona, "Efficient MMSE Estimation and Uncertainty Processing for Multienvironment Robust Speech Recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 5, pp. 1206 –1220, july 2011.

[27] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *Automatic Control, IEEE Transactions on*, vol. 17, no. 4, pp. 439 – 448, aug 1972.

[28] I. Bilik and J. Tabrikian, "MMSE-Based Filtering in Presence of Non-Gaussian System and Measurement Noise," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 46, no. 3, pp. 1153 –1170, july 2010.

[29] ——, "Maneuvering Target Tracking in the Presence of Glint using the Nonlinear Gaussian Mixture Kalman Filter," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 46, no. 1, pp. 246 –262, jan. 2010.

[30] J. Kotecha and P. Djuric, "Gaussian Sum Particle Filtering," *Signal Processing, IEEE Transactions on*, vol. 51, no. 10, pp. 2602 – 2612, oct. 2003.

[31] G. Ackerson and K. Fu, "On State Estimation in Switching Environments," *Automatic Control, IEEE Transactions on*, vol. 15, no. 1, pp. 10 – 17, feb 1970.

[32] J. Tugnait and A. Haddad, "Adaptive Estimation in Linear Systems with Unknown Markovian Noise Statistics," *Information Theory, IEEE Transactions on*, vol. 26, no. 1, pp. 66 – 78, jan 1980.

[33] H. L. V. Trees, *Detection, Estimation, and Modulation Theory, Part 1*. John Wiley and Sons, 1968.

[34] B. Bobrovsky and M. Zakai, "A Lower Bound on the Estimation Error for Certain Diffusion Processes," *Information Theory, IEEE Transactions on*, vol. 22, no. 1, pp. 45 – 52, jan 1976.

[35] A. Weiss and E. Weinstein, "A Lower Bound on the Mean-Square Error in Random Parameter Estimation (Corresp.)," *Information Theory, IEEE Transactions on*, vol. 31, no. 5, pp. 680 – 682, sep 1985.

[36] I. Reuven and H. Messer, "A Barankin-Type Lower Bound on the Estimation Error of a Hybrid Parameter Vector," *Information Theory, IEEE Transactions on*, vol. 43, no. 3, pp. 1084 –1093, may 1997.

[37] T. Anderson, *An Introduction to Multivariate Statistical Analysis*, 2nd ed. New York [u.a.]: Wiley, 1984.

[38] M. Carreira-Perpinan, "Mode-Finding for Mixtures of Gaussian Distributions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1318 – 1323, Nov. 2000.

[39] A. Kundu, S. Chatterjee, A. Sreenivasa Murthy, and T. Sreenivas, "GMM based Bayesian approach to Speech Enhancement in Signal / Transform domain," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-april 4 2008, pp. 4893 –4896.

[40] J. T. Flåm, J. Jaldén, and S. Chatterjee, "Gaussian Mixture Modeling for Source Localization," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, may 2011, pp. 2604 –2607.

[41] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[42] M. Gudmundson, "Correlation Model for Shadow Fading in Mobile Radio Systems," *Electronics Letters*, vol. 27, no. 23, pp. 2145 –2146, nov. 1991.

[43] N. Patwari, J. Ash, S. Kyperountas, I. Hero, A.O., R. Moses, and N. Correal, "Locating the Nodes: Cooperative Localization in Wireless Sensor Networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 54 – 69, july 2005.

[44] J. Flåm, D. Ryan, and G. Kraidy, "Using a Sensor Network to Localize a Source under Spatially Correlated Shadowing," *Vehicular Technology Conference, 2010 IEEE 71st*, may. 2010.

[45] C. Morelli, M. Nicoli, V. Rampa, U. Spagnolini, and C. Alippi, "Particle Filters for RSS-Based Localization in Wireless Sensor Networks: An Experimental Study," in *ICASSP 2006*, vol. 4, may 2006.

