January 01, 2009

Topics in nonlinear and robust estimation theory

Roni Mittelman

Northeastern University

Topics in Nonlinear and Robust Estimation Theory

A Dissertation Presented

by

Roni Mittelman

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Electrical Engineering

Northeastern University
Boston, Massachusetts

August 12, 2009

Northeastern University
Graduate School of Engineering

Dissertation title: Topic in Nonlinear and Robust Estimation Theory

Author: Roni Mittelman

Department: Electrical and Computer Engineering

Approved for Dissertation Requirement of Doctor of Philosophy Degree:

Thesis Advisor: Prof. Eric Miller **Date**

Thesis Committee Member: Prof. Hanoch Lev-Ari **Date**

Thesis Committee Member: Prof. Gilead Tadmor **Date**

Department Chair: Prof. Ali Abur **Date**

Graduate School Notified of Acceptance:

Director of Graduate School: Prof. Yaman Yener **Date**

Abstract

We propose new methods to improve nonlinear filtering and robust estimation algorithms. In the first part of the dissertation, we propose an approach to approximating the Chapman-Kolmogorov equation (CKE) for particle-based nonlinear filtering algorithms, using a new proposal distribution and the improved Fast Gauss Transform (IFGT). The new proposal distribution, used to obtain a Monte Carlo (MC) approximation of the CKE, is based on the proposal distribution found in the auxiliary marginal particle filter (AMPF). By using MC integration to approximate the integrals of the AMPF proposal distribution as well as the CKE, we demonstrate significant improvement in terms of both error and computation time. We consider the additive state noise case where the evaluation of the CKE is equivalent to performing kernel density estimation (KDE), thus fast methods such as the IFGT can be used. We also provide much improved performance bounds for the IFGT, and which unlike the previous bound, are consistent with the expectation that the error decreases as the truncation order of the IFGT increases. The experimental results show that we can obtain similar error to the sequential importance sampling (SIS) particle filter, while using fewer particles.

In the second part, we consider the problem of estimating a Gaussian random parameter vector that is observed through a linear transformation with added white Gaussian noise when there are both eigenvalue and elementwise uncertainties in the covariance matrix. When the covariance matrix is known then the solution to the problem is given by the minimum mean squared error (MMSE) estimator. Recently a minimax approach in which the estimator is chosen to minimize the worst case of two criteria called the difference regret [28] and the ratio regret [29] in an eigenvalue uncertainty region was proposed. A closed form solution was also presented under the assumption that the Gram matrix of the model matrix weighted by the inverse covariance matrix of the noise vector, and the random parameter's covariance matrix, are diagonalized by the same unitary matrix (we refer to this as the *joint diagonalizability assumption*). This assumption significantly limits the applicability of the estimator. In this work we present a new criterion for the minimax estimation problem which we call the generalized difference regret (GDR), and derive the minimax estimator which is based on the GDR criterion where the region of uncertainty is defined not only using upper and lower bounds on the eigenvalues of the parameter's covariance matrix, but also using upper and lower bounds on the individual elements of the covariance matrix itself. Furthermore the GDR estimator does not require the assumption of joint diagonalizability and it can be obtained efficiently using semidefinite programming. The experimental results show that we can obtain improved mean squared error (MSE) results compared to the MMSE estimator and the difference and ratio regret estimators.

Finally, we propose a new approach for robust parameter estimation under sensor positional uncertainty of parameters which are used as features for an unexploded ordnance (UXO) classification scheme. Obtaining better parameter estimates by addressing the uncertainty that may be present in the locations of the sensors, is important in order to obtain improved classification results. Future work will include simulation and validation of the new approach with a UXO classification scheme.

Acknowledgements

I would like to express my gratitude to my advisor Prof. Eric Miller. His determination in research, and his vast interest and knowledge in many different fields are rarely matched in a single individual. It is my hope that I was able to acquire some of his curiosity and passion for research through this Ph.D. work.

I would also like to express my gratitude to the two other members of my thesis committee, Prof. Hanoch Lev-Ari and Prof. Gilead Tadmor. Their support and words of advice throughout the four years that I spent at Northeastern Univsersity have meant a lot to me, and encouraged me to complete this dissertation.

Finally, I would like to thank my parents Zoltan and Iehudit Mittelman, and my brother Yehuda Mittelman. This dissertation is dedicated to them.

Contents

1 Introduction 1

2 Background- Fast Kernel Density Estimation and Nonlinear Filtering 5

2.1 Fast KDE 5

2.1.1 Discrete approximation techniques 6

2.1.2 The fast Gauss transform 7

2.1.3 The improved fast Gauss transform 10

2.2 Filtering 12

2.2.1 The Kalman filter 13

2.2.2 Deterministic nonlinear filtering 14

2.2.3 Particle filtering 15

3 Background- Convex Optimization, Minimax Estimation, and Sensor Networks 21

3.1 Convex optimization 22

3.1.1 Sets 22

3.1.2 Convex sets 23

3.1.3 Convex functions 24

3.1.4 Convex optimization problems 25

3.1.5 Duality 28

3.2 Minimax theory 29

3.3 Estimation with covariance matrix uncertainties 30

3.3.1 Minimax regret estimators 31

3.4 Sensor networks 34

3.4.1 Gaussian Markov random fields 35

3.4.2 Gaussian processes 36

4 Tighter IFGT Performance Bounds and Nonlinear Filtering using the IFGT 39

4.1 New error bounds for the IFGT 39

4.2 Analysis of the new error bounds and choosing the IFGT parameters using the new bounds 43

4.2.1 Analysis of the error bounds 43

4.2.2 Choosing the IFGT parameters using the new bounds 45

4.3 Nonlinear filtering using a new proposal distribution and the IFGT 46

4.3.1 The new filtering algorithm 46

4.3.2 Experimental results 47

4.3.3 Choosing r_0 vs. choosing the number of clusters 50

4.4 Conclusion 52

5 Robust Minimax Estimation in a Linear Gaussian Model 53

5.1 Minimax estimation with joint eigenvalue and elementwise covariance uncertainties using the GDR criterion 54

5.2 Relationship of the GDR to the difference regret estimator 59

5.2.1 Equivalence of the GDR estimator with eigenvalue alone uncertainties to the difference regret estimator 59

5.2.2 The GDR as a lower bound on the MSE obtained using the MMSE estimator 60

5.3 Example of the GDR estimator 62

5.3.1 Estimating the uncertainty region 62

5.3.2 Experimental results 65

5.4 Robust estimation in a sensor network 67

5.5 Conclusion 70

6 Robust Parameter Estimation with Sensor Positional Uncertainty 71

6.1 The measurement model and nonlinear least squares parameter estimation . 72

6.2 Previous work on parameter estimation with positional uncertainty with the dipole model 73

6.3 A nonlinear programming approach to robust parameter estimation with sensor positional uncertainty 75

7 Conclusions and Future Work 79

Bibliography **81**

List of Figures

2.1 An example of the space partitioning obtained using a kd-tree 7

2.2 An example of the kd tree for the space partitioning in Figure 2.1 8

3.1 An example of the interpretation of the conjugate function 26

4.1 Empirical upper error bound and and upper error bounds (4.1) and (2.17) vs. truncation order for source points generated from a zero mean Gaussian with covariance matrix $0.4I$ 44

4.2 ϵ_p for the $IFGT_2$ and $IFGT_3$ vs. r_0, for $\kappa = 1$, for truncation orders 5, and 8. 45

4.3 The sensor network, the real and the estimated tracks 51

5.1 MSE vs. the SNR for the GDR estimator, difference and ratio regret estimators, and the MMSE estimator matched to the estimated covariance, for $\boldsymbol{H} = \boldsymbol{I}$ 65

5.2 MSE vs. SNR for the GDR estimator and for the MMSE estimator matched to the estimated covariance, with $\boldsymbol{H}$ in a toeplitz form 66

5.3 MSE vs. SNR for the GDR estimator and for the MMSE estimator matched to the estimated covariance, with $\boldsymbol{H}$ in a diagonal form 67

5.4 MSE vs. SNR for different estimators 69

List of Tables

3.1 Different covariance functions 38

4.1 Time [sec], and RMSE when using the AMPF and AMPF-IS proposal distributions with direct evaluation and IFGT, and when using the SIS particle filter 48

4.2 Time [sec], and RMSE for different values of r_0 and p when using the AMPF-IS proposal distribution with the IFGT 49

4.3 Time [sec], and RMSE when using the AMPF and AMPF-IS proposal distributions with direct evaluation and IFGT, and when using the SIS particle filter 50

4.4 Time [sec], and RMSE when using the AMPF-IS proposal distribution and IFGT for the example in Example 1 for different values of r_0 and K 51

4.5 Time [sec], and RMSE when using the AMPF-IS proposal distribution and IFGT for the example in Example 2 for different values of r_0 and K 51

Chapter 1

Introduction

Many areas of signal processing require the estimation of some parameter from a set of observations. When the parameter takes values from a discrete and finite set then the problem is called hypothesis testing, and the area of research is known as detection theory. When the parameter takes values from a continuous set then the area of research is known as estimation theory. As explained in [39] a model for estimation has four components:

- *Parameter Space.* The domain of the parameter that we wish to estimate, this could be $\mathbb{R}^n$ for an n dimensional parameter vector.

- *Probabilistic Mapping from the Parameter Space to the Observation Space.* The parameter is related to the observation through this probability distribution.

- *Observation Space.* The domain of the observation, this could be $\mathbb{R}^m$ for an m dimensional observation vector.

- *Estimation Rule.* The estimate of the parameter is obtained using the estimation rule which is a mapping from the observation space into the parameter space. The estimation rule is designed such that it satisfies some optimality criteria.

Most estimation methods are characterized as either maximum likelihood or Bayesian estimation. In maximum likelihood estimation (ML) it is assumed that the parameter is completely unknown, i.e. there is no apriori information about its distribution. On the other hand in Bayesian estimation it is assumed that the parameter is a realization of some probability distribution. One type of a Bayesian estimator is the maximum aposteriori (MAP) estimator which maximizes the posterior distribution.

One example of an estimation model that is often used in practice is the linear model of the form $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w}$, where a vector $\boldsymbol{x}$ is observed through a linear transformation $\boldsymbol{H}$ and corrupted by additive Gaussian noise $\boldsymbol{w}$. When there is no statistical information about the vector $\boldsymbol{x}$ then the ML estimator is equivalent to the least squares estimate, therefore the estimate $\hat{\boldsymbol{x}}$ is optimal in the sense that it minimizes $\mathbb{E}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2$, where $\boldsymbol{y}$ is the observation vector, and $\hat{\boldsymbol{y}} = \boldsymbol{H}\hat{\boldsymbol{x}}$ is the estimated observation vector based on the estimated vector $\hat{\boldsymbol{x}}$. If in addition we assume that $\boldsymbol{x}$ is a Gaussian random vector then the model is known as the linear Gaussian model and it can be shown that the MAP approach also minimizes the MSE of the estimate of $\boldsymbol{x}$, i.e. $\mathbb{E}\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2$. A particular case of the linear Gaussian model is when the vector $\boldsymbol{x}$ evolves temporally in the form of a linear dynamical system with Gaussian additive noise. The vector $\boldsymbol{x}$ is known as the state and the Bayes estimate at each time instant can be obtained recursively using the Kalman filter [1]. The problem is also known as linear filtering. The Kalman filter is optimal in the sense that it minimizes the mean squared error of the estimate at each time step.

In practice some of the assumptions in the aforementioned estimation models may be invalid. For example, in the linear Gaussian model the matrix $\boldsymbol{H}$ or the covariance matrix of $\boldsymbol{x}$ may only be known to lie in some region of uncertainty, thus the least squares and MMSE estimators are not optimal. The robust estimation approach optimizes a criteria that is less sensitive to such uncertainties in the estimation model. In the linear filtering example the true dynamics are often nonlinear and therefore the Kalman filter is no longer optimal, therefore leading to the use of nonlinear filters which approximate the optimal filter and perform better than the Kalman filter under such conditions.

One of the earliest approaches to nonlinear filtering is the extended Kalman filter (EKF) [1] where the nonlinear functions are replaced with their first order Taylor series approximation. More recent nonlinear filtering approaches are known as particle filters (PF) [8] where the probability distribution of the state is approximated using weighted samples in the state space, such that the state estimate can be obtained as the weighted mean of the samples. The samples are obtained using a process which is common to all PF algorithms and is known as importance sampling (IS). Since sampling from the state's posterior distribution is usually very hard, using IS one can sample from some proposal distribution with the same support and evaluate the weight of the sample using the posterior. The PF approach has been shown to provide significant improvement compared to the EKF in areas such as target tracking and communications [16].

In order to deal with the model uncertainties in the estimation model one popular approach is to use a minimax estimator, which is the estimator that minimizes the worst case of some cost function in the region of uncertainty [40] of the parameter. Recently there has been much work published regarding such minimax robust estimation using the linear Gaussian model with model uncertainties e.g. [26]- [29].

The contribution of this work in the area of nonlinear filtering is to consider the use of the IFGT to perform nonlinear filtering. We also present a new proposal distribution which is shown experimentally to improve the MSE performance significantly, and derive new performance bounds for the IFGT which are significantly tighter than the previous bound. We demonstrate how these bounds facilitate the choice of the IFGT parameters for the filtering algorithm in practice.

The contribution of this work in the area of robust estimation is to consider the minimax estimation of a random parameter in a linear Gaussian model. In a recent work in this area [28, 29] new minimax estimators were developed assuming that the *Gram* matrix [40] of $\boldsymbol{H}$ weighted by the inverse covariance matrix of the noise vector, and the random parameter's covariance matrix, are diagonalized by the same unitary matrix. This assumption significantly limits the applicability of these estimators. In this dissertation we derive a new minimax which we call the generalized difference regret (GDR) estimator which lifts the joint diagonalizability by the same unitary matrix assumption. Furthermore the uncertainty region is expressed as the intersection set of the eigenvalue uncertainty region and an elementwise uncertainty region, unlike the previous estimators in which the region of uncertainty was expressed using lower and upper bounds on each of the eigenvalues alone. This leads to a smaller uncertainty region which may be less pessimistic than the uncertainty set, and therefore lead to improved results in cases where the SNR is low. The new estimator can be obtained efficiently using semidefinite programming. Using the robust optimization techniques considered for the estimation problem in the linear Gaussian model, we also propose a new approach for robust estimation of certain parameters that are used as features for an unexploded ordnance (UXO) classification algorithm, when there exists uncertainty regarding their exact position. Specifically, we build on the minimax the approach presented in [57] which was solved using an approximate dynamic programming technique, and present a nonlinear programming solution that is based on replacing the maximization problem with the upper bound that is given by the dual form of the problem.

The dissertation is organized as follows: In Chapter 2 we provide the background on fast Kernel density estimation (KDE) and nonlinear filtering algorithms, and in Chapter 3 we provide the background on convex optimization, robust minimax estimation and sensor networks. In Chapter 4 we derive new performance bounds for the IFGT and perform nonlinear filtering using a new proposal distribution and using the IFGT. In Chapter 5 we develop the new GDR robust minimax estimator and present its experimental results. In Chapter 6 we develop a nonlinear programming approach for robust parameter estimation with sensor positional uncertainty for the UXO classification problem. In Chapter 7 we make concluding remarks and present directions for future research in this field.

Chapter 2

Background- Fast Kernel Density Estimation and Nonlinear Filtering

In this chapter we provide the theoretical background about linear and nonlinear filtering as well as background about fast KDE methods. We later show how KDE methods and in particular the IFGT can be used to perform nonlinear filtering.

2.1 Fast KDE

KDE is most often used to determine a non-parametric PDF from a set of samples $\{\boldsymbol{s}_j\}_{j=1}^{N}$ [3]. The KDE evaluated at a target point t then takes the form

$$G(\boldsymbol{t}) \triangleq \sum_{j=1}^{N} q_j K\left(\frac{\boldsymbol{t}-\boldsymbol{s}_j}{h}\right) \tag{2.1}$$

where $K(\cdot)$ is a kernel function with scale parameter h (the "bandwidth"), $\boldsymbol{s}_j \in \mathbb{R}^D$ are referred to as source points, $q_j \in \mathbb{R}$ are the source strengths, and D denotes the dimension. For the purpose of obtaining a good non-parametric estimate of the PDF, the optimal bandwidth h^* has to be estimated [9]. Evaluating $G(\cdot)$ in (2.1) at N different target points has quadratic computational complexity, therefore there have been several approaches suggested in literature to reduce this computational complexity while compromising on the accuracy.

The different methods to fast evaluation of KDE rely on the divide and conquer approach where either the source space or the joint source and target space are first partitioned into different regions. Beyond the initial partitioning the various algorithms can be categorized into discrete approximation techniques such as the single and dual tree algorithms, [4, 9] and continuous approximation algorithms such as the FGT and IFGT. The single and dual tree algorithms are based on a space partitioning of the source and target points. This space partitioning is then used in order to avoid computations whose overall contribution is below some threshold. Furthermore, these algorithms can be used under several choices of kernel functions and the speed of these methods is independent of the dimension. The FGT applies only in the case of a Gaussian kernel where a Hermite polynomial expansion is used to approximate the Gaussian kernel function beyond the initial space partitioning. The Hermite expansion enables (2.1) to be approximated to any degree of accuracy using an alternative expression whose computational complexity is linear in the number of samples, however the computational constant grows exponentially with the dimension. The IFGT solves this problem by using a different form of expansion for the Gaussian kernel such that the computational constant grows more moderately with the dimension. Both the FGT and IFGT can offer a significant speedup when the bandwidth is larger than the optimal bandwidth, whereas the single and dual tree algorithms are most effective when the bandwidth is smaller than the optimal bandwidth. This observation was used in [64] to construct a hybrid dual tree FGT algorithm which was used in order to speed up the process of estimating the optimal bandwidth. In such a scenario one has to compute a score function for many choices of the bandwidth parameter which are different from the optimal one.

2.1.1 Discrete approximation techniques

Discrete approximation techniques are based on building a space-partitioning tree for a set of points. The space partitioning tree, which is also known as a *kd*-tree [23], is a binary tree data structure where in each bifurcation the points are split along one of the dimensions up until the leaf level that holds a single point. Each node then represents a box that bounds all the points that are in lower levels in the tree. Figure 2.1 shows the space partitioning that is obtained using the *kd*-tree that is shown in Figure 2.2, where the black circles represent the points within each bounding box. For the sake of presentation clarity, the *kd*-tree in Figure 2.2 does not show partitioning until each point is bounded by a unique box, however in a real *kd*-tree used in a fast KDE algorithm the leaf level includes a single point. Two algorithms that make use of the *kd*-tree data structure to perform fast KDE evaluation are the single tree and dual tree algorithms [9].

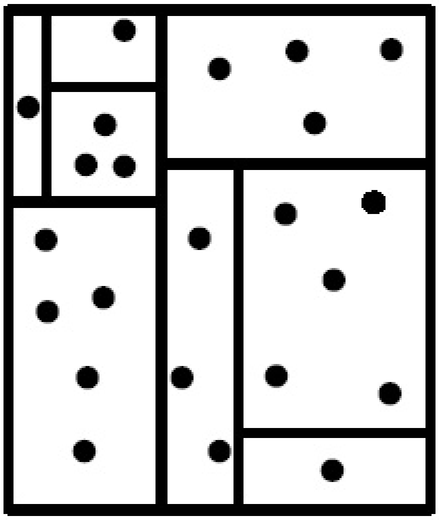

Figure 2.1: An example of the space partitioning obtained using a kd-tree

Single tree algorithm

The single tree algorithm uses a space partitioning tree for the source points. Subsequently each target point is traversed along the tree and the bounding boxes are used to exclude the source points whose distance is larger than a pre-specified constant from the target point.

Dual tree algorithm

The dual tree algorithm uses a space partitioning tree both for the source points and for the target points. Subsequently both trees are traversed simultaneously such that larger chunks of source and target points whose distance from each other is larger from a pre-specified constant can be excluded from computation.

2.1.2 The fast Gauss transform

The FGT considers the special case where the kernel function in (2.1) is a Gaussian:

$$G(\boldsymbol{t}) = \kappa \sum_{j=1}^{N} q_j e^{-\|\boldsymbol{t}-\boldsymbol{s}_j\|^2/2\sigma^2} \tag{2.2}$$

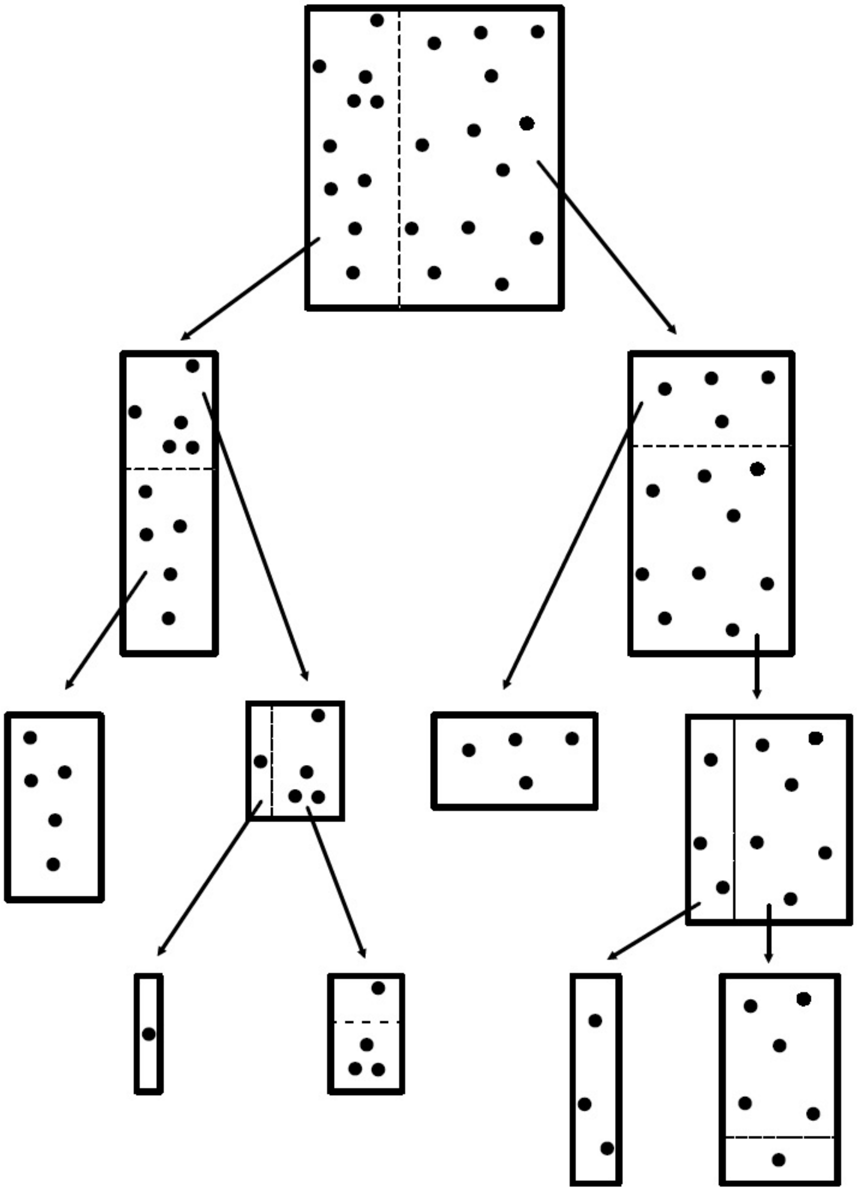

Figure 2.2: An example of the kd tree for the space partitioning in Figure 2.1

where σ^2 denotes the variance of the Gaussian kernel, and $\kappa^{-1} = (2\pi\sigma^2)^{\frac{D}{2}}$. The basis for the FGT is the one dimensional Hermite expansion of the Gaussian function around point $s^* \in \mathbb{R}$:

$$e^{-(t-s)^2/2\sigma^2} = \sum_{n=1}^{\infty} \frac{1}{n!} \left(\frac{s - s^*}{\sqrt{2}\sigma} \right)^n h_n \left(\frac{t - s^*}{\sqrt{2}\sigma} \right) \tag{2.3}$$

where $t, s \in \mathbb{R}$ and where $h_\alpha(t) \triangleq (-1)^\alpha \frac{d^\alpha}{dt^\alpha} e^{-t^2}$, and satisfies the recursion: $h_{\alpha+1}(t) = 2th_\alpha(t) - 2\alpha h_{\alpha-1}(t)$.

The extension to the multidimensional case is then obtained using the products of the one dimensional expansions. We adopt the multi-index notation used in [5], where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_D)$ is a D dimensional tuple of nonnegative integers. For any $t \in \mathbb{R}^D$ we define $\boldsymbol{t}^{\boldsymbol{\alpha}} = t_1^{\alpha_1} \cdots \alpha_D^{\alpha_D}$, and $h_{\boldsymbol{\alpha}}(t)$=$h_{\alpha_1}(t_1) \cdots h_{\alpha_D}(t_D)$. The factorial of the multi-index $\boldsymbol{\alpha}$ is defined as $\boldsymbol{\alpha}! = \alpha_1! \cdots \alpha_D!$, and the length is defined as $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_D$. The multidimensional expansion of a Gaussian around $s^* \in \mathbb{R}^D$ therefore takes the form

$$e^{-\frac{\|\boldsymbol{t}-\boldsymbol{s}\|^2}{2\sigma^2}} = \sum_{\boldsymbol{\alpha} \geq 0} \frac{1}{\boldsymbol{\alpha}!} h_{\boldsymbol{\alpha}} \left(\frac{\boldsymbol{\Delta t}}{\sqrt{2}\sigma} \right) \left(\frac{\boldsymbol{\Delta s}}{\sqrt{2}\sigma} \right)^{\boldsymbol{\alpha}} \tag{2.4}$$

where $t, s \in \mathbb{R}^D$, $\boldsymbol{\Delta t} = \boldsymbol{t} - \boldsymbol{s}^*$, and $\boldsymbol{\Delta s} = \boldsymbol{s} - \boldsymbol{s}^*$.

Truncating (2.4) after the first p terms and substituting into (2.2) yields

$$\begin{aligned} G(\boldsymbol{t}) &= \kappa \cdot \sum_{\boldsymbol{\alpha} < p} A_{\boldsymbol{\alpha}} h_{\boldsymbol{\alpha}} \left(\frac{\boldsymbol{\Delta t}}{\sqrt{2}\sigma} \right) + E_T, & (2.5) \\ A_{\boldsymbol{\alpha}} &= \frac{1}{\boldsymbol{\alpha}!} \sum_{j=1}^{N} q_j \left(\frac{\boldsymbol{\Delta s}_j}{\sqrt{2}\sigma} \right)^{\boldsymbol{\alpha}}, & (2.6) \end{aligned}$$

where $\boldsymbol{\Delta s}_j = \boldsymbol{s}_j - \boldsymbol{s}^*$, E_T is the truncation error, the notation $\boldsymbol{\alpha} < p$ denotes the set of all multi-indices $\boldsymbol{\alpha}$ with elements between 0 and $p - 1$, and $A_{\boldsymbol{\alpha}}$ are the FGT coefficients.

Let N_B sources with weights $\{q_j\}_{j=1}^{N_B}$ lie in a D dimensional box with center $\boldsymbol{s}_B$ and side lengths $2r\sigma$, with $r < 1$. Then the error E_T due to truncating the series (2.5) after p^D terms satisfies the bound [6]:

$$|E_T| \leq Q_B \epsilon_p^{FGT} \tag{2.7}$$

where,

$$\epsilon_p^{FGT} = \frac{\kappa}{(1-r)^D} \sum_{d=0}^{D-1} \binom{D}{d} (1 - r^p)^d \left(\frac{r^p}{\sqrt{p!}} \right)^{D-d} \tag{2.8}$$

and where $Q_B = \sum_{j=1}^{N_B} |q_j|$

The FGT partitions the source space into D dimensional boxes of side $2r\sigma$ with $r < 1$, which are parallel to the axes. Each source point s_j is then assigned to the box in which it lies, and the FGT coefficients (2.6) for each box are computed using the sources assigned to it where the s^* for each box is the center of the box. The FGT evaluates each of the target points at each of the $(2n + 1)^D$ $n \in \mathbb{N}$ nearest boxes, which adds an error which is bounded by $Qe^{-2r^2n^2}$ due to ignoring all the source points in the other boxes. The truncation error incurred by evaluating the target points at the $(2n+1)^D$ nearest boxes using (2.5) is bounded by $Q\epsilon_p^{FGT}$, where $Q = \sum_{j=1}^{N} |q_j|$. Therefore the FGT can approximate the KDE to any degree of accuracy.

As can be seen from (2.5) and (2.6), the computational complexity of the FGT is linear with the number of samples, however since the number of FGT coefficients grows exponentially with the dimension so does the computational constant. Since (2.5) has to be evaluated at each cluster, the computational complexity of the FGT also grows linearly with the number of clusters. The FGT can also make use of translation operators which can reduce the computational cost, however the computational constant still grows exponentially with the dimension.

2.1.3 The improved fast Gauss transform

The IFGT [7] also deals with the Gaussian kernel case, however it uses another form of expansion around a point $\boldsymbol{s}^* \in \mathbb{R}^D$ instead of the Hermite expansion (2.4)

$$\begin{aligned} e^{-\|\boldsymbol{t}-\boldsymbol{s}\|^2/2\sigma^2} &= e^{(-\|\boldsymbol{\Delta t}\|^2-\|\boldsymbol{\Delta s}\|^2+2\boldsymbol{\Delta t}^T\boldsymbol{\Delta s})/2\sigma^2} \\ &= e^{-\frac{\|\boldsymbol{\Delta t}\|^2+\|\boldsymbol{\Delta s}\|^2}{2\sigma^2}} \sum_{n=0}^{\infty} \frac{2^n}{n!}\left(\frac{\boldsymbol{\Delta t \Delta s}}{2\sigma^2}\right)^n \qquad (2.9) \end{aligned}$$

where $\boldsymbol{t}, \boldsymbol{s} \in \mathbb{R}^D$, $\boldsymbol{\Delta t} = \boldsymbol{t} - \boldsymbol{s}^*$, $\boldsymbol{\Delta s} = \boldsymbol{s} - \boldsymbol{s}^*$, and where (2.9) follows from expanding the exponent of the cross term using a Taylor expansion. Truncating (2.9) after the first p terms and substituting into (2.2), we have:

$$\begin{aligned} G(\boldsymbol{t}) &= \kappa e^{-\|\boldsymbol{\Delta t}\|^2/2\sigma^2} \cdot \sum_{|\boldsymbol{\alpha}|<p} \tilde{A}_{\boldsymbol{\alpha}} \left(\frac{\boldsymbol{\Delta t}}{\sigma}\right)^{\boldsymbol{\alpha}} + E_T, \qquad (2.10) \\ \tilde{A}_{\boldsymbol{\alpha}} &= \frac{1}{\boldsymbol{\alpha}!} \sum_{j=1}^{N} \tilde{q}_j \left(\frac{\Delta \boldsymbol{s}_j}{\sigma}\right)^{\boldsymbol{\alpha}}, \qquad (2.11) \end{aligned}$$

where $|\boldsymbol{\alpha}| < p$ denotes the set of multi-indices $\boldsymbol{\alpha}$ whose sum of elements is less than p, and where $\tilde{q}_j = q_j e^{-\|\boldsymbol{\Delta s}_j\|^2/2\sigma^2}$. Since the summation in (2.10) is over all the IFGT coefficients with multi-index $\boldsymbol{\alpha}$ such that $|\boldsymbol{\alpha}| < p$, then the computational constant grows more moderately with the dimension compared to the computational constant of the FGT that grows exponentially with the dimension.

Let r_t and r_s denote upper bounds on $\|\boldsymbol{\Delta t}\|$ and $\|\boldsymbol{\Delta s}\|$ respectively, and let N_B sources with weights $\{q_j\}_{j=1}^{N_B}$ lie in a ball with center $\boldsymbol{s}_B$ and radius r_s. Then the error E_T due to truncating the series (2.10) after the pth order, when evaluating a target point within distance r_t from the ball's center, satisfies the bound [7]:

$$|E_T| \leq Q_B \epsilon_p^{IFGT} \tag{2.12}$$

where

$$\epsilon_p^{IFGT} = \kappa \frac{2^p}{p!} \left(\frac{r_t}{\sigma}\right)^p \left(\frac{r_s}{\sigma}\right)^p \tag{2.13}$$

and where $Q_B = \sum_{j=1}^{N_B} |q_j|$.

The IFGT clusters the source points into balls with radius r_s and computes the IFGT coefficients $\tilde{A}_{\boldsymbol{\alpha}}$ for each ball using (2.11). For each target point the IFGT finds the clusters whose centers lie within the range r_t, and evaluates the target point using (2.10) at each of these clusters. The error due to ignoring all the source points outside of range r_t from the clusters' centers is bounded by $Qe^{-(r_t-r_s)^2/2\sigma}$ [14]. The truncation error incurred by evaluating the target point at each of the clusters is bounded by $Q\epsilon_p^{IFGT}$, where $Q = \sum_{j=1}^{N} |q_j|$. Therefore for every choice of r_s r_t and p, we can bound the approximation error of the KDE. However since the truncation error and the error due to ignoring the source points outside of range r_t are coupled, it is unclear from this formulation how to choose r_s and r_t such that the approximation is smaller than a specified value $Q\epsilon$. Furthermore as was noted in [14,17] the bound (2.12) is very pessimistic. An approach to determining the truncation order and the parameters r_t,r_s which is based on a bound which is tighter than (2.12) was presented in [17]. First in order to decouple the truncation error and the error due to ignoring source points outside of range r_t, it is proposed to use r_t of the form

$$r_t = r_s + R \tag{2.14}$$

where $R \in \mathbb{R}$, in which case the error due to ignoring source points outside of range r_t is $Qe^{-R^2/2\sigma}$ and we can choose R such that this error is smaller than $Q\epsilon$. Second, the source points are clustered into K balls where K is chosen according to some heuristic criteria which is related to the computational constant of the IFGT rather than its accuracy. The parameter r_s for each ball can then be obtained as the radius of the ball. The truncation order p for each cluster is then determined such that $\epsilon_p^{IFGT_2}(\|\boldsymbol{\Delta s}_j\|/\sigma) < \epsilon \ \forall j = 1, \ldots, N_B$, where the truncation error for the Gaussian kernel is given by

$$\epsilon_p^{IFGT_2}(r_0) = \kappa \frac{2^p}{p!}(r_0 \|\boldsymbol{\tau}\|_*) e^{-(r_0 - \|\boldsymbol{\tau}\|_*)^2} \tag{2.15}$$

where

$$\|\boldsymbol{\tau}\|_* = \min(\frac{r_0 + \sqrt{r_0^2 + 2p}}{2}, \frac{r_t}{\sigma}) \tag{2.16}$$

The truncation error E_T in (2.10) is therefore bounded by

$$|E_T| \leq \sum_{j=1}^{N} |q_j| \epsilon_p^{IFGT_2}(\|\boldsymbol{\Delta s}_j\|/\sigma) \leq Q\epsilon \tag{2.17}$$

Since the number of clusters is chosen based on a heuristic criteria rather than the bound and subsequently the truncation order is chosen such that a specified upper bound is satisfied, there is no guarantee that this choice is optimal under any criteria.

2.2 Filtering

Filtering is concerned with dynamic systems of the form:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= f_k(\boldsymbol{x}_k, \boldsymbol{w}_k) && (2.18) \\ \boldsymbol{z}_k &= h_k(\boldsymbol{x}_k) + \boldsymbol{v}_k && (2.19) \end{aligned}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$ denote the state and measurement vectors at time k respectively, $\boldsymbol{w}_k$, and $\boldsymbol{v}_k$ denote two independent noise sequences, and $f()$, $h()$ can be any two functions of the state vector. The aim of filtering is to find the posterior state distribution $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$, where $\boldsymbol{z}_{1:k} \triangleq \{\boldsymbol{z}_1, ..., \boldsymbol{z}_k\}$. The posterior state distribution can be computed recursively using [11]:

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) &= \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}) \boldsymbol{dx}_{k-1} && (2.20) \\ p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) &= \frac{1}{c_k} p(\boldsymbol{z}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) && (2.21) \\ c_k &\triangleq \int p(\boldsymbol{z}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) \boldsymbol{dx}_k && (2.22) \end{aligned}$$

When the state and measurement equations are linear and the noises are Gaussian, the one step ahead prediction PDF $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})$ and the posterior PDF $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ are Gaussian and (2.20)-(2.22) can be evaluated analytically using the Kalman filter [1], otherwise the recursive equations can only be approximated and the problem is known as nonlinear filtering. The different approaches to approximating (2.20)-(2.22) can be categorized into deterministic and sequential Monte Carlo (SMC) methods. The deterministic methods such as the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) are in close relation to the Kalman filter and are based on the assumption that the posterior and one step ahead prediction PDFs are Gaussian. The SMC approach, which is also known as particle filtering [8, 21], uses discrete samples to approximate the posterior. Most particle filters approximate the joint density $p(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k})$ although only the marginal $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ is of interest, thereby necessitating the use of a resampling stage. The other paradigm to particle filtering [2, 11] approximates the marginal $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ directly and was termed the marginal particle filter (MPF) in [2]. The MPF uses Monte Carlo (MC) integration and importance sampling (IS) to approximate the integrals in the Bayes recursion equations (2.20)-(2.22). Using IS requires the definition of a proposal distribution from which one draws the samples that are used to obtain the MC approximation of the integral. Although in theory any probability density function (PDF) with the same support as the posterior can be used, in practice due to the limited computational resources it is advisable that the proposal be as similar as possible to the posterior in order to obtain the best performance.

2.2.1 The Kalman filter

The discrete time Kalman filter applies to the case where the state and measurement equations are linear thus (2.18), (2.21) take the form:

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k\boldsymbol{x}_k + \boldsymbol{G}_k\boldsymbol{w}_k \tag{2.23}$$

$$\boldsymbol{z}_k = \boldsymbol{H}_k^T\boldsymbol{x}_k + \boldsymbol{v}_k \tag{2.24}$$

Additionally assume that the zero mean Gaussian noise sequences $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ satisfy

$$\mathbb{E}[\boldsymbol{v}_k\boldsymbol{v}_\ell^T] = \boldsymbol{R}_k\delta_{k\ell} \qquad \mathbb{E}[\boldsymbol{w}_k\boldsymbol{w}_\ell^T] = \boldsymbol{Q}_k\delta_{k\ell}, \tag{2.25}$$

and the initial sate $\boldsymbol{x}_0$ is a Gaussian random variable with mean $\bar{\boldsymbol{x}}_0$ and covariance $\boldsymbol{P}_0$ independent of $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$. Under these conditions the one step ahead prediction and the posterior PDFs are Gaussian and are fully defined by their mean and covariance matrix.

The Kalman filter can be used to compute the estimates

$$\hat{\boldsymbol{x}}_{k|k-1} = \mathbb{E}[\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}] \qquad \hat{\boldsymbol{x}}_{k|k} = \mathbb{E}[\boldsymbol{x}_k|\boldsymbol{z}_{1:k}], \tag{2.26}$$

and similarly the associated error covariance matrices

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbb{E}\{[\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}][\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}]^T|\boldsymbol{z}_{1:k-1}\} \tag{2.27}$$

and

$$\boldsymbol{\Sigma}_{k|k} = \mathbb{E}\{[\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}][\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}]^T|\boldsymbol{z}_{1:k}\}. \tag{2.28}$$

The Kalman filter computes the state and covariance estimates recursively as follows:

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{\Sigma}_{k|k-1}\boldsymbol{H}_k(\boldsymbol{H}_k^T\boldsymbol{\Sigma}_{k|k-1}\boldsymbol{H}_k + \boldsymbol{R}_k)^{-1}(\boldsymbol{z}_k - \boldsymbol{H}_k^T\hat{\boldsymbol{x}}_{k|k-1}) && (2.29)\\
\boldsymbol{\Sigma}_{k|k} &= \boldsymbol{\Sigma}_{k|k-1} - \boldsymbol{\Sigma}_{k|k-1}\boldsymbol{H}_k(\boldsymbol{H}_k^T\boldsymbol{\Sigma}_{k|k-1}\boldsymbol{H}_k + \boldsymbol{R}_k)^{-1}\boldsymbol{H}_k^T\boldsymbol{\Sigma}_{k|k-1} && (2.30)\\
\hat{\boldsymbol{x}}_{k+1|k} &= \boldsymbol{F}_k\hat{\boldsymbol{x}}_{k|k} && (2.31)\\
\boldsymbol{\Sigma}_{k+1|k} &= \boldsymbol{F}_k\boldsymbol{\Sigma}_{k|k}\boldsymbol{F}_k^T + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T && (2.32)
\end{aligned}$$

where the filter is initialized using $\hat{\boldsymbol{x}}_{0|-1} = \bar{\boldsymbol{x}}_0$ and $\boldsymbol{\Sigma}_{0|-1} = \boldsymbol{P}_0$.

2.2.2 Deterministic nonlinear filtering

The extended Kalman filter

The EKF approximates the nonlinear functions in the state space model with a first order Taylor series around the state and noise expected means. The linearized state space model therefore takes the form

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \boldsymbol{F}_k\boldsymbol{x}_k + \boldsymbol{G}_k\boldsymbol{w}_k + \boldsymbol{u}_k && (2.33)\\
\boldsymbol{z}_k &= \boldsymbol{H}_k^T\boldsymbol{x}_k + \boldsymbol{v}_k + \boldsymbol{y}_k && (2.34)
\end{aligned}$$

where

$$\boldsymbol{F}_k = \left.\frac{\partial f_k(\boldsymbol{x},\boldsymbol{w})}{\partial x}\right|_{\substack{x=\hat{x}_{k|k}\\ \boldsymbol{w}=0}} \quad , \quad \boldsymbol{H}_k^T = \left.\frac{\partial h_k(\boldsymbol{x})}{\partial x}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k|k}} \quad , \quad \boldsymbol{G}_k = \left.\frac{\partial f_k(\boldsymbol{x},\boldsymbol{w})}{\partial w}\right|_{\substack{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k|k-1}\\ \boldsymbol{w}=0}} \tag{2.35}$$

and

$$\boldsymbol{u}_k = f_k(\hat{\boldsymbol{x}}_{k|k}) - \boldsymbol{F}_k\hat{\boldsymbol{x}}_{k|k} \quad , \quad \boldsymbol{y}_k = h_k(\hat{\boldsymbol{x}}_{k|k-1}) - \boldsymbol{H}_k^T\hat{\boldsymbol{x}}_{k|k-1} \tag{2.36}$$

The Kalman filter is then used with the linearized dynamic system and the equations are a simple variation of (2.29)-(2.32):

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{\Sigma}_{k|k-1}\boldsymbol{H}_k(\boldsymbol{H}_k^T\boldsymbol{\Sigma}_{k|k-1}\boldsymbol{H}_k + \boldsymbol{R}_k)^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k)) && (2.37)\\
\boldsymbol{\Sigma}_{k|k} &= \boldsymbol{\Sigma}_{k|k-1} - \boldsymbol{\Sigma}_{k|k-1}\boldsymbol{H}_k(\boldsymbol{H}_k^T\boldsymbol{\Sigma}_{k|k-1}\boldsymbol{H}_k + \boldsymbol{R}_k)^{-1}\boldsymbol{H}_k^T\boldsymbol{\Sigma}_{k|k-1} && (2.38)\\
\hat{\boldsymbol{x}}_{k+1|k} &= f_k(\hat{\boldsymbol{x}}_{k|k}) && (2.39)\\
\boldsymbol{\Sigma}_{k+1|k} &= \boldsymbol{F}_k\boldsymbol{\Sigma}_{k|k}\boldsymbol{F}_k^T + \boldsymbol{G}_k\boldsymbol{Q}_k\boldsymbol{G}_k^T && (2.40)
\end{aligned}$$

2.2.3 Particle filtering

The particle filters (PF) paradigm to nonlinear filtering replaces the parametric Gaussian approximation of the state distribution with a non-parametric discrete particle approximation where each particle consists a sample and a weight. Given sufficient particles one can approximate any distribution of the state vector. In order to obtain this approximation PF rely on importance sampling, and therefore require the design of proposal distributions that approximate the posterior distribution. Although in theory any probability density function (PDF) with the same support as the posterior can be used, in practice due to the limited computational resources it is advisable that the proposal be as similar as possible to the posterior in order to obtain the best performance. Most PF approximate the joint density $p(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k})$ although only the marginal $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ is of interest, thereby necessitating the use of a resampling stage in which samples with high weights are multiplied and samples with low weights are discarded. Another class of PF which is called the marginal particle filter (MPF) uses Monte Carlo (MC) integration and importance sampling (IS) to approximate the integrals in the Bayes recursion equations (2.20)-(2.22), therefore the marginal $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ is approximated directly. The use of kernel density estimation (KDE) in particle filters has been considered in [2, 19, 20], however whereas in [19, 20] KDE is used as a part of the resampling stage, in [2] the KDE was regarded as another interpretation of the Chapman-Kolmogorov equation (CKE) (2.20).

MC integration and IS

Consider the generic problem of evaluating integrals of the form:

$$I = \mathbb{E}_f[h(\boldsymbol{x})] = \int h(\boldsymbol{x})f(\boldsymbol{x})dx. \tag{2.41}$$

and suppose that $f()$ is a valid PDF. The MC framework approximates (2.41) by drawing the samples $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ from $f(\boldsymbol{x})$, and approximating (2.41) using the empirical average

$$I_N = \frac{1}{N}\sum_{i=1}^{N} h(\boldsymbol{x}_i), \tag{2.42}$$

where I_N converges almost surely to I [10].

When it is difficult to sample from the distribution $f(\boldsymbol{x})$, then IS can be used to approximate the integral. Assume that we can obtain samples from a proposal distribution $\pi(\boldsymbol{x})$ that satisfies the condition support$(\pi) \supseteq$ support(f), then by rewriting (2.41) as

$$I = \mathbb{E}_\pi[h(\boldsymbol{x})\frac{f(\boldsymbol{x})}{\pi(\boldsymbol{x})}] = \int h(\boldsymbol{x})\frac{f(\boldsymbol{x})}{\pi(\boldsymbol{x})}\pi(\boldsymbol{x})\boldsymbol{dx}, \tag{2.43}$$

we can use the the MC integration framework to approximate (2.43). By drawing the samples $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ from $\pi(\boldsymbol{x})$ then the approximation takes the form

$$I_N = \frac{1}{N}\sum_{i=1}^{N} h(\boldsymbol{x}_i)\frac{f(\boldsymbol{x}_i)}{\pi(\boldsymbol{x}_i)}, \tag{2.44}$$

and since (2.44) is a Monte Carlo estimator it converges almost surely to I. Although the IS framework applies to a broad range of proposal distributions, the convergence rate depends on how close the proposal distribution $\pi(\boldsymbol{x})$ is to $f(\boldsymbol{x})$. For poor choices of $\pi(\boldsymbol{x})$ the convergence rate may be quite slow.

Another important factor influencing the quality of the MC approximation is the quality of the samples obtained from the proposal distribution. Using entirely random sampling does not explore the sample space in the most uniform way, whereas quasi-MC (QMC) methods [11] (also known as low discrepancy point sets) which are deterministic in nature, lead to a more uniform exploration of the sample space. QMC methods have been shown empirically as well as theoretically to lead to a faster convergence compared to entirely random sampling. There have been several approaches presented in literature to generating QMC points [12]. In this work we use the Halton sequence which pertains to the digital nets low discrepancy point sets family. The use of QMC methods in this work is limited to sampling from a Gaussian mixture of the form

$$\sum_{i=1}^{N} w_k^{(i)} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}), \tag{2.45}$$

where $\{w^{(i)}\}_{i=1}^{N}$ are weights and $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$ is a Gaussian with mean $\boldsymbol{\mu}_k^{(i)}$ and covariance matrix $\boldsymbol{\Sigma}_k^{(i)}$. The exact details regarding QMC sampling of this form are given in algorithm 1. A comprehensive study on the use of QMC methods for nonlinear filtering was presented in [11].

Algorithm 1: [QMC sampling from a Gaussian mixture]

```
• Draw $\{j^{(\ell)}\}_{\ell=1}^{N}$ from the probability mass function $\{w_{k-1}^{(i)}\}_{i=1}^{N}$.
• for  $i = 1, \dots, N$
    – Find the set $E_i \triangleq \{\ell | j^{(\ell)} = i\}$, and let $|E_i|$ denote the number of elements in $E_i$.
    – Generate $|E_i|$ QMC points $\{\boldsymbol{u}^{(\ell)}\}_{\ell=1}^{|E_i|}$ in $[0,1)^D$.
    – Transform $\{\boldsymbol{u}^{(\ell)}\}_{\ell=1}^{|E_i|}$ to $\{\boldsymbol{x}^{(j)}\}_{j=|E_{i-1}|+1}^{|E_i|}$:
        ∗ perform Cholesky decomposition on $\boldsymbol{\Sigma}_k^{(i)}$, i.e. $\boldsymbol{\Sigma}_k^{(i)} = \boldsymbol{R}^T\boldsymbol{R}$
        ∗ transform $\{\boldsymbol{u}^{(\ell)}\}_{\ell=1}^{|E_i|}$ to $\{\boldsymbol{y}^{(\ell)}\}_{\ell=1}^{|E_i|}$, via $y_d^{(\ell)} = \phi^{-1}(u_d^{(\ell)}), d = 1, \dots, D$, where $\phi(x) = (1/\sqrt{2\pi}) \int_{-\infty}^{x} \exp(-w^2/2) dw$.
        ∗ set $\boldsymbol{x}^{(|E_{i-1}|+\ell)} = \boldsymbol{\mu}_k^{(i)} + \boldsymbol{R}\boldsymbol{y}^{(\ell)}$, $\ell = 1, \dots, |E_i|$.
  end for
```

Sequential importance sampling

The sequential importance sampling (SIS) PF approximates the joint density $p(\boldsymbol{x}_{1:k-1}|\boldsymbol{z}_{1:k-1})$ using N samples $\{w_{1:k-1}^{(i)}, \boldsymbol{x}_{1:k-1}^{(i)}\}$. The samples for time k are obtained by drawing from a proposal distribution $\boldsymbol{x}_k^{(i)} \sim q(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{y}_{1:k})$ and setting $\boldsymbol{x}_{1:k}^{(i)} = (\boldsymbol{x}_{1:k-1}^{(i)}, \boldsymbol{x}_k^{(i)})$ $\forall i = 1 \dots N$. The weights are then updated using

$$w_{1:k}^{(i)} \propto w_{1:k-1}^{(i)} \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)}) p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})}{q(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{z}_{1:k})}. \tag{2.46}$$

and subsequently normalized such that $\sum_{i=1}^{N} w_{1:k}^{(i)} = 1$. The choice of proposal distribution that has been advocated by many researchers [24] is $q(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{z}_{1:k}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{z}_{1:k})$, however since often it may be hard to sample from this distribution a popular choice is to use $q(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{z}_{1:k}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$. The main drawback with the latter choice is that it does not incorporate the latest observation $\boldsymbol{z}_k$, therefore it may require the use of more particles in the filtering algorithm. Another limitation associated with the SIS PF is the degeneracy of the particles over time [24], where after a few iterations one of the normalized weights tends to 1 while the remaining weights tend to zero. In order to avoid the degeneracy problem it is necessary to resample the distribution once every few iterations, such that particles with large weights are multiplied and particles with small weights are discarded.

The marginal particle filter

The nonlinear filtering approach that we use in this work follows that in [11] where IS and QMC methods are used to approximate the integrals in (2.20)-(2.22). Assuming we have a proposal denisty $\pi(\boldsymbol{x}_k)$ from which we can sample easily, then using (2.44) to approximate the integrals we can replace the recursion (2.20)-(2.22) with:

$$\hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) = \sum_{i=1}^{N} w_{k-1}^{(i)} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}) \tag{2.47}$$

$$w_k^{(i)} \triangleq \frac{\hat{p}(\boldsymbol{x}_k^{(i)}|\boldsymbol{z}_{1:k})}{\pi(\boldsymbol{x}_k^{(i)})} = \frac{1}{\hat{c}_k} p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)}) \frac{\hat{p}(\boldsymbol{x}_k^{(i)}|\boldsymbol{z}_{1:k-1})}{\pi(\boldsymbol{x}_k^{(i)})} \tag{2.48}$$

$$\text{where } \hat{c}_k = \sum_{i=1}^{N} p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)}) \frac{\hat{p}(\boldsymbol{x}_k^{(i)}|\boldsymbol{z}_{1:k-1})}{\pi(\boldsymbol{x}_k^{(i)})} \tag{2.49}$$

and the state estimator can also be approximated using the IS framework:

$$\hat{\boldsymbol{x}}_k \triangleq \frac{1}{\tilde{c}_k} \int \boldsymbol{x}_k \cdot p(\boldsymbol{z}_k|\boldsymbol{x}_k) \hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) \boldsymbol{dx}_k \approx \sum_{i=1}^{N} \boldsymbol{x}_k^{(i)} w_k^{(i)} \tag{2.50}$$

where $\tilde{c}_k = \int p(\boldsymbol{z}_k|\boldsymbol{x}_k)\hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})\boldsymbol{dx}_k$, and the result is obtained when using IS to approximate both the integrand and $\tilde{c}_k$.

The MPF filtering algorithm is summarized as follows.

Algorithm 2: [The MPF]

```
• Initialization: Sample $N$ points $\left\{\boldsymbol{x}_0^{(i)}\right\}_{i=1}^{N}$ from the prior distribution $p(\boldsymbol{x}_0)$, and set $w_0^{(i)} = \frac{1}{N}$, $i = 1, \ldots, N$.
• for  $k = 1, 2, \ldots$
    – Sample $N$ points $\left\{\boldsymbol{x}_k^{(i)}\right\}_{i=1}^{N}$ from the proposal distribution $\pi(\boldsymbol{x}_k)$.
    – Compute the predictive density $\hat{p}(\boldsymbol{x}_k^{(i)}|\boldsymbol{z}_{1:k-1})$, $i = 1, \ldots, N$, using (2.47).
    – Compute the posterior weights $w_k^{(i)}$, $i = 1, \ldots, N$, using (2.48) and (2.49).
    – Compute the state estimator using (2.50).
  end for
```

The choice of the proposal distribution has a significant affect on the performance of the MPF. Next we describe several choices for proposal distributions.

1. <u>SIS</u>- If we take

$$\pi(\boldsymbol{x}_k) = p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) \tag{2.51}$$

it simplifies (2.48)

$$w_k^{(i)} \propto p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)}) \tag{2.52}$$

thereby avoiding the computationally expensive stage of evaluating the predictive density values (2.47). It can easily be verified that the use of the MPF using this proposal distribution is equivalent to using the SIS where resampling is performed at each iteration.

2. <u>AMPF</u>- The optimal choice of a proposal distribution is the posterior distribution $\hat{p}(x_k|z_{1:k})$, however it is often hard to obtain samples from the posterior. Using the AMPF [2] it is possible to obtain samples from a proposal distribution that is close to the posterior distribution. The posterior distribution can equivalently be written as

$$\begin{aligned} \hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) &\propto p(\boldsymbol{z}_k|\boldsymbol{x}_k)\sum_{i=1}^{N} w_{k-1}^{(i)} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}) \\ &= \sum_{i=1}^{N} w_{k-1}^{(i)} p(\boldsymbol{z}_k|\boldsymbol{x}_{k-1}^{(i)}) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k) = \sum_{i=1}^{N} p(i|\boldsymbol{z}_{1:k}) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k) \quad (2.53) \\ p(i|\boldsymbol{z}_{1:k}) &\propto w_{k-1}^{(i)} p(\boldsymbol{z}_k|\boldsymbol{x}_{k-1}^{(i)}) = \boldsymbol{w}_{k-1}^{(i)} \int p(\boldsymbol{z}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}) \boldsymbol{d}\boldsymbol{x}_k \quad (2.54) \end{aligned}$$

Using (2.53) we can sample from the proposal distribution $\hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ by first sampling i, which is known as the auxiliary variable from $\hat{p}(i|\boldsymbol{z}_{1:k})$, and then sampling $\boldsymbol{x}_k$ from $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k)$. Since (2.54) can not usually be evaluated analytically, it was suggested in [2] to use the approximation $\hat{p}(i|\boldsymbol{z}_{1:k}) \propto w_{k-1}^{(i)} p(\boldsymbol{z}_k|\boldsymbol{\mu}_k^{(i)})$ where $\boldsymbol{\mu}_k^{(i)} = \mathbb{E}[\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}]$. Furthermore since sampling from $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k)$ is not always possible, we follow [2] and sample from $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$ instead, where in all the cases which we consider $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$ is a Gaussian distribution from which samples are easily drawn. Therefore the AMPF proposal distribution is [2]

$$\pi(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \propto \sum_{i=1}^{N} \hat{p}(i|\boldsymbol{z}_{1:k}) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}) \tag{2.55}$$

Chapter 3

Background- Convex Optimization, Minimax Estimation, and Sensor Networks

In this chapter we review the necessary background for our work in robust estimation. As mentioned in the introduction, the estimation model is often subject to uncertainties which have to be accounted for in order to achieve satisfactory performance. The uncertainty may often be a result of unknown parameters in the estimation model, such as the covariance matrix or the positions of the sensors. The robust estimation approach therefore involves solving minimax optimization problems where one minimizes over the parameters of interest while maximizing over the uncertain parameters in their region of uncertainty. Such minimax problems are often difficult to solve unless they satisfy certain properties which we discuss in this chapter, along with convex optimization techniques which are also very useful when dealing with such problems.

Our work on robust estimation is primarily concerned with the linear Gaussian model which is tightly related to estimation models such as Gaussian Markov random fields (GMRF) and Gaussian processes (GP) which are often encountered in the context of sensor networks applications. We review in this chapter the *difference regret* and *ratio regret* estimators which are a recent advancement in the area of robust estimation in the linear Gaussian model with covariance matrix uncertainties. We also give a brief introduction to sensor network applications and explain the relevance of GMRF and GP models for such estimation problems.

3.1 Convex optimization

Convex optimization problems deal with minimization of a convex objective function over a convex domain, which is defined using a set of inequalities and affine equality constraints that define a convex set. Unlike general nonlinear problems, convex optimization problems can be solved efficiently using interior points methods in polynomial complexity [34]. When the objective function and inequality constraints are linear, the problem is known as linear programming and can be solved efficiently using the simplex algorithm [43].

Two important properties of convex functions are demonstrated by the conjugate function and the dual problem. Using the conjugate function one can describe a convex function using all the tangent surfaces to the function rather than the traditional pointwise description. Additionally using the conjugate function it is easy to obtain a lower bound for the function. The dual problem on the other hand can be used to obtain a lower bound on a minimization problem. An important property of convex minimization problems is that the lower bound is achieved with equality provided that certain conditions known as Slater's rules exist. Since the dual problem may sometime be easier to solve than the original problem (also known as the primal problem), this may prove to be significant in some problems. The solutions to the primal and dual problems are related through the Karush Kuhn Tucker (KKT) conditions. The KKT conditions are therefore useful for obtaining the solution to the primal problem once the solution to the dual problem is known. Furthermore in certain cases it is possible to obtain a solution to the problem by solving the KKT conditions directly.

Following is a short introduction to the basic concepts of convex optimization which is based on [32].

3.1.1 Sets

Lines and segments

Let $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ denote two points in $\mathbb{R}^n$ then the points $\boldsymbol{y}$ of the form

$$\boldsymbol{y} = \theta \boldsymbol{x}_1 + (1-\theta)\boldsymbol{x}_2, \tag{3.1}$$

where $\theta \in \mathbb{R}$, forms a line passing through the points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. When $0 \leq \theta \leq 1$ then $\boldsymbol{y}$ is the line segment between the points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$.

Affine sets

A set C is affine if for every two points $\boldsymbol{x}_1, \boldsymbol{x}_2 \in C$ and $\theta \in \mathbb{R}$ we have that $\theta\boldsymbol{x}_1+(1-\theta)\boldsymbol{x}_2 \in C$, i.e. C contains all the lines that pass through any two points in C. An example of an affine set is the solution set of linear equations $C = \Big\{\boldsymbol{x}|\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\Big\}$ where $\boldsymbol{b} \in \mathbb{R}^m$ and $\boldsymbol{A} \in \mathbb{R}^{m\times n}$.

3.1.2 Convex sets

A set C is convex if for every two points $\boldsymbol{x}_1, \boldsymbol{x}_2 \in C$ the line segment between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ lies entirely in C i.e. $\theta\boldsymbol{x}_1 + (1-\theta)\boldsymbol{x}_2 \in C$ $\forall 0 \leq \theta \leq 1$.

Cones

A set C is called a cone if for every $\boldsymbol{x} \in C$ and $\theta \geq 0$, we have $\theta\boldsymbol{x} \in C$. A cone C is also convex if for every $\boldsymbol{x}_1, \boldsymbol{x}_2 \in C$ and $\theta_1, \theta_2 \geq 0$, we have $\theta_1\boldsymbol{x}_1 + \theta_2\boldsymbol{x}_2 \in C$. Two important examples of convex cones are the norm cone which is the set $C = \{(\boldsymbol{x}, t)|\|\boldsymbol{x}\| \leq t\} \subseteq \mathbb{R}^{n+1}$ where $\boldsymbol{x} \in \mathbb{R}^n$, $t \in \mathbb{R}$, and the positive semidefinite cone $S^n_+ = \{\boldsymbol{X} \in S^n|\boldsymbol{X} \succeq 0\}$, where $S^n = \{\boldsymbol{X} \in \mathbb{R}^{n\times n}|\boldsymbol{X} = \boldsymbol{X}^T\}$, and $\boldsymbol{X} \succeq 0$ denote that $\boldsymbol{X}$ is positive semidefinite.

Proper cones and generalized inequalities

A proper cone $K \subseteq \mathbb{R}^n$ satisfies the following properties:

- It is a convex cone.
- It is closed.
- It is solid, i.e. it has nonempty interior.
- It is pointed, i.e. it contains no line. Alternatively if $\boldsymbol{x} \in K$ and $-\boldsymbol{x} \in K$ then $\boldsymbol{x} = 0$.

Using a proper cone one can define a generalized inequality such that for every $\boldsymbol{x}, \boldsymbol{y} \in K$ we have that $\boldsymbol{x} \preceq_K \boldsymbol{y} \Leftrightarrow \boldsymbol{y} - \boldsymbol{x} \in K$. Both the norm and positive semidefinite cones are proper cones and therefore can be used to define generalized inequalities.

Properties of convex sets

- The intersection of a finite of infinite number of convex sets is a convex set.
- The image $f(S) = \{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}|\boldsymbol{x} \in S\}$ where S is a convex set, $\boldsymbol{x} \in \mathbb{R}^n$ $A \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, is a convex set.

3.1.3 Convex functions

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is convex if its domain which is denoted by $\mathbf{dom} f$ is a convex set, and if for every two points $\boldsymbol{x}, \boldsymbol{y} \in \mathbf{dom} f$ and for $0 \leq \theta \leq 1$ we have

$$f(\theta \boldsymbol{x} + (1-\theta)\boldsymbol{y}) \leq \theta f(\boldsymbol{x}) + (1-\theta) f(\boldsymbol{y}). \tag{3.2}$$

Equivalent first and second order conditions for convex functions

In some cases it may be easier to determine whether a function is convex or not using the following conditions:

- If the gradient ∇f exists at each point in $\mathbf{dom} f$ and $\mathbf{dom} f$ is open, then f is convex if and only if $\mathbf{dom} f$ is convex and for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbf{dom} f$ we have that

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T (\boldsymbol{y} - \boldsymbol{x}). \tag{3.3}$$

- If the Hessian $\nabla^2 f$ exists at each point in $\mathbf{dom} f$ and $\mathbf{dom} f$ is open, then f is convex if and only if $\mathbf{dom} f$ is convex and the Hessian is positive semidefinite, i.e. for all $\boldsymbol{x} \in \mathbf{dom} f$ we have that $\nabla^2 f(\boldsymbol{x}) \succeq 0$ ($\boldsymbol{A} \succeq 0$ denotes that $\boldsymbol{A}$ is a positive semidefinite matrix).

- Let $g(x,t) = tf(x/t)$ where $f : \mathbb{R}^n \to \mathbb{R}$ and where $g : \mathbb{R}^{n+1} \to \mathbb{R}$ and is known as the perspective function of f, then if f is convex then g is also convex.

- A function $f(\boldsymbol{x})$ is convex if and only if its epigraph is a convex set, where the epigraph is defined as the set $\{(\boldsymbol{x}, t) | \boldsymbol{x} \in \mathrm{dom} f, f(\boldsymbol{x}) \leq t\}$.

Properties of convex functions

- The sublevel sets of a convex function form a convex set, i.e. let $f : \mathbb{R}^n \to \mathbb{R}$ then the set $C_\alpha = \{\boldsymbol{x} \in \mathbf{dom} f | f(\boldsymbol{x}) \leq \alpha\}$ is a convex set for any $\alpha \in \mathbb{R}$.

- A nonnegative weighted sum of convex functions is a convex function, i.e. let $f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})$ be convex functions, then $f(\boldsymbol{x}) = w_1 f_1(\boldsymbol{x}) + \cdots + w_n f_n(\boldsymbol{x})$ is a convex function for all $w_1, \ldots, w_n \geq 0$, and $\mathbf{dom} f = \bigcap_{i=1}^n \mathbf{dom} f_i$.

- The pointwise maximum and supremum is a convex function, i.e. let $f_1(\boldsymbol{x})$, $f_2(\boldsymbol{x})$ be two convex functions then $f(\boldsymbol{x}) = \max(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}))$ is also a convex function.

The conjugate function

A convex function $f(\boldsymbol{x})$ can be represented equivalently using the conjugate function $f^*(\boldsymbol{y})$ where

$$f^*(\boldsymbol{y}) = \max_{\boldsymbol{x}} (\boldsymbol{y}^T\boldsymbol{x} - f(\boldsymbol{x})), \tag{3.4}$$

and $f(\boldsymbol{x})$ can be obtained from $f^*(\boldsymbol{y})$ using

$$f(\boldsymbol{x}) = \max_{\boldsymbol{y}} (\boldsymbol{x}^T\boldsymbol{y} - f^*(\boldsymbol{y})). \tag{3.5}$$

The conjugate function replaces the locus of points representation given by $f(\boldsymbol{x})$ with an envelope of tangents representation given by $f^*(\boldsymbol{y})$ [25]. This can be seen from the example in Figure 3.1 where the dotted vertical segments represent values of $yx - f(x)$ and it can be seen that the line yx has to be shifted by the amount which is the maximum of $yx - f(x)$ in order to obtain a line which is tangent to $f(x)$ with the slope y. The intersection of the tangent line with slope y with the vertical axis is therefore at $-f^*(y)$.

Similarly for concave functions we have

$$f^*(\boldsymbol{y}) = \min_{\boldsymbol{x}} (\boldsymbol{y}^T\boldsymbol{x} - f(\boldsymbol{x})), \tag{3.6}$$

and $f(\boldsymbol{x})$ can be obtained from $f^*(\boldsymbol{y})$ using

$$f(\boldsymbol{x}) = \min_{\boldsymbol{y}} (\boldsymbol{x}^T\boldsymbol{y} - f^*(\boldsymbol{y})). \tag{3.7}$$

3.1.4 Convex optimization problems

The standard form of an optimization problem is composed of a scalar objective function $f_0(\boldsymbol{x})$ that is to be minimized, and a set of scalar inequalities $f_i(\boldsymbol{x}) \leq 0 \ i = 1, \ldots, m$ and equalities $h_i(\boldsymbol{x}) = 0 \ i = 1, \ldots, p$ that define the domain for the problem. The common notation for such a problem is

$$\min_{\boldsymbol{x}} f_0(\boldsymbol{x}) \tag{3.8}$$

subject to

$$f_i(\boldsymbol{x}) \leq 0, i = 1, \ldots, m$$

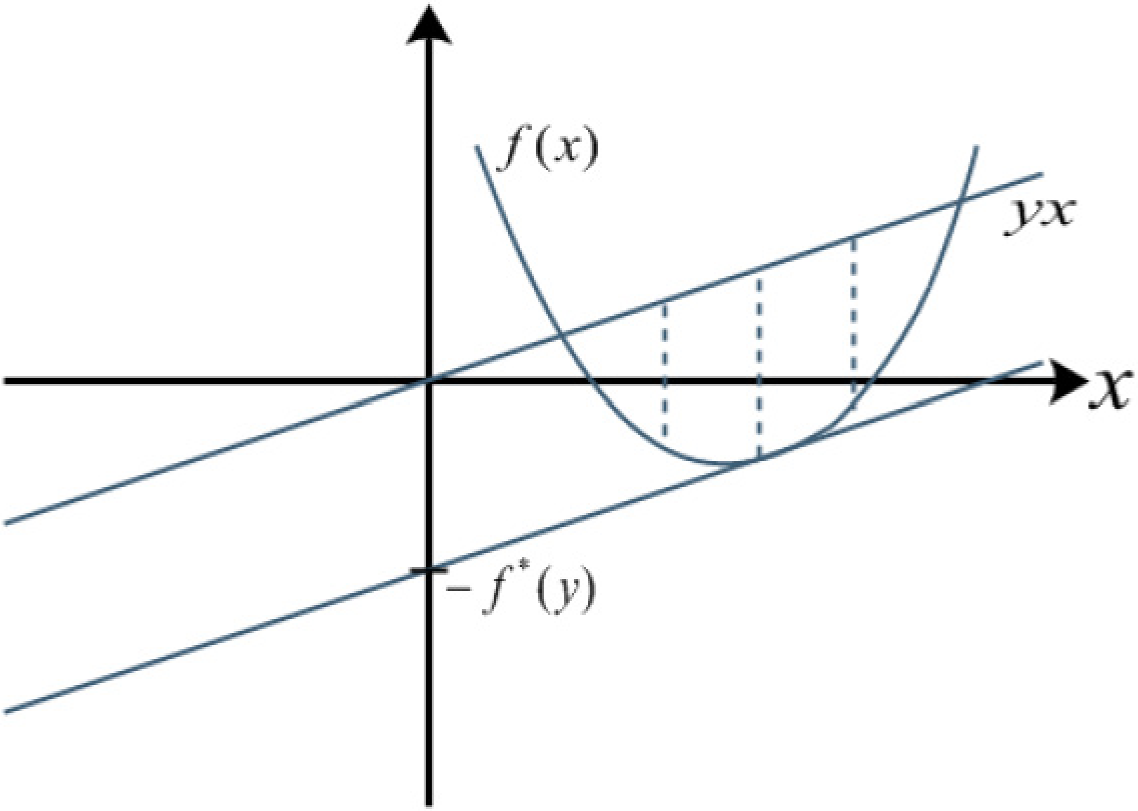

Figure 3.1: An example of the interpretation of the conjugate function

$$h_i(\boldsymbol{x}) \quad = \quad 0, i = 1, \ldots, p \tag{3.9}$$

and the domain is the intersection of all the domains of all the functions i.e.

$$\Big(\bigcap_{i=0}^{m} \mathbf{dom} f_i\Big) \bigcap \Big(\bigcap_{i=1}^{p} \mathbf{dom} h_i\Big) \tag{3.10}$$

If either the objective function or the domain are not convex then the problem is known as a nonlinear program and may be very difficult to solve. If the functions $f_i(\boldsymbol{x}) \leq 0$ $i = 0, \ldots, m$ are convex and the functions $h_i(\boldsymbol{x}) = 0$ $i = 1, \ldots, p$ are affine, i.e. of the form $h_i(\boldsymbol{x}) = \boldsymbol{a}_i^T \boldsymbol{x} - b_i$, then since the domain of the sublevel sets of of convex functions are convex and the intersection of convex sets is also convex, the domain of the problem is convex and the problem is a convex optimization problem. Common examples of such problems are linear programming, second order cone programming, and quadratic programming. One extension to the standard form given by the constraints in (3.9) is when the constraints functions $f_i(\boldsymbol{x})$

$i = 1, \ldots, m$ are vector valued and the inequalities are interpreted as generalized inequalities with respect to proper cones. Since proper cones are convex the domain of the problem is convex and the problem is a convex optimization problem. An examples of such a convex optimization problem is semidefinite programming where the semidefinite cone is considered. Second order cone programming can also be interpreted in this form where the norm cone is considered.

Linear programming

When all the functions in (3.8) and (3.9) are linear the problem is known as linear programming and can be solved efficiently using the simplex algorithm.

Second order cone programming

A second order cone optimization program is of the form

$$\min_{\boldsymbol{x}} \boldsymbol{f}^T \boldsymbol{x} \tag{3.11}$$

subject to

$$\begin{gathered} \|\boldsymbol{A}_i \boldsymbol{x} - \boldsymbol{b}_i\|_2 \leq \boldsymbol{c}_i^T \boldsymbol{x} + d_i, \ i = 1, \ldots, m \\ \boldsymbol{F}\boldsymbol{x} = \boldsymbol{g} \end{gathered} \tag{3.12}$$

where $\boldsymbol{x}, \boldsymbol{f} \in \mathbb{R}^n$, $\boldsymbol{A}_i \in \mathbb{R}^{n_i \times n}$, $\boldsymbol{F} \in \mathbb{R}^{p \times n}$, $\boldsymbol{g} \in \mathbb{R}^p$, $\boldsymbol{b}_i \in \mathbb{R}^{n_i}$, and $d_i \in \mathbb{R}$.

Semidefinite programming

A semidefinite programming optimization problem is of the form

$$\min_{\boldsymbol{x}} \boldsymbol{c}^T \boldsymbol{x} \tag{3.13}$$

subject to

$$\begin{gathered} x_1 \boldsymbol{F}_1 + \ldots + x_n \boldsymbol{F}_n + \boldsymbol{G} \preceq 0 \\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{gathered}$$

where $\boldsymbol{G}, \boldsymbol{F}_1, \ldots, \boldsymbol{F}_n$ are symmetric matrices, $x_1, \ldots, x_n$ denote the elements of $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{p \times n}$, $\boldsymbol{b} \in \mathbb{R}^p$, and the generalized inequality is with respect to the positive semidefinite cone. The following lemma is often used in order to transform optimization problems into the semidefinite programming form:

Lemma 1. *(Schur's Complement [26]): Let*

$$M = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{C} \end{bmatrix}$$

be a Hermitian matrix with $\boldsymbol{C} \succ 0$. *Then* $\boldsymbol{M} \succeq 0$ *if and only if* $\boldsymbol{A} - \boldsymbol{B}^T\boldsymbol{C}^{-1}\boldsymbol{B} \succeq 0$.

3.1.5 Duality

An important property of convex optimization problems is that they can be solved equivalently using the dual problem, provided that certain conditions known as Slater's conditions are satisfied. In order to obtain the dual problem for (3.8),(3.9) we first have to define the Lagrangian and the dual function. The Lagrangian is defined by

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{x}) + \sum_{i=1}^{p} \nu_i h_i(\boldsymbol{x}) \tag{3.14}$$

where the vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ are the lagrange multipliers associated with the inequality and equality constraints. The dual function is obtained by minimizing the Lagrangian over the domain given in (3.10) denoted by , D i.e.

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\boldsymbol{x} \in D} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}). \tag{3.15}$$

For any $\boldsymbol{\lambda} \geq 0$ and for any $\boldsymbol{\nu}$ we have $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq p^*$ where p^* denotes the optimal value for the primal problem given by (3.8),(3.9). In order to minimize the gap between the optimal value for the primal problem and the dual function we have to maximize the dual function over $\boldsymbol{\lambda} \geq 0$ and over any $\boldsymbol{\nu}$. The dual problem therefore takes the form

$$\max_{\boldsymbol{\lambda} \geq 0,\ \boldsymbol{\nu}} g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \tag{3.16}$$

Let d^* denote the optimal value for the dual problem (3.16) then if $d^* = p^*$ we say that strong duality holds. Strong duality does not hold in general, however a sufficient condition for strong duality is given by Slater's condition.

Slater's constraint

If the primal problem is convex i.e. $f_0, \dots, f_m$ are convex, and the problem is strictly feasible, i.e. there exists $\boldsymbol{x} \in D$ such that

$$f_i(\boldsymbol{x}) < 0,\ i = 1, \dots, m \qquad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{3.17}$$

then strong duality holds.

Nevertheless Slater's condition is a sufficient condition for strong duality however it is not necessary. For example, strong duality holds for a quadratic objective with a single quadratic inequality (without any other linear inequalities) even if the objective is not convex.

3.2 Minimax theory

Minimax theory deals with optimization problems of the form

$$\min_{v \in D} \max_{u \in C} K(u, v) \tag{3.18}$$

where C and D denote two non-empty sets and $K : C \times D \longrightarrow [-\infty, \infty]$. The solution of such optimization problems is not straightforward in the general case, however if the objective function K satisfies certain conditions, then there exist minimax theorems that can facilitate the solution. In particular if the objective function has a saddle point then it must be a solution of the minimax problem (although it may not be a unique solution).

Definition 1. *Let C and D denote two non-empty sets and let $K : C \times D \longrightarrow [-\infty, \infty]$, then a point $(\tilde{u}, \tilde{v}) \in C \times D$ is called a saddle point of K with respect to maximizing over C and minimizing over D if*

$$K(u, \tilde{v}) \leq K(\tilde{u}, \tilde{v}) \leq K(\tilde{u}, v),\ \forall u \in C,\ \forall v \in D$$

The following Lemma [25] states necessary and sufficient conditions for a point to be a saddle point:

Lemma 2. *Let K be any function from a non-empty product set $C \times D$ to $[-\infty, \infty]$. A point $(\tilde{u}, \tilde{v})$ is a saddle point of K (with respect to maximizing over C and minimizing over*

D) if and only if the maximum in the expression

$$\max_{u \in C} \min_{v \in D} K(u, v) \tag{3.19}$$

is attained at $\tilde{u}$, the minimum in the expression

$$\min_{v \in D} \max_{u \in C} K(u, v) \tag{3.20}$$

is attained at $\tilde{v}$, and these two extrema are equal.

An important Lemma that states sufficient conditions for a function to have a saddle point is given next

Lemma 3. *[25] Let C and D be two non-empty closed convex sets in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively, and let K be a continuous finite concave-convex function on $C \times D$ (i.e. concave in C and convex in D). If either C or D is bounded, one has*

$$\min_{v \in D} \max_{u \in C} K(u, v) = \max_{u \in C} \min_{v \in D} K(u, v) \tag{3.21}$$

Since (3.21) satisfies the necessary conditions required by Lemma 2 for a saddle point, then Lemma 3 gives sufficient conditions for the solution to be a saddle point. Most importantly if the conditions in Lemma 3 are satisfied then the order of the maximization and minimization can be interchanged, which can simplify the solution of the minimax problem in many cases.

3.3 Estimation with covariance matrix uncertainties

The classic solution to estimating a Gaussian random vector $\boldsymbol{x}$ that is observed through a linear transformation and corrupted by Gaussian noise is obtained using the minimum squared error (MMSE) estimator which assumes full knowledge of the covariance matrix of the random vector $\boldsymbol{x}$ and the covariance matrix of the observation noise. Specifically let

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w}, \tag{3.22}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is the observation, $\boldsymbol{H} \in \mathbb{R}^{n \times m}$, and $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{w} \in \mathbb{R}^n$ are independent zero mean Gaussian random vectors with covariance matrices $\boldsymbol{C_x}$ and $\boldsymbol{C_w}$ respectively, then given an observation vector $\boldsymbol{y}$ the MMSE estimate of $\boldsymbol{x}$ takes the form [28]

$$\hat{\boldsymbol{x}} = \boldsymbol{C_x}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{C_x}\boldsymbol{H}^T + \boldsymbol{C_w})^{-1}\boldsymbol{y}. \tag{3.23}$$

In many applications it is reasonable to expect that the estimate of the covariance matrix of the observation noise is accurate. However the estimate of the covariance matrix of $\boldsymbol{x}$ may often be highly inaccurate and lead to severe performance degradation when using the MMSE estimator. Therefore in practice it is necessary to require the filter to be robust with respect to such uncertainty. The common approach to achieve such robustness is through the use of a minimax estimator which minimizes the worst case performance over some criterion in the region of uncertainty [30, 31].

One such performance measure is the mean squared error (MSE), where the estimator is chosen such that the worst case MSE in the region of uncertainty of the covariance matrix of $\boldsymbol{x}$ is minimized. However as was noted in [28] this choice may be too pessimistic and therefore the performance of an estimator designed this way may be unsatisfactory. Instead it is proposed in [28] to minimize the worst case *difference regret* which is defined as the difference between the MSE when using a linear estimator of the form $\hat{\boldsymbol{x}} = \boldsymbol{G}\boldsymbol{y}$ and the MSE when using the MMSE estimator matched to a covariance matrix $\boldsymbol{C_x}$, where $\boldsymbol{G}$ is a matrix with the appropriate dimensions. The motivation for this choice is that the worst case difference regret criterion is less pessimistic than the worst case MSE criterion. Similarly the *ratio regret* estimator proposed in [29], minimized the worst case ratio regret which is defined as the ratio between the MSE when using a linear estimator of the form $\hat{\boldsymbol{x}} = \boldsymbol{G}\boldsymbol{y}$ and the MSE when using the MMSE estimator matched to a covariance matrix $\boldsymbol{C_x}$. The motivation for the ratio regret estimator is similar to the difference regret where the MSE is measured in decibels. The difference and ratio regret estimators presented in [28, 29] assume that the the eigenvector matrix of $\boldsymbol{C_x}$ is known and is identical to the eigenvector matrix of $\boldsymbol{H}^T\boldsymbol{C}_{\boldsymbol{w}}^{-\boldsymbol{1}}\boldsymbol{H}$, which is also called the *joint diagonalizable matrices* assumption. Furthermore the region of uncertainty is expressed using upper and lower bounds on each of the eigenvalues of $\boldsymbol{C_x}$.

3.3.1 Minimax regret estimators

The aim of the minimax regret estimators is to achieve robustness to the uncertainty in the covariance matrix by finding a linear estimator of the form $\hat{\boldsymbol{x}} = \boldsymbol{G}\boldsymbol{y}$ that minimizes the worst performance of the regret in the region of uncertainty of the covariance matrix $\boldsymbol{C_x}$. Specifically let $\Re(\boldsymbol{C_x}, \boldsymbol{G})$ denote the regret, and let $\Omega \subset S_+$, where S_+ denotes the set of positive semidefinite matrices, denote the region of uncertainty of $\boldsymbol{C_x}$. The minimax estimator is then obtained by solving

$$\boldsymbol{G} = \arg\min_{\boldsymbol{G}} \max_{\boldsymbol{C_x}\in\Omega} \Re(\boldsymbol{C_x}, \boldsymbol{G}). \tag{3.24}$$

The difference and ratio regret criteria are defined as the difference and the ratio between the MSE when using an estimator of the form $\hat{\boldsymbol{x}} = \boldsymbol{Gy}$ and the MSE when using the MMSE estimator. The MSE when estimating $\boldsymbol{x}$ using a linear estimator of the form $\hat{\boldsymbol{x}} = \boldsymbol{Gy}$ is given by [28]

$$\mathbb{E}(\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2) = Tr(\boldsymbol{G}\boldsymbol{C_w}\boldsymbol{G}^T) + Tr(\boldsymbol{C_x}(\boldsymbol{I} - \boldsymbol{GH})^T(\boldsymbol{I} - \boldsymbol{GH})) \tag{3.25}$$

where $Tr(\boldsymbol{A})$ denotes the trace of the square matrix $\boldsymbol{A}$, and $\boldsymbol{I}$ denotes the identity matrix with the appropriate dimensions. The MSE when using the MMSE estimator takes the form [28]

$$MSE^0 = Tr((\boldsymbol{H}^T\boldsymbol{C_w^{-1}}\boldsymbol{H} + \boldsymbol{C_x^{-1}})^{-1}). \tag{3.26}$$

Both the difference and ratio estimators presented in [28, 29] assume that the region of uncertainty Ω is expressed as uncertainties in the eigenvalues of the covariance matrix $\boldsymbol{C_x}$ assuming that the eigenvectors are known. Specifically, let $\boldsymbol{V}$ denote the eigenvectors matrix of $\boldsymbol{C_x}$, and let u_i and ℓ_i denote upper and lower bounds on the eigenvalues δ_i $i = 1, \ldots, m$, then $\Omega = \{\boldsymbol{V\Delta V}^T | \boldsymbol{\Delta} = \text{diag}(\delta_1, \ldots, \delta_m), \ell_i \leq \delta_i \leq u_i\}$.

Difference regret estimator

The difference regret is defined as the difference between (3.25) and (3.26)

$$\begin{aligned} \Re(\boldsymbol{C_x}, \boldsymbol{G}) &= \mathbb{E}(\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2) - MSE^0 \\ &= Tr(\boldsymbol{C_x}(\boldsymbol{I} - \boldsymbol{GH})^T(\boldsymbol{I} - \boldsymbol{GH})) \\ &+ Tr(\boldsymbol{G}\boldsymbol{C_w}\boldsymbol{G}^T) - Tr((\boldsymbol{H}^T\boldsymbol{C_w^{-1}}\boldsymbol{H} + \boldsymbol{C_x^{-1}})^{-1}). \end{aligned} \tag{3.27}$$

Assuming that $\boldsymbol{H}^T\boldsymbol{C_w^{-1}}\boldsymbol{H} = \boldsymbol{V\Lambda V}^T$ where $\boldsymbol{\Lambda}$ is a diagonal matrix with the diagonal elements λ_i $i = 1, \ldots, m$, it is shown in [28] that

$$\boldsymbol{G} = \boldsymbol{VD\Lambda}^{-1}\boldsymbol{V}^T\boldsymbol{H}^T\boldsymbol{C_w^{-1}}, \tag{3.28}$$

where $\boldsymbol{D}$ is an $m \times m$ diagonal matrix with diagonal elements

$$d_i = 1 - \frac{1}{\sqrt{(1+\lambda_i\zeta_i)^2 - \lambda_i^2\epsilon_i^2}}, \tag{3.29}$$

and where $\zeta_i = (u_i + \ell_i)/2$ and $\epsilon_i = (u_i - \ell_i)/2$.

The difference regret estimator can also be interpreted as the MMSE estimator (3.23) with an equivalent covariance matrix $\boldsymbol{C_x} = \boldsymbol{V}\boldsymbol{X}\boldsymbol{V}^T$ where $\boldsymbol{X}$ is a diagonal matrix with the diagonal elements

$$x_i = \frac{1}{\lambda_i}\Big(\sqrt{(1+\lambda_i\zeta_i)^2 - \lambda_i^2\epsilon_i^2} - 1\Big). \tag{3.30}$$

Ratio regret estimator

The ratio regret is defined as the ratio between (3.25) and (3.26)

$$\Re(\boldsymbol{C_x}, \boldsymbol{G}) = \frac{Tr(\boldsymbol{G}\boldsymbol{C_w}\boldsymbol{G}^T) + Tr(\boldsymbol{C_x}(\boldsymbol{I} - \boldsymbol{G}\boldsymbol{H})^T(\boldsymbol{I} - \boldsymbol{G}\boldsymbol{H}))}{Tr((\boldsymbol{H}^T\boldsymbol{C_w^{-1}}\boldsymbol{H} + \boldsymbol{C_x^{-1}})^{-1})} \tag{3.31}$$

Using (3.31) in (3.24) it is shown in [29] that the optimal $\boldsymbol{G}$ can be obtained by solving,

$$t(\gamma) = \min_{\boldsymbol{G}} \max_{\boldsymbol{C_x} \in \Omega} \Re_\gamma(\boldsymbol{C_x}, \boldsymbol{G}) \tag{3.32}$$

where

$$\Re_\gamma(\boldsymbol{C_x}, \boldsymbol{G}) = Tr(\boldsymbol{G}\boldsymbol{C_w}\boldsymbol{G}^T) + Tr(\boldsymbol{C_x}(\boldsymbol{I} - \boldsymbol{G}\boldsymbol{H})^T(\boldsymbol{I} - \boldsymbol{G}\boldsymbol{H})) - \gamma Tr((\boldsymbol{H}^T\boldsymbol{C_w^{-1}}\boldsymbol{H} + \boldsymbol{C_x^{-1}})^{-1}) \tag{3.33}$$

where it is shown that $t(\gamma)$ is a continuous and strictly decreasing function of $\gamma \geq 0$, and γ should be chosen such that $t(\gamma) = 0$. This can be achieved by perfoming a line search over γ.

Assuming that $\boldsymbol{H}^T\boldsymbol{C_w^{-1}}\boldsymbol{H} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ where $\boldsymbol{\Lambda}$ is a diagonal matrix with the diagonal elements λ_i $i = 1, \ldots, m$, it is shown in [28] that the optimal solution takes the form

$$\boldsymbol{G} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{\Lambda}^{-1}\boldsymbol{V}^T\boldsymbol{H}^T\boldsymbol{C_w^{-1}} \tag{3.34}$$

where $\boldsymbol{D}$ is an $m \times m$ diagonal matrix with diagonal elements d_i that are given by

$$d_i = \begin{cases} 1 - \frac{\sqrt{\gamma}}{\sqrt{(1+\lambda_i\zeta_i)^2 - \lambda_i^2\epsilon_i^2}}, & 1 \leq \gamma \leq \gamma_i^0 \\ \frac{\lambda_i(\zeta_i - \epsilon_i)}{1+\lambda_i(\zeta_i-\epsilon_i)}, & \gamma \geq \gamma_i^0 \end{cases} \tag{3.35}$$

where

$$\gamma_i^0 = \frac{1+\lambda_i(\zeta_i+\epsilon_i)}{1+\lambda_i(\zeta_i-\epsilon_i)}, \tag{3.36}$$

and where γ is chosen such that $\sum_{i=1}^{n} t_i(\gamma) = 0$, where $t_i(\gamma)$ is given by.

$$t_i(\gamma) = \begin{cases} \frac{1}{\lambda_i} - \frac{2\sqrt{\gamma}}{\lambda_i\sqrt{(1+\lambda_i\zeta_i)^2 - \lambda_i^2\epsilon_i^2}} + \frac{\gamma(\lambda_i^2(\epsilon_i^2 - \zeta_i^2)+1)}{\lambda_i((1+\lambda_i\zeta_i)^2 - \lambda_i^2\epsilon_i^2)}, & 1 \leq \gamma \leq \gamma_i^0 \\ \frac{(\gamma-1)(\epsilon_i - \zeta_i)}{1+\lambda_i(\zeta_i-\epsilon_i)}, & \gamma \geq \gamma_i^0 \end{cases} \tag{3.37}$$

3.4 Sensor networks

A sensor network is comprised of many autonomous sensors that are spread in an environment, collecting data and communicating with each other [41]. Each sensor node also has some computational resources and can process the data that it acquires and the transmission that it receives from other sensors independently. Since the sensors are usually battery powered, a major concern in such applications is reducing the energy consumption, especially the energy spent on communication between the sensors, which is significantly larger than any other cause for energy consumption. The straightforward approach to estimation in sensor networks is to transmit all the data collected by the sensors to a centralized location and perform the estimation there, however this approach is very inefficient energy wise since an enormous amount of data has to be transmitted. Instead the more energy efficient approach is to transmit messages between the sensor nodes and have the sensors perform the estimation collectively.

Estimation in sensor networks is concerned with estimating the distributions, or statistical quantities such as the mean or variance of random variables that are associated with the sensor network's nodes, and may be correlated or statistically dependent. Such a structure can be efficiently represented using a graphical model (GM). A GM is comprised of a set of vertices representing the random variables, and a set of edges representing a statistical dependency between the two random variables that it connects. When the GM contains no cycles then the estimation problem can be solved efficiently using the belief propagation (BP) algorithm (also known as the sum product algorithm) [42]. The BP algorithm involves transmitting messages between adjacent nodes until convergence is obtained. When the GM does contain cycles then it is possible to obtain exact inference by transforming the graph to the junction tree form [36], however the computational cost is exponential in the size of the added junction nodes, therefore in many practical applications this cost is unacceptable [35, 37]. There also exist approximate inference algorithms such as the loopy BP where BP is used similarly to the cycle free case, however the algorithm is not guaranteed to converge and if it does converge then the inferences should be viewed only as approximations to their true values [35]. For the Gaussian GM (also known as Gaussian Markov random fields) case there also exist iterative algorithms that perform a distributed computation of the MMSE estimation [37, 38]. Such methods have been shown to converge faster than loopy BP, and there exist conditions under which convergence is guaranteed.

Another statistical model which can used to capture the correlation structure of the nodes in a sensor network is the Gaussian Process (GP) [44]. A GP extends the discrete set of sites that form a Gaussian random vector into a continuous domain, where every set of points in the domain forms a Gaussian random vector. This can be useful in cases where there is uncertainty in the position of the sensors or in order to determine the optimal placement of the sensors [45].

3.4.1 Gaussian Markov random fields

A Gaussian Markov random field (GMRF) assumes that the random variables in the GM are jointly Gaussian with a covariance matrix that is parameterized by a form that is determined by the edges of the graph.

Let $g = (\nu, \varepsilon)$ denote the set of vertices ν and the set of edges ε in the graph g. The set of random variables is $X = \{x_s | s \in \nu\}$. The Markov property states that given the set of neighboring nodes to s (the nodes that share an edge with s and are denoted by $\Gamma(s)$), the distribution of the random variable x_s is independent of all the other random variables in the graph, i.e.

$$p(x_s | x_{\nu \backslash s}) = p(x_s | x_{\Gamma(s)}). \tag{3.38}$$

The Hammersley-Clifford theorem stipulates that for any Markov random field (MRF) (where GMRF is a special case of MRF) the distribution of the random variable x_s is a Gibbs distribution

$$p(x_s | x_{\nu \backslash s}) = \frac{1}{Z} e^{-\sum_{c \in C_s} V_c(X)} \tag{3.39}$$

where C_s denotes the set of all cliques that contain node s, where a clique is defined as any group of nodes that are all connected to s and can contain between one node to the number of nodes connected to s, $V_c(X)$ is known as the potential function for clique c, and Z is a normalization constant.

The choice of the clique types and the potential functions clearly have an important effect on the MRF distribution. For a GMRF only single and pairwise cliques containing either one or two nodes are considered. The potential function for a single node takes the form $V_s^{(1)}(x_s) = -x_s^2 P_s$, and for the two node case the potential function connecting notes s and t takes the form

$$V_{s,t}^{(2)}(x_s, x_t) = \; -[x_s \quad x_t] \begin{bmatrix} P_{s(t)} & P_{s,t} \\ P_{t,s} & P_{t(s)} \end{bmatrix} \begin{bmatrix} x_s \\ x_t \end{bmatrix} \tag{3.40}$$

where $P_{s,t}$ is the (s,t) element of the inverse covariance matrix $\boldsymbol{P}$, and where $P_{s(t)}$ are chosen such that $P_{s,s} = P_s + \sum_{t\in\Gamma(s)} P_{s(t)}$. The pairwise potential functions $V_{s,t}^{(2)}(x_s, x_t)$ do not necessarily have to be valid PDFs, i.e. the inverse covariance matrix of the pairwise potential functions

$$\begin{bmatrix} P_{s(t)} & P_{s,t} \\ P_{t,s} & P_{t(s)} \end{bmatrix} \tag{3.41}$$

do not have to be positive semidefinite [51], however the inverse covariance matrix $\{P_{s,t}\}_{s,t\in\nu}$ must be positive semidefinite. The distribution of the random vector X then takes the form

$$p(X) = \frac{1}{\bar{Z}} e^{-\sum_{(s,t)\in\varepsilon} V_{s,t}^{(2)}(x_s,x_t) - \sum_{s\in\nu} V_s^{(1)}(x_s)} \tag{3.42}$$

where $\bar{Z}$ is a normalization constant.

3.4.2 Gaussian processes

A GP is defined as follows:

Definition 2. *A Gaussian Process is a collection of variables, any finite number of which have a joint Gaussian distribution.*

Clearly any the elements of a Gaussian random vector would be classified as a GP, however we can obtain a much richer statistical model using this definition. To show this we first consider a distribution that is defined over a function. Let $f(\boldsymbol{x})$ denote a real process, then the mean function $m(\boldsymbol{x})$ and covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ take the form

$$\begin{aligned} m(\boldsymbol{x}) &= \mathbb{E}[f(\boldsymbol{x})] \quad (3.43) \\ k(\boldsymbol{x}, \boldsymbol{x}') &= \mathbb{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x'}) - m(\boldsymbol{x'}))] \quad (3.44) \end{aligned}$$

where $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ where $\mathcal{X}$ is the domain of $f(\boldsymbol{x})$, e.g. $\mathbb{R}^D$. For every $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ we have that the covariance matrix of the random variables $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)$ is positive semidefinite. This is clear because for every $z_1, \ldots, z_n \in \mathbb{R}$ we have [47]

$$\sum_{i,j=1}^{n} z_i z_k k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbb{E}\Big[\Big|\sum_{i=1}^{n} z_i (f(\boldsymbol{x}_i) - m(\boldsymbol{x}_i))\Big|^2\Big] \geq 0 \quad (3.45)$$

A simple example of a Gaussian process can be obtained using the process $f(\boldsymbol{x}) = \phi(\boldsymbol{x})^T \boldsymbol{w}$, where $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_p)$, and where $\phi(\boldsymbol{x}) : \mathbb{R}^D \rightarrow \mathbb{R}^N$ is a projection into some high dimension space. It is clear that this choice of $f(\boldsymbol{x})$ satisfies definition 2 such that for any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ we have that $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)$ are mutually Gaussian. The mean and covariance functions for such a GP are:

$$\begin{aligned} m(\boldsymbol{x}) &= \phi(\boldsymbol{x})^T \mathbb{E}[\boldsymbol{w}] = 0 \quad (3.46) \\ k(\boldsymbol{x}, \boldsymbol{x'}) &= \phi(\boldsymbol{x})^T \mathbb{E}[\boldsymbol{w}\boldsymbol{w}^T] \phi(\boldsymbol{x'}) = \phi(\boldsymbol{x})^T \boldsymbol{\Sigma}_p \phi(\boldsymbol{x'}) \quad (3.47) \end{aligned}$$

It should be noted that $m(\boldsymbol{x})$ and $k(\boldsymbol{x}, \boldsymbol{x'})$ are scalars regardless of N the dimensionality of the projection $\phi(\boldsymbol{x})$. Therefore we may even have $N \rightarrow \infty$ since we only need to specify the kernel function $k(\boldsymbol{x}, \boldsymbol{x'})$ in order to completely specify the GP. The only condition that the kernel function must satisfy is that it is positive semidefinite, i.e. it satisfies that for any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ and any $z_1, \ldots, z_n \in \mathbb{R}$ we have

$$\sum_{i,j=1}^{n} z_i z_k k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0 \quad (3.48)$$

A major distinction between different kernel functions is whether the kernel function is stationary or not, i.e. whether it can be expressed as a function of $\boldsymbol{r} = \|\boldsymbol{x} - \boldsymbol{x}'\|$, where $r = \|\boldsymbol{x} - \boldsymbol{x}'\|$, and $\tilde{\boldsymbol{x}} = [1\boldsymbol{x}^T]^T$. Table 3.1 shows different stationary and non-stationary choices for covariance functions. A method to estimate a nonstationary kernel function using a set of estimates of the covariance matrix at several known locations was presented in [48].

Table 3.1: Different covariance functions

covariance function	expression	stationary
constant	σ_0^2	$\surd$
linear	$\sum_{d=1}^{D} \sigma_d^2 x_d \cdot x'_d$	
polynomial	$(\boldsymbol{x} \cdot \boldsymbol{x}' + \sigma_0^2)^p$	
squared exponential	$\exp\left(-\frac{r^2}{2\ell^2}\right)$	$\surd$
exponential	$\exp\left(-\frac{r}{\ell}\right)$	$\surd$
γ-exponential	$\exp\left(-\frac{r}{\ell}\right)^{\gamma}$	$\surd$
rational quadratic	$(1 + \frac{r^2}{2\alpha\ell^2})^{-\alpha}$	$\surd$
neural network	$\arcsin\left(\frac{2\tilde{\boldsymbol{x}}^T \boldsymbol{\Sigma} \tilde{\boldsymbol{x}}'}{\sqrt{(1+2\tilde{\boldsymbol{x}}^T \boldsymbol{\Sigma} \tilde{\boldsymbol{x}})(1+2\tilde{\boldsymbol{x}}'^T \boldsymbol{\Sigma} \tilde{\boldsymbol{x}}')}}\right)$	

Chapter 4

Tighter IFGT Performance Bounds and Nonlinear Filtering using the IFGT

In this chapter we first derive new performance bounds for the IFGT which are significantly tighter than the previous bounds and unlike the old bounds agree with our expectation that the error decreases as the truncation order increases. The new bounds also leads to a new definition of the IFGT parameters. Subsequently we develop a new proposal distribution which uses Monte Carlo integration and importance sampling to approximate the AMPF integral in (2.54) and use it to perform nonlinear filtering, where the IFGT is used to reduce the computational complexity of the filtering algorithm. The IFGT parameters in this case have to be chosen experimentally such that a compromise between accuracy and computation time is obtained, however we demonstrate that the new bounds give new insight into the effect that the IFGT parameters have on the error, thus facilitating their experimental choice.

4.1 New error bounds for the IFGT

In this section we derive new upper error bounds for the IFGT. The first bound assumes that the source points and their weights are given, whereas the second bound applies to the case where only the radius of a ball containing the source points is known.

Theorem 1. *The error E_T due to truncating the series (2.10) after pth order satisfies the* *bound*

$$|E_T| \le \kappa \max_{\|\boldsymbol{\tau}\|>0} e^{-\|\boldsymbol{\tau}\|^2/2}\Theta_1(\|\boldsymbol{\tau}\|) \tag{4.1}$$

where

$$\Theta_1(\|\boldsymbol{\tau}\|) = \sum_{j=1}^{N} |q_j| e^{-\|\boldsymbol{\xi}_j\|^2/2+\|\boldsymbol{\xi}_j\|\|\boldsymbol{\tau}\|} - \sum_{n=0}^{p-1} d_n \|\boldsymbol{\tau}\|^n, \tag{4.2}$$

$$d_n = \sum_{j=1}^{N} \frac{1}{n!} |q_j| e^{-\|\boldsymbol{\xi}_j\|^2/2} \|\boldsymbol{\xi}_j\|^n \tag{4.3}$$

and where $\boldsymbol{\xi}_j = \boldsymbol{\Delta s}_j/\sigma$, $\boldsymbol{\tau} = \boldsymbol{\Delta t}/\sigma$. Furthermore the global maximum in (4.1) can be found by performing a line search on $\|\boldsymbol{\tau}\|$, and the the upper bound decreases as the truncation order p increases.

Proof. The truncation error in (2.10) can be bounded by,

$$|E_T| \le \kappa \sum_{j=1}^{N} |q_j| |e^{-\|\boldsymbol{\xi}_j\|^2/2} e^{-\|\boldsymbol{\tau}\|^2/2} \sum_{n=p}^{\infty} \frac{1}{n!} (\boldsymbol{\xi}_j \cdot \boldsymbol{\tau})^n|$$

$$\le \kappa e^{-\|\boldsymbol{\tau}\|^2/2} \sum_{j=1}^{N} |q_j| e^{-\|\boldsymbol{\xi}_j\|^2/2} \sum_{n=p}^{\infty} \frac{1}{n!} (\|\boldsymbol{\xi}_j\|\|\boldsymbol{\tau}\|)^n \tag{4.4}$$

$$= \kappa e^{-\|\boldsymbol{\tau}\|^2/2} \sum_{j=1}^{N} |q_j| e^{-\|\boldsymbol{\xi}_j\|^2/2} (e^{\|\boldsymbol{\xi}_j\|\|\boldsymbol{\tau}\|} - \sum_{n=0}^{p-1} \frac{1}{n!} (\|\boldsymbol{\xi}_j\|\|\boldsymbol{\tau}\|)^n) \tag{4.5}$$

$$\le \kappa \max_{\|\boldsymbol{\tau}\|>0} e^{-\|\boldsymbol{\tau}\|^2/2} \Theta_1(\|\boldsymbol{\tau}\|) \tag{4.6}$$

where (4.4) follows from the Cauchy-Schwartz inequality, (4.5) follows since the infinite sum in (4.4) is the tail of the Taylor series expansion of the exponent function, and (4.6) is obtained by maximizing over $\|\boldsymbol{\tau}\|$. In order to show that the maximization can be obtained using a line search we show that there exists a maximum and it is either a global maximum or one of several local maxima that have the same objective value.

Let $\Gamma_1(\|\boldsymbol{\tau}\|) = e^{-\|\boldsymbol{\tau}\|^2/2}\Theta_1(\|\boldsymbol{\tau}\|)$. We note that $\Gamma_1(0) = 0$, and since

$$0 \le \Gamma_1(\|\boldsymbol{\tau}\|) \le \sum_{j=1}^{N} |q_j| e^{-\|\boldsymbol{\tau}\|^2/2} e^{\|\boldsymbol{\xi}_j\|\|\boldsymbol{\tau}\|} = \sum_{j=1}^{N} |q_j| e^{-\|\boldsymbol{\tau}\|(\|\boldsymbol{\tau}\|/2 - \|\boldsymbol{\xi}_j\|)} \longrightarrow 0 \tag{4.7}$$

as $\|\boldsymbol{\tau}\| \to \infty$ for any bounded $\{\|\boldsymbol{\xi}_j\|\}_{j=1}^N$, we have $\Gamma_1(\infty) = 0$. Therefore since $\Gamma_1(\|\boldsymbol{\tau}\|)$ is nonnegative there must be a maximum. Next we show that we can find the maximum in (4.6) using a line search. Let a be such that $\frac{\partial}{\partial\|\boldsymbol{\tau}\|}\Gamma_1(\|\boldsymbol{\tau}\|)\Big|_{\|\boldsymbol{\tau}\|=a} = 0$ and assume that there exists another point b such that $\frac{\partial}{\partial\|\boldsymbol{\tau}\|}\Gamma_1(\|\boldsymbol{\tau}\|)\Big|_{\|\boldsymbol{\tau}\|=b} = 0$. Using $\Gamma_1(\|\boldsymbol{\tau}\|) = e^{-\|\boldsymbol{\tau}\|^2/2}\Theta_1(\|\boldsymbol{\tau}\|)$ we obtain that

$$\frac{\partial}{\partial\|\boldsymbol{\tau}\|}\Gamma_1(\|\boldsymbol{\tau}\|) = (-\|\boldsymbol{\tau}\|\Theta_1(\|\boldsymbol{\tau}\|) + \frac{\partial}{\partial\|\boldsymbol{\tau}\|}\Theta_1(\|\boldsymbol{\tau}\|))e^{-\|\boldsymbol{\tau}\|^2/2} = 0, \text{ for } \|\boldsymbol{\tau}\| = a \text{ or } \|\boldsymbol{\tau}\| = b, \tag{4.8}$$

or equivalently

$$\frac{\frac{\partial}{\partial\|\boldsymbol{\tau}\|}\Theta_1(\|\boldsymbol{\tau}\|)}{\Theta_1(\|\boldsymbol{\tau}\|)} = \|\boldsymbol{\tau}\| \text{ for } \|\boldsymbol{\tau}\| = a \text{ or } \|\boldsymbol{\tau}\| = b. \tag{4.9}$$

Equation (4.9) is a differential equation and is therefore equivalent to

$$\Theta_1(\|\boldsymbol{\tau}\|) = Ce^{\|\boldsymbol{\tau}\|^2/2}, \text{ for } \|\boldsymbol{\tau}\| = a \text{ or } \|\boldsymbol{\tau}\| = b, \tag{4.10}$$

where C is some constant. Using $\|\boldsymbol{\tau}\| = a$ and $\|\boldsymbol{\tau}\| = b$ in (4.10) we have that $C = \Gamma_1(a) = \Gamma_1(b)$. Therefore if $a = b$ then we have a global maximum, and if $a \neq b$ then a and b are local maxima that have the same objective value. This result can be easily extended to any number of maxima. Additionally since the infinite sum in (4.4) decreases as p increases, so does $\Theta_1(\|\boldsymbol{\tau}\|)$. Therefore the upper bound decreases as the truncation order increases. □

In order to derive the second bound we first have to prove the following two propositions.

Proposition 1. *Let,*

$$\Gamma_2(\|\boldsymbol{\tau}\|, r_0) = e^{-\|\boldsymbol{\tau}\|^2/2}\Theta_2(\|\boldsymbol{\tau}\|, r_0) \tag{4.11}$$

where

$$\Theta_2(\|\boldsymbol{\tau}\|, r_0) = e^{-r_0^2/2}(e^{r_0\cdot\|\boldsymbol{\tau}\|} - \sum_{n=0}^{p-1}\frac{1}{n!}(r_0 \cdot \|\boldsymbol{\tau}\|)^n) \tag{4.12}$$

then for a given $r_0 > 0$ the solution to

$$a = \mathit{arg}\max_{\|\boldsymbol{\tau}\|>0} \Gamma_2(\|\boldsymbol{\tau}\|, r_0) \tag{4.13}$$

satisfies $a > r_0$.

Proof. Taking the derivative of $\Gamma_2(\|\boldsymbol{\tau}\|, r_0)$ with respect to $\|\boldsymbol{\tau}\|$ we obtain

$$\frac{\partial}{\partial\|\boldsymbol{\tau}\|}\Gamma_2(\|\boldsymbol{\tau}\|, r_0) = \left((r_0 - \|\boldsymbol{\tau}\|)\sum_{n=p}^{\infty}\frac{1}{n!}(r_0\|\boldsymbol{\tau}\|)^n + \frac{r_0^p\|\boldsymbol{\tau}\|^{p-1}}{(p-1)!}\right)e^{-\|\boldsymbol{\tau}\|^2/2-r_0^2/2},$$

and therefore we have that $\frac{\partial}{\partial\|\boldsymbol{\tau}\|}\Gamma_2(\|\boldsymbol{\tau}\|, r_0) = 0$ only if $\|\boldsymbol{\tau}\| > r_0$ or equivalently $a > r_0$. □

Proposition 2. *If $\|\boldsymbol{\tau}\| > r_0 > 0$, it follows that*

$$\frac{\partial}{\partial r_0}\Theta_2(\|\boldsymbol{\tau}\|, r_0) > 0 \tag{4.14}$$

Proof. Taking the derivative of $\Theta_2(\|\boldsymbol{\tau}\|, r_0)$ with respect to r_0 we obtain

$$\frac{\partial}{\partial r_0}\Theta_2(\|\boldsymbol{\tau}\|, r_0) = \left((\|\boldsymbol{\tau}\| - r_0)\sum_{n=p}^{\infty}\frac{1}{n!}(r_0\|\boldsymbol{\tau}\|)^n + \frac{\|\boldsymbol{\tau}\|^p r_0^{p-1}}{(p-1)!}\right)e^{-r_0^2/2}$$

which is strictly positive for every $\|\boldsymbol{\tau}\| > r_0 > 0$. □

Theorem 2. *Let N_B sources with weights $\{q_j\}_{j=1}^{N_B}$ lie in a ball with radius $r_0\sigma$, then the error E_T due to truncating the series (2.10) after pth order satisfies the bound*

$$|E_T| \le Q_B \epsilon_p^{IFGT_3}(r_0) \tag{4.15}$$

where

$$\epsilon_p^{IFGT_3}(r_0) = \kappa \max_{\|\boldsymbol{\tau}\|>0} e^{-\|\boldsymbol{\tau}\|^2/2}\Theta_2(\|\boldsymbol{\tau}\|, r_0) \tag{4.16}$$

Furthermore the global maximum in (4.16) can be found by performing a line search on $\|\boldsymbol{\tau}\|$, and the upper bound decreases as the truncation order p increases.

Proof. Taking $N = 1$, $\|\boldsymbol{\xi}_1\| = r_0$ in Theorem 3 it can be seen that (4.15) is satisfied. All that is left to be shown is that $\epsilon_p^{IFGT_3}(r_0)$ in (4.16) is a monotonically increasing function of r_0. Let a be defined as in (4.13) then using proposition 1 we have $a > r_0 > 0$. Using proposition 2, and the definition of the derivative we have

$$\lim_{\delta\to 0}\frac{1}{\delta}(\Theta_2(a, r_0+\delta) - \Theta_2(a, r_0)) > 0 \tag{4.17}$$

Multiplying both sides of (4.17) by $e^{-a^2/2}$ we get

$$\lim_{\delta\to 0}\frac{1}{\delta}(\Gamma_2(a, r_0+\delta) - \Gamma_2(a, r_0)) > 0 \tag{4.18}$$

By the definition of a and $\epsilon_p^{IFGT_3}(r_0)$ in (4.13) and (4.16) respectively, we have

$$\epsilon_p^{IFGT_3}(r_0) = \kappa\Gamma_2(a, r_0) \tag{4.19}$$

and,

$$\epsilon_p^{IFGT_3}(r_0+\delta) \geq \kappa\Gamma_2(a, r_0+\delta) \tag{4.20}$$

Using (4.19) and (4.20) in (4.18) we obtain

$$\lim_{\delta\to 0} \frac{1}{\delta}(\epsilon_p^{IFGT_3}(r_0+\delta) - \epsilon_p^{IFGT_3}(r_0)) > 0 \tag{4.21}$$

which is equivalent to $\nabla_{r_0}\epsilon_p^{IFGT_3}(r_0) > 0$. Therefore $\epsilon_p^{IFGT_3}(r_0)$ is an increasing function of r_0. □

4.2 Analysis of the new error bounds and choosing the IFGT parameters using the new bounds

In this section we first analyse the new error bounds and demonstrate that the new error bounds (a) they are significantly tighter than the old bound and (b) they are consistent with our expectation that the error decreases as the truncation order increases. Since in practice the truncation order and the number of clusters that are used to partition the source points have to be chosen experimentally such that a compromise between the speed and accuracy of the filtering algorithm is obtained, these new bounds provide important insight into this choice. We then show how the IFGT parameters r_s, r_t, and p can be chosen using the new bounds such that the KDE can be approximated to any order of accuracy.

4.2.1 Analysis of the error bounds

Analysis of the error bound in Theorem 1

We evaluate the new upper error bound in Theorem 3 and the old upper error bound (2.17), assuming that the source points are drawn from a multivariate Gaussian PDF, and the weights are obtained by evaluating the multivariate Gaussian PDF at the source points. This is an appropriate model for the filtering case where the source points and weights represent a PDF that could be modeled as a Gaussian mixture where each cluster of source points originates from a different mixture. We compare the upper error bounds to the maximum error obtained in the synthetically generated target points set when using the IFGT, where a single cluster is used for all the source points. We generated 5000 source points from a 4D multivariate Gaussian with covariance matrix $0.4I$, and set the weights of all the source

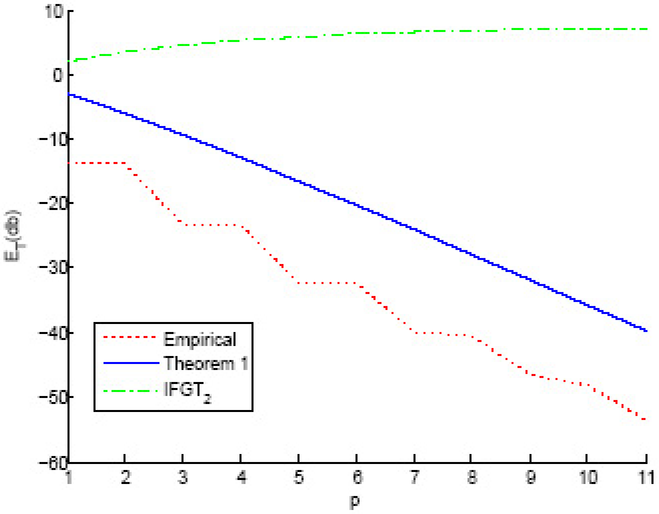

Figure 4.1: Empirical upper error bound and and upper error bounds (4.1) and (2.17) vs. truncation order for source points generated from a zero mean Gaussian with covariance matrix $0.4I$

points to the same value. For simplicity we used $\sigma = 1$ in (2.2), and we assume $\kappa = 1$. We generated 5000 target points from a uniform distribution over a hyper-cube with edges bounded by $[-5, 5]$ in each axis. Figure 4.1 compares the upper error bounds (4.1) and (2.17), and the empirical maximum error obtained for the generated data. It can be seen that the new bound is significantly tighter compared to the old bound. Since as discussed above the experimental scenario that we used to evaluate the bound given in Theorem 3 is very similar to what is expected in a nonlinear filtering problem, Figure 1 gives strong evidence that using small truncation orders should be sufficient to obtain satisfactory performance in nonlinear filtering problems.

Analysis of the error bound in Theorem 2

Figure 4.2 plots $\epsilon_p^{IFGT_3}(r_0)$ in (4.16), and $\epsilon_p^{IFGT_2}(r_0)$ in (2.15) for $0 \leq r_0 \leq 3$, and for truncation orders $p = 5, 8$. It can be seen that the new bound $\epsilon_p^{IFGT_3}(r_0)$ is significantly tighter than the existing bound $\epsilon_p^{IFGT_2}(r_0)$. Furthermore it can be seen that the old bound is not consistent with our expectation that the error decreases as the truncation order increase. Theorem 2 on the other hand stipulates that the error decreases as the truncation order increases. In the next subsection we describe our implementation of the IFGT which clusters the source points into balls with radius σr_0. Therefore based on Theorem 2 we can expect the error performance of the filtering algorithm to improve as the truncation order increases

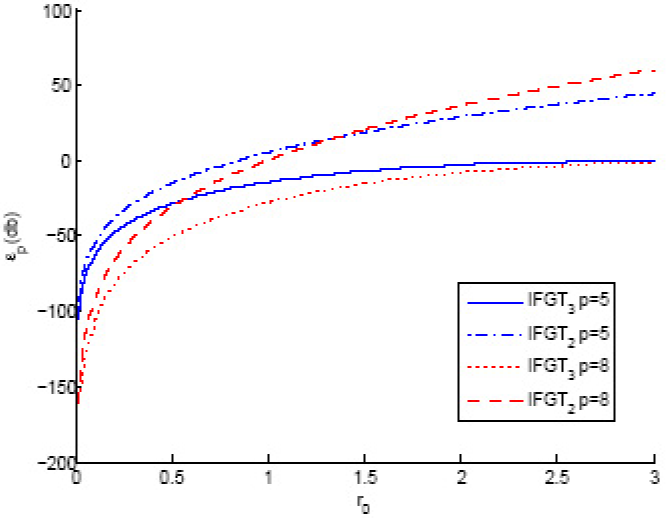

Figure 4.2: ϵ_p for the $IFGT_2$ and $IFGT_3$ vs. r_0, for $\kappa = 1$, for truncation orders 5, and 8.

for any value of r_0.

4.2.2 Choosing the IFGT parameters using the new bounds

Choosing parameters that satisfy a specified error bound

Similarly to (2.14) we take $r_t = r_s + n\sigma$, thus the error due to ignoring all the other clusters is bounded by $Qe^{-n^2/2}$. Taking $r_s = r_0\sigma$, the error due to truncating the expansion after the pth order is bounded by $Q\epsilon_p^{IFGT_3}(r_0)$. Therefore by fixing r_0 and p according to plots such as Figure 2 we can bound the maximum truncation error. Similarly we can choose n such that $Qe^{-n^2/2}$ is less than $Q\epsilon_p^{IFGT_3}(r_0)$. Therefore we define the new IFGT parameters to be r_0, n, and p. The specific details of the IFGT using this choice of parameters is summarized in Algorithm 3. We may obtain an even tighter bound than $Q\epsilon_p^{IFGT_3}(r_0)$ for the truncation error by applying the bound given in Theorem 1 for each cluster and summing the error bounds over all the clusters.

Choosing the parameters experimentally

As is evident from the previous discussion about choosing the parameters that satisfy a specified error bound, the parameters that determine the performance of the IFGT are n, r_0, and p. Since n is independent of r_0 and p it can be specified separately, whereas it can be seen from Figure 2 that r_0 and p are dependent. Furthermore the larger p is, and the smaller

r_0 is the more accurate the IFGT becomes. The process of choosing r_0 and p experimentally therefore requires incrementing p and decreasing r_0 until a satisfactory performance of the filtering algorithm is obtained. It should be noted that as p increases and r_0 decreases the computational constant of the IFGT also increases therefore the filtering algorithm becomes slower.

IFGT Algorithm:

- Cluster all the source points into balls with radius $r_0\sigma$.
- Compute the IFGT coefficients for each cluster using (2.11).
- For each target point, sum over all evaluations of (2.10) at all the clusters with the centers that lie within $(n + r_0)\sigma$ from the target point.

4.3 Nonlinear filtering using a new proposal distribution and the IFGT

4.3.1 The new filtering algorithm

AMPF-IS

We propose a new proposal distribution which we call AMPF-IS, where (2.54) is approximated using IS. For each i we draw m samples $\left\{\tilde{\boldsymbol{x}}_k^{(n)}\right\}_{n=1}^{m}$ from $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$ and approximate (2.54) using

$$\hat{p}(i|\boldsymbol{z}_{1:k}) \propto w_{k-1}^{(i)} \sum_{n=1}^{m} p(\boldsymbol{z}_k|\tilde{\boldsymbol{x}}_k^{(n)}) \tag{4.22}$$

Sampling from the proposal $\pi(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ proceeds similarly to the AMPF case using (2.55), using the approximation in (4.22) and using Algorithm 1.

Reducing the computational complexity of the AMPF using the IFGT

The AMPF and AMPF-IS improve on the SIS proposal distribution since the information provided by the latest measurement $\boldsymbol{z}_k$ is used to perform sampling in areas of high probability of the posterior $\hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$, however there is an added computational cost incurred by having to evaluate (2.47) and (2.55), both of which have $O(N^2)$ computational complexity. Next we show that this computational complexity can be reduced to $O(N)$ using the IFGT.

In this chapter we consider the case where the state noise in (2.18) is a zero mean Gaussian with a covariance matrix $\boldsymbol{\Sigma}$. Factorizing $\boldsymbol{\Sigma}$ into $\boldsymbol{\Sigma} = \boldsymbol{V}^T \boldsymbol{\Lambda} \boldsymbol{V}$ where $\boldsymbol{V}$ is an orthonormal matrix and where $\boldsymbol{\Lambda}$ is a diagonal matrix with nonnegative elements, we can rewrite (2.47) as

$$\hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) = \kappa \sum_{i=1}^{N} w_{k-1}^{(i)} e^{-\|\boldsymbol{\nu}_k - \boldsymbol{s}_{k-1}^{(i)}\|^2/2} \tag{4.23}$$

where $\boldsymbol{\nu}_k = \boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{V} \boldsymbol{x}_k$, and $\boldsymbol{s}_k^{(i)} = \Lambda^{\frac{1}{2}} \boldsymbol{V} f(\boldsymbol{x}_k^{(i)})$. Therefore (4.23) is equivalent to (2.2) where $\sigma = 1$, $q_j = w_{k-1}^{(i)}$, and $\boldsymbol{\nu}_k$ and $\boldsymbol{s}_k^{(i)}$ are the target and source points respectively. Thus (4.23) can be approximated using the IFGT. Similarly we can approximate (2.55) using the IFGT.

4.3.2 Experimental results

In the following we show experimental results for the new framework to nonlinear filtering. We also compare our results to using the SIS filter discussed previously.

Example 1

For our simulations we use the a four dimensional state space model which is an extension of the one dimensional state space model used in [8]

$$x_{t+1}^{(d)} = \frac{x_t^{(u_1(d))}}{2} + 25 \frac{x_t^{(u_2(d))}}{1 + \left(x_t^{(u_2(d))}\right)^2} + 8\cos(1.2t) + w_t^{(d)} \tag{4.24}$$

$$y_t^{(d)} = \frac{\left(x_t^{(d)}\right)^2}{20} + v_t^{(d)} \tag{4.25}$$

where $d = 1 \dots 4$, $\boldsymbol{u}_1 = [2, 4, 1, 3]^T$, $\boldsymbol{u}_2 = [3, 4, 1, 2]^T$, $w_t^{(d)} \sim N(0, 10)$, $v_t^{(d)} \sim N(0, 1)$, and $x_0^{(d)} \sim \mathcal{N}(0, 5)$. Each experiment had 200 time steps, and the results were averaged over 100 different experiments. We used $n = 4$, $r_0 = 3$, and $p = 3$ in the IFGT. In Table 4.1 we compare the root mean square error (RMSE) and timing results for the different algorithms when using the IFGT and when using direct evaluation of (2.55), (4.23). In the upper group of algorithms we compare the results when using the MPF with the AMPF proposal distribution and with the AMPF-IS proposal distribution that we developed in this work. In the bottom of the Table we show the results obtained when using the SIS particle filter. It can be seen that the new AMPF-IS proposal distribution significantly improves the RMSE over the AMPF proposal distribution, and that increasing the number of samples m used in the AMPF-IS proposal distribution, decreases the RMSE. Furthermore the use of the IFGT

Table 4.1: Time [sec], and RMSE when using the AMPF and AMPF-IS proposal distributions with direct evaluation and IFGT, and when using the SIS particle filter

Algorithm	Direct		IFGT		N	m
	RMSE	time (sec)	RMSE	time (sec)		
AMPF	6.3	46.2	6.4	3.7	1000	1
AMPF-IS	5.9	12.3	5.9	2	500	10
AMPF-IS	5.5	12.6	5.5	2.8	500	50
AMPF-IS	5.6	14.5	5.5	3.8	500	100
AMPF-IS	5.6	46.6	5.7	4.3	1000	10
AMPF-IS	5.2	49.2	5.3	6	1000	50
AMPF-IS	5.1	51.4	5.2	8.2	1000	100
SIS	6	1.9			5000	
SIS	5.6	3.8			10000	
SIS	5.5	5			15000	
SIS	5.4	8.2			20000	

instead of the direct evaluation speeds up the execution time significantly while increasing the error only slightly. The timing and RMSE results using the AMPF-IS with the IFGT are comparable to the results obtained using the SIS particle filter, however the number of samples in the SIS particle filter is significantly larger.

In Table 4.2 we show the RMSE and timing results obtained for different values of r_0 and truncation orders p for the same nonlinear filtering example using $n = 4$, $m_0 = 50$ and $N = 1000$. Each experiment had 200 time steps, and the results were averaged over 100 different experiments. It can be seen that the results verify what is predicted by the new bounds (a) the error increases as the value of r_0 increases (b) the error decreases as the value of p increases (c) satisfactory RMSE performance can be obtained for low truncation orders. Tables such as Table 4.2 can be used to find the optimal compromise between accuracy and speed of the filtering algorithm.

Example 2

In the next example we apply the new nonlinear filtering framework to the task of bearings only tracking of a single target in a sensor network [11], [16]. The state vector $\boldsymbol{x}_k$ includes the position of the target and its velocity on the $x - y$ plane, and takes the form

$$\boldsymbol{x}_{k+1} = \begin{pmatrix} I_2 & \Delta t I_2 \\ 0 & I_2 \end{pmatrix} \boldsymbol{x}_k + \begin{pmatrix} \frac{(\Delta t)^2}{2} I_2 \\ \Delta t I_2 \end{pmatrix} w_k \tag{4.26}$$

Table 4.2: Time [sec], and RMSE for different values of r_0 and p when using the AMPF-IS proposal distribution with the IFGT

$r_0 \diagdown p$	1		3		5		7	
	RMSE	time (sec)	RMSE	time (sec)	RMSE	time (sec)	RMSE	time (sec)
1	5.5	4.8	5.4	5.7	5.3	9.4	5.3	19.2
2	5.6	4.5	5.4	5.2	5.4	8	5.3	15.5
3	5.6	3.9	5.5	4.4	5.4	6.4	5.4	11.8
4	5.8	3.4	5.6	3.9	5.5	5.5	5.5	9.2
5	5.9	3.2	5.7	3.3	5.7	4.3	5.6	6.4

where Δt is the time difference between two consecutive measurements, I_2 denotes the 2×2 identity matrix, and $\boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{\Sigma})$. The measurement equation which we use follows the model used in [13]

$$z_k \sim WC\left(\arctan\left(\frac{y_k - \bar{y}}{x_k - \bar{x}}\right), \rho\right) \tag{4.27}$$

where $(\bar{x}, \bar{y})$ denotes the position of the sensor, and where $WC(\mu, \rho)$, denotes the two parameter wrapped Cauchy distribution [15], that takes the form

$$f(z_k|\mu) = \frac{1}{2\pi}\frac{1-\rho^2}{1+\rho^2 - 2\rho\cos(z_k - \mu)}, \tag{4.28}$$

where $0 \leq \rho \leq 1$, $-\pi < z_k, \mu < \pi$.

Each sensor has to choose the next leader sensor to perform the filtering once it has performed the measurement update. We use the scheme that was used in [11] where the new leader ℓ^* is chosen using the following decision rule

$$\ell^* = \arg\min_{i \in \Omega(i)} [(\bar{x}_i - x_{k+1|k})^2 + (\bar{y}_i - y_{k+1|k})^2] \tag{4.29}$$

where $\Omega(\ell)$ denotes the set of sensors that sensor ℓ can communicate with, $(\bar{x}_i, \bar{y}_i)$ are the coordinates of sensor i, and $(x_{k+1|k}, y_{k+1|k})$ is the one step ahead prediction made by the previous leader for the location of the target. Once the new leader sensor is chosen, the previous leader sensor transmits its state estimate $\hat{p}(\boldsymbol{x}_k|z_{1:k})$ to the new leader sensor.

For the experiments presented here we used $\Delta t = 1$, and $\Sigma = 0.5I_2$, where the track with 200 time steps was generated once and used for all the experiments. The measurements were generated using $\rho = 1 - \sigma^2$, where we used $\sigma = 0.02$ to generate the measurements, and $\sigma = 0.1$ to evaluate $p(z_k|x_k^{(i)})$ in (2.48) since this was necessary in order to avoid divergence

Table 4.3: Time [sec], and RMSE when using the AMPF and AMPF-IS proposal distributions with direct evaluation and IFGT, and when using the SIS particle filter

Algorithm	Direct					IFGT					N	m
	RMSE				Time	RMSE				Time		
	x_k	y_k	$\dot{x}_k$	$\dot{y}_k$		x_k	y_k	$\dot{x}_k$	$\dot{y}_k$			
AMPF	12.01	11.65	1.16	1.27	14.45	19.94	12.36	1.28	1.32	2.91	600	1
AMPF-IS	3.09	3.37	0.93	1	14.68	2.73	3.3	0.88	0.98	3.63	600	20
AMPF-IS	2.27	2.92	0.88	0.97	15.23	2.67	3.38	0.89	0.99	4.43	600	50
AMPF	3.33	4.4	0.95	1.06	39.34	8.74	8.47	1.09	1.17	5.26	1000	1
AMPF-IS	2.25	2.84	0.88	0.96	39.95	2.08	2.83	0.86	0.95	6.73	1000	20
AMPF-IS	2.14	2.82	0.87	0.95	40.81	2.1	2.81	0.87	0.95	8.35	1000	50
SIS	11.21	10.95	1.18	1.22	0.23						500	
SIS	3.56	3.35	0.93	0.99	0.35						1000	
SIS	2.15	2.78	0.86	0.96	0.55						2000	
SIS	2.07	2.62	0.85	0.93	1.08						4000	

from the real track for all the tested algorithms. The initial state estimate that we used was distributed as $\mathcal{N}([x_0; y_0]^T, 100I_2)$. For approximating the AMPF-IS integral using (4.22) we use $\sigma = 0.02$. We used $n = 4$, $r_0 = 3$, and $p = 5$ in the IFGT. The timing and RMSE results were averaged over 100 different runs. The sensor network included 200 sensors. Figure 3 shows the sensors, the target's track, and its estimate for a single experiment using the AMPF-IS and the IFGT algorithm.

Table 4.3 shows the RMSE and timing results for the tracking experiments. It can be seen that using the AMPF-IS and the IFGT we can obtain similar RMSE results to the SIS particle filter using fewer particles, however the SIS filter is faster. It should be noted that since in such a sensor network application the particles have to be transmitted between the different sensors, energy conservation requirements mandate using as few samples as possible. Thus using fewer samples may be more important than the computation time. It can also be seen that the IFGT improves the computation time over the direct evaluation, and the use of the AMPF-IS proposal distribution significantly improves the RMSE results over using the AMPF proposal distribution.

4.3.3 Choosing r_0 vs. choosing the number of clusters

As was discussed previously, the new bounds suggest that one has to choose r_0 such that the radius of the balls that are used to cluster the data is $r_0\sigma$. This is as opposed to the approach taken in [8] where one had to specify the number of clusters instead. In order to compare

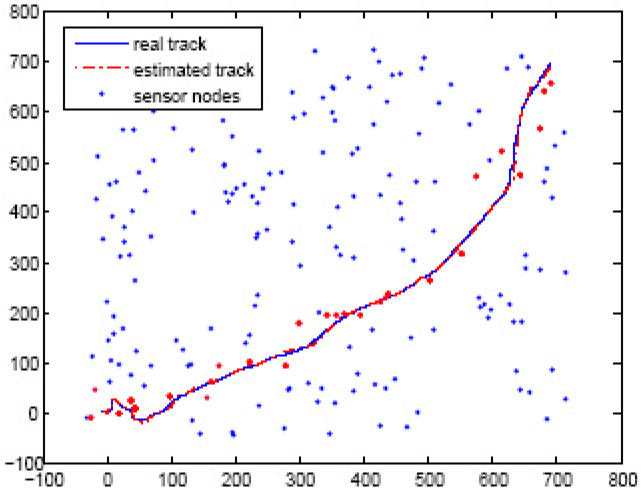

Figure 4.3: The sensor network, the real and the estimated tracks

Table 4.4: Time [sec], and RMSE when using the AMPF-IS proposal distribution and IFGT for the example in Example 1 for different values of r_0 and K

r_0	RMSE	Time (sec)	K	RMSE	Time (sec)
3	5.6	4.2	10	5.5	4.2
2	5.35	5	15	5.36	5
1	5.35	6	20	5.35	5.8

Table 4.5: Time [sec], and RMSE when using the AMPF-IS proposal distribution and IFGT for the example in Example 2 for different values of r_0 and K

r_0	RMSE				Time (sec)	K	RMSE				Time (sec)
	x_k	y_k	$\dot{x}_k$	$\dot{y}_k$			x_k	y_k	$\dot{x}_k$	$\dot{y}_k$	
3	2.26	2.84	0.87	0.96	8.5	140	2.37	2.9	0.88	0.96	11.6
2	2.28	2.81	0.86	0.95	11.4	170	3.2	4.5	0.9	1	13.5
1	2.42	2.94	0.88	0.97	14.94	200	2.35	2.86	0.88	0.96	15.38

these two approaches for the nonlinear filtering case we show the results of the two nonlinear filtering examples for different choices of r_0, and K in Tables 4.4 and 4.5 respectively. The minimal value for K was chosen such that none of the experiments diverged.

It can be seen that similar results can be obtained when fixing the right parameters for r_0 or K, however the values for r_0 remain the same for the two filtering examples, whereas the values for K change significantly between the two examples. This indicates that the formulation that is proposed here may have a significant advantage for the nonlinear filtering case as it requires less effort when choosing the IFGT parameters.

4.4 Conclusion

In this chapter we considered the use of the IFGT to perform fast KDE for evaluating the Chapman Kolmogorov equation in nonlinear filtering. We presented new upper error bounds for the IFGT which are significantly tighter than the existing bound. We related the truncation error to the distance of a source point from the center of the expansion and to the order of truncation for any target point analytically, and we showed that this error increases as the radius increases, and decreases as the truncation order increases. Since in practice the IFGT parameters have to be fixed experimentally, the new bounds facilitate the choice of parameters. We applied the IFGT to nonlinear filtering using the new AMPF-IS proposal distribution that uses IS to approximate not only the integrals in Bayes recursion equations, but also to approximate the integral in the AMPF. The experimental results using the new framework verify its effectiveness and show that it can be used to reduce the number of particles that have to be used. We also examined the effect that the IFGT parameters have on the nonlinear filtering performance, and confirmed the predictions of the the new bounds.

Chapter 5

Robust Minimax Estimation in a Linear Gaussian Model

In Chapter 3 we reviewed robust estimation in linear Gaussian models using the difference and ratio regret estimators [28, 29]. The difference and ratio regret estimators assume that the the eigenvector matrix of $\boldsymbol{C_x}$ is known and is identical to the eigenvector matrix of $\boldsymbol{H}^T\boldsymbol{C}_{\boldsymbol{w}}^{\boldsymbol{-1}}\boldsymbol{H}$, which can also be interpreted as assuming that they are diagonalized by the same unitary matrix. We therefore refer to this assumption here as the *jointly diagonalizable matrices* assumption. Furthermore the region of uncertainty is expressed using upper and lower bounds on each of the eigenvalues of $\boldsymbol{C_x}$.

In this chapter we develop a new criterion for the robust estimation problem which we call the generalized difference regret (GDR). Rather than subtracting the MSE when using the MMSE estimator matched to a covariance matrix $\boldsymbol{C_x}$ from the MSE when using an estimator $\hat{\boldsymbol{x}} = \boldsymbol{Gy}$, for the GDR we subtract another function of $\boldsymbol{C_x}$ and $\boldsymbol{C_w}$. More specifically, we develop a collection of qualifications that this function should satisfy, which are aimed at guaranteeing the scale independence of the obtained estimator and ensuring that the GDR criterion is not more pessimistic than the MSE criterion. Functions satisfying these criteria are termed *admissible regret functions*. While the choice of an admissible regret function is far from unique, in this paper, we make one suggestion which we use as the basis for the development of a new robust estimator.

The estimator we propose here generalizes the ideas in both [28] and [29] in a number of ways and can thus be used to address a far broader range of estimation problems. Most importantly, our approach does not require the joint diagonalizability assumption and allows for uncertainty in both the eigenvalues as well as the individual elements of $\boldsymbol{C_x}$. Our GDR-based scheme can also be computed easily using semidefinite programming. When considering only eigenvalue uncertainties and using the joint diagonalizable matrices assumption, we show that the resulting estimator is identical to the difference regret estimator. This result gives insight into why the new criterion is an effective tool for designing robust estimators, and helps to explain the experimental results. Additionally, we are able to show that our proposed admissible regret function is also a lower bound on the MSE when using the MMSE estimator matched to a covariance matrix $\boldsymbol{C_x}$, provided that a certain commutability assumption (which is a relaxed version of the joint diagonalizability assumption) holds. This gives additional motivation for our choice.

We test the GDR estimator using two examples. First we consider the same example used in [28, 29] when the covariance matrix is obtained from a stationary process and where the MSE is computed using the same samples that are used to find the robust estimator, and also use it for cases in which the joint diagonalizable matrices assumption does not hold. Subsequently we consider using the GDR estimator in an estimation problem in a sensor network, where unlike the previous example different samples are used to compute the MSE and to find the estimator. A major concern in sensor networks applications is the power loss due to the communication of messages between the sensor nodes rather than the energy lost during computation [22, 46]. We show that the GDR estimator can be used to reduce the number of samples which have to be transmitted to a centralized location in order to estimate a covariance matrix which is required in order to use the MMSE estimator. The experimental results of the new estimator show improved MSE compared to presently available methods.

5.1 Minimax estimation with joint eigenvalue and elementwise covariance uncertainties using the GDR criterion

In this section we propose a new criterion for the minimax problem which we call the generalized difference regret (GDR) criterion, and subsequently we use this criterion to develop a new robust estimator which has two major differences compared to the difference and ratio regret estimators. It does not necessitate the joint diagonalizable matrices assumption, and the region of uncertainty can be defined as the intersection of the eigenvalue and elementwise uncertainty regions.

As was demonstrated in [28], the MSE is a very conservative criterion for the minimax estimation problem and performs poorly, therefore the difference regret criterion was motivated as being less pessimistic than the MSE criterion. We define the GDR as the difference between the MSE when using an estimator $\boldsymbol{G}$ and a function $f(\boldsymbol{C_x}, \boldsymbol{C_w})$

$$\Re_g(\boldsymbol{C_x}, \boldsymbol{G}) = Tr(\boldsymbol{G}\boldsymbol{C_w}\boldsymbol{G}^T) + Tr(\boldsymbol{C_x}(\boldsymbol{I} - \boldsymbol{G}\boldsymbol{H})^T(\boldsymbol{I} - \boldsymbol{G}\boldsymbol{H})) - f(\boldsymbol{C_x}, \boldsymbol{C_w}). \tag{5.1}$$

It can be seen that if we take $f(\boldsymbol{C_x}, \boldsymbol{C_w})$ equal to the MSE when using the MMSE estimator matched to a covariance matrix $\boldsymbol{C_x}$ (3.26), then we obtain the difference regret as a special case of the GDR criterion. More generally we consider functions $f(\boldsymbol{C_x}, \boldsymbol{C_w})$ that satisfy the qualifications given in the following:

Definition 3. *A function $f(\boldsymbol{C_x}, \boldsymbol{C_w})$ is called an admissible regret function if it satisfies the following:*

1. $f(\boldsymbol{C_x}, \boldsymbol{C_w}) \geq 0$

2. $f(\alpha\boldsymbol{C_x}, \alpha\boldsymbol{C_w}) = \alpha f(\boldsymbol{C_x}, \boldsymbol{C_w}),\ \forall\ \alpha > 0.$

The first qualification ensures that the GDR in (5.1) is not greater than the MSE when using an estimator $\boldsymbol{G}$ as in (3.25) since the GDR criterion in (5.1) is defined as the difference between the MSE when using an estimator $\boldsymbol{G}$ and the function $f(\boldsymbol{C_x}, \boldsymbol{C_w})$, and therefore it is not more pessimistic than the MSE criterion. Using the second qualification we have that the GDR criterion satisfies

$$\Re_g(\alpha\boldsymbol{C_x}, \alpha\boldsymbol{C_w}) = \alpha\Re_g(\boldsymbol{C_x}, \boldsymbol{C_w}),\ \forall\ \alpha > 0, \tag{5.2}$$

and therefore the second qualification ensures that the obtained estimator is invariant to the scaling of $\boldsymbol{C_x}$ and $\boldsymbol{C_w}$.

In Lemma 4 we give the admissible regret function which we use to develop the new GDR estimator in Theorem 3. The admissible regret function that we propose is convex in $\boldsymbol{C_x}$ and leads to a GDR criterion which is a convex-concave function (i.e. convex in $\boldsymbol{G}$ and concave in $\boldsymbol{C_x}$), and simplifies the solution of the minimax problem significantly using the results of Lemma 3. We give more motivation for this choice of an admissible regret function in the next section.

Lemma 4. *Let $\boldsymbol{C_x} = \boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T$ where $\boldsymbol{\Delta}$ is a diagonal matrix with the nonnegative elements $\{\delta_i\}_{i=1}^m$ and where $\boldsymbol{V}$ is a unitary matrix and where $\ell_i \leq \delta_i \leq u_i$, and let $\boldsymbol{H}^T\boldsymbol{C_w^{-1}}\boldsymbol{H} = \boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^T$*

where $\boldsymbol{\Lambda}$ *is a diagonal matrix with the nonnegative elements* $\{\lambda_i\}_{i=1}^m$ *and where* $\boldsymbol{T}$ *is a unitary matrix. Let*

$$f(\boldsymbol{\delta}, \boldsymbol{C_w}) = Tr(\boldsymbol{P}^2(\boldsymbol{\Lambda}^{1/2}\boldsymbol{T}^T\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T\boldsymbol{T}\boldsymbol{\Lambda}^{1/2} + \boldsymbol{I})^{-1}), \tag{5.3}$$

where $\boldsymbol{P}$ *is a diagonal matrix with the diagonal elements* $\sqrt{\ell_i} + \kappa_i(\delta_i - \ell_i)$*, and where* $\kappa_i = (\sqrt{u_i} + \sqrt{\ell_i})^{-1}$*. We then have that* $f(\boldsymbol{\delta}, \boldsymbol{C_w})$ *is an admissible regret function and convex in* $\boldsymbol{\delta}$*.*

Proof. The nonnegativity of $f(\boldsymbol{\delta}, \boldsymbol{C_w})$ follows since $\boldsymbol{P}(\boldsymbol{\Lambda}^{1/2}\boldsymbol{T}^T\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T\boldsymbol{T}\boldsymbol{\Lambda}^{1/2} + \boldsymbol{I})^{-1}\boldsymbol{P}$ is a positive semidefinite matrix. To prove the second qualification of Definition 3 we note that $\boldsymbol{\Lambda}^{1/2}\boldsymbol{T}^T\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T\boldsymbol{T}\boldsymbol{\Lambda}^{1/2}$ is invariant to the scaling of $\boldsymbol{\delta}$ and $\boldsymbol{C_w}$, and that the scaling of $\boldsymbol{\ell}$ and $\boldsymbol{u}$ is the same as that of $\boldsymbol{\delta}$. Therefore we have

$$f(\alpha\boldsymbol{\delta}, \alpha\boldsymbol{C_w}) = Tr((\boldsymbol{P}_\alpha)^2(\boldsymbol{\Lambda}^{1/2}\boldsymbol{T}^T\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T\boldsymbol{T}\boldsymbol{\Lambda}^{1/2} + \boldsymbol{I})^{-1}) = \alpha f(\boldsymbol{\delta}, \boldsymbol{C_w}), \tag{5.4}$$

where the $\boldsymbol{P}_\alpha$ is a diagonal matrix with the diagonal elements $\sqrt{\alpha\ell_i} + (\sqrt{\alpha\ell_i} + \sqrt{\alpha u_i})^{-1}(\alpha\delta_i - \alpha\ell_i)$. In order to show that $f(\boldsymbol{\delta}, \boldsymbol{C_w})$ is convex in $\boldsymbol{\delta}$, we will shown that its epigraph is a convex set (see equivalent conditions for convexity in Chapter 3). Using Lemma 1 the epigraph of $f(\boldsymbol{\delta}, \boldsymbol{C_w})$ is the set

$$\left\{\tau \middle| \begin{bmatrix} \boldsymbol{X} & \boldsymbol{P} \\ \boldsymbol{P} & \boldsymbol{\Lambda}^{1/2}\boldsymbol{T}^T\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T\boldsymbol{T}\boldsymbol{\Lambda}^{1/2} + \boldsymbol{I} \end{bmatrix} \succeq 0,\ \tau = Tr(\boldsymbol{X})\right\}, \tag{5.5}$$

which is the projection on one of the dimensions (τ) of a convex set, where the set is convex since it is the intersection of two other convex sets, one is a proper cone and the other is an affine hyperplane. □

Next we derive in theorem 3 the new minimax estimator that uses the GDR criterion.

Theorem 3. *Let* $\boldsymbol{x}$ *denote the unknown parameter vector in the linear Gaussian model* $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w}$ *where* $\boldsymbol{H} \in \mathbb{R}^{n\times m}$ *and where* $\boldsymbol{x} \in \mathbb{R}^m$ *and* $\boldsymbol{w} \in \mathbb{R}^n$ *are independent zero mean Gaussian random vectors with covariance matrices* $\boldsymbol{C_x}$ *and* $\boldsymbol{C_w}$ *respectively. Let* $\boldsymbol{U}$ *and* $\boldsymbol{L}$ *denote elementwise upper and lower bounds on the elements of* $\boldsymbol{C_x}$ *such that* $\boldsymbol{L} \leq \boldsymbol{C_x} \leq \boldsymbol{U}$*, and let* $\boldsymbol{V}$ *denote a unitary matrix such that* $\boldsymbol{C_x} = \boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T$ *where* $\boldsymbol{\Delta}$ *is a diagonal matrix with the diagonal elements* δ_i *such that* $0 \leq \ell_i \leq \delta_i \leq u_i$*,* $i = 1, \ldots, m$*. Furthermore, let* $\boldsymbol{H}^T\boldsymbol{C}_{\boldsymbol{w}}^{\boldsymbol{-1}}\boldsymbol{H} = \boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^T$ *where* $\boldsymbol{\Lambda}$ *is a diagonal matrix with the diagonal elements* $\lambda_i \geq 0$*,*

$i = 1, \ldots, m$ and where $\boldsymbol{T}$ is a unitary matrix. Then the solution to the problem

$$\min_{\hat{\boldsymbol{x}}=\boldsymbol{G}\boldsymbol{y}} \max_{\boldsymbol{\delta}\in\Omega} \Re_g(\boldsymbol{\Delta}, \boldsymbol{G}), \tag{5.6}$$

where

$$\begin{aligned} \Re_g(\boldsymbol{\Delta}, \boldsymbol{G}) &= Tr(\boldsymbol{G}\boldsymbol{C_w}\boldsymbol{G}^T) + Tr(\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T(\boldsymbol{I}-\boldsymbol{G}\boldsymbol{H})^T(\boldsymbol{I}-\boldsymbol{G}\boldsymbol{H})) \\ &- Tr(\boldsymbol{P}^2(\boldsymbol{\Lambda}^{1/2}\boldsymbol{T}^T\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T\boldsymbol{T}\boldsymbol{\Lambda}^{1/2}+\boldsymbol{I})^{-1}), \end{aligned} \tag{5.7}$$

and where $\Omega = \{\boldsymbol{\delta} | \boldsymbol{\Delta} = diag(\boldsymbol{\delta}), \boldsymbol{L} \leq \boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T \leq \boldsymbol{U}, \ell_i \leq \delta_i \leq u_i\}$, takes the form

$$\boldsymbol{G} = \boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T\boldsymbol{H}^T + \boldsymbol{C_w})^{-1} \tag{5.8}$$

where the diagonal elements of $\boldsymbol{\Delta}$ can be obtained as follows

1. *$\boldsymbol{\delta}$ can be obtained by solving the semidefinite program*

$$\min_{\boldsymbol{Z}_1,\boldsymbol{Z}_2,\boldsymbol{\delta}} Tr