[46] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel-Based Positioning in Wireless Local Area Networks," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 6, pp. 689 –705, june 2007.

[47] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press, 1995.

[48] R. Jain, “Model T: An Empirical Model for User Registration Patterns in a Campus Wireless LAN,” in *In Proc. MobiCom*, 2005, pp. 170–184.

[49] Z. Ghahramani, “Solving Inverse Problems using an EM approach to Density Estimation,” in *Proceedings of the 1993 Connectionist Models Summer School*, 1993, pp. 316–323.

[50] D. J. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. [Online]. Available: http://www.inference.phy.cam.ac.uk/itprnn/book.pdf

[51] J. Chen and A. Khalili, “Order Selection in Finite Mixture Models with a Nonsmooth Penalty,” *Journal of the American Statistical Association*, pp. 1674 –1683, dec. 2008.

[52] J. Aspnes, T. Eren, D. Goldenberg, A. Morse, W. Whiteley, Y. Yang, B. Anderson, and P. Belhumeur, “A Theory of Network Localization,” *Mobile Computing, IEEE Transactions on*, vol. 5, no. 12, pp. 1663 –1678, dec. 2006.

[53] S. Dasgupta, “Learning Mixtures of Gaussians,” in *Foundations of Computer Science, 1999. 40th Annual Symposium on*, 1999, pp. 634 –644.

[54] M. P. Perrone and L. N. Cooper, “When Networks Disagree: Ensemble Methods for Hybrid Neural Networks,” in *Artificial Neural Networks for Speech and Vision*. Chapman and Hall, 1993, pp. 126–142.

[55] D. Ormoneit and V. Tresp, “Averaging, Maximum Penalized Likelihood and Bayesian Estimation for Improving Gaussian Mixture Probability Density Estimates,” *Neural Networks, IEEE Transactions on*, vol. 9, no. 4, pp. 639 –650, jul 1998.

[56] K. Todros and J. Tabrikian, “General Classes of Performance Lower Bounds for Parameter Estimation; Part II: Bayesian Bounds,” *Information Theory, IEEE Transactions on*, vol. 56, no. 10, pp. 5064 –5082, oct. 2010.

[57] H. Robbins and S. Monro, “A Stochastic Approximation Method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. pp. 400–407, 1951. [Online]. Available: http://www.jstor.org/stable/2236626

[58] H. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer Verlag, 2003, vol. 35.

[59] D. Katselis, E. Kofidis, and S. Theodoridis, “On Training Optimization for Estimation of Correlated MIMO Channels in the Presence of Multiuser Interference,” *Signal Processing, IEEE Transactions on*, vol. 56, no. 10, pp. 4892 –4904, oct. 2008.

[60] M. Biguesh and A. Gershman, “Training-based MIMO Channel Estimation: a Study of Estimator Tradeoffs and Optimal Training Signals,” *Signal Processing, IEEE Transactions on*, vol. 54, no. 3, pp. 884 – 893, march 2006.

[61] E. Björnson and B. Ottersten, "A Framework for Training-Based Estimation in Arbitrarily Correlated Rician MIMO Channels With Rician Disturbance," *Signal Processing, IEEE Transactions on*, vol. 58, no. 3, pp. 1807 –1820, march 2010.

[62] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, 2nd ed. John Wiley & Sons, 1999.

[63] J. Medbo, H. Asplund, J. Berg, and N. Jaldén, "Directional Channel Characteristics in Elevation and Azimuth at an Urban Macrocell Base Station," in *6th European Conference on Antennas and Propagation*, March 2012.

[64] V. Cellini and G. Dona, "A Novel Joint Channel and Multi-user Interference Statistics Estimator for UWB-IR Based on Gaussian Mixture Model," in *Ultra-Wideband, 2005. ICU 2005. 2005 IEEE International Conference on*, sept. 2005, pp. 655 – 660.

[65] G. Healey and R. Kondepudy, "Radiometric CCD Camera Calibration and Noise Estimation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 3, pp. 267 –276, mar 1994.

[66] H. S. Wang and N. Moayeri, "Finite-State Markov Channel-a Useful Model for Radio Communication Channels," *Vehicular Technology, IEEE Transactions on*, vol. 44, no. 1, pp. 163 –171, feb 1995.

[67] P. Sadeghi, R. Kennedy, P. Rapajic, and R. Shams, "Finite-State Markov Modeling of Fading Channels - a Survey of Principles and Applications," *Signal Processing Magazine, IEEE*, vol. 25, no. 5, pp. 57 –80, september 2008.

[68] A. Scaglione, P. Stoica, S. Barbarossa, G. Giannakis, and H. Sampath, "Optimal Designs for Space-Time Linear Precoders and Decoders," *Signal Processing, IEEE Transactions on*, vol. 50, no. 5, pp. 1051 –1064, may 2002.

[69] D. Pham, H. Tuan, B.-N. Vo, and T. Nguyen, "Jointly Optimal Precoding/Postcoding for Colored MIMO Systems," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 4, may 2006, p. IV.

[70] F. Dietrich and W. Utschick, "Pilot-Assisted Channel Estimation based on Second-Order Statistics," *Signal Processing, IEEE Transactions on*, vol. 53, no. 3, pp. 1178 – 1193, march 2005.

[71] J. Kotecha and A. Sayeed, "Transmit Signal Design for Optimal Estimation of Correlated MIMO Channels," *Signal Processing, IEEE Transactions on*, vol. 52, no. 2, pp. 546 – 557, feb. 2004.

[72] Y. Liu, T. Wong, and W. Hager, "Training Signal Design for Estimation of Correlated MIMO Channels With Colored Interference," *Signal Processing, IEEE Transactions on*, vol. 55, no. 4, pp. 1486 –1497, april 2007.

[73] J. Flåm, S. Chatterjee, K. Kansanen, and T. Ekman, "On MMSE Estimation - A Linear Model under Gaussian Mixture Statistics," *IEEE Transactions on Signal Processing*, 2012.

[74] D. Love and R. Heath, "Limited Feedback Unitary Precoding for Spatial Multiplexing Systems," *Information Theory, IEEE Transactions on*, vol. 51, no. 8, pp. 2967 –2976, aug. 2005.

[75] I. Kim, S. Park, D. Love, and S. Kim, "Improved Multiuser MIMO Unitary Precoding using Partial Channel State Information and Insights from the Riemannian Manifold," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 8, pp. 4014 –4023, august 2009.

[76] J. Kiefer and J. Wolfowitz, "Stochastic Estimation of the Maximum of a Regression Function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. pp. 462–466, September 1952. [Online]. Available: http://www.jstor.org/stable/2236690

[77] J. Flåm, M. Vehkaperä, D. Zachariah, and E. Tsakonas, "Mean Square Error Reduction by Precoding of Mixed Gaussian Input," in *International Symposium on Information Theory and its Applications, ISITA 2012*, Honolulu - Hawaii, Oct 28-Nov 01, 2012.

[78] A. Hjørungnes, *Complex-Valued Matrix Derivatives : with Applications in Signal Processing and Communications*. Cambridge: Cambridge University Press, 2011. [Online]. Available: http://opac.inria.fr/record=b1132562

[79] W. Weichselberger, M. Herdin, H. Ozcelik, and E. Bonek, "A Stochastic MIMO Channel Model with Joint Correlation of Both Link Ends," *Wireless Communications, IEEE Transactions on*, vol. 5, no. 1, pp. 90 – 100, jan. 2006.

[80] C. Oestges, "Validity of the Kronecker Model for MIMO Correlated Channels," in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, vol. 6, may 2006, pp. 2818 –2822.

[81] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., ser. Springer series in operations research and financial engineering. New York, NY: Springer, 2006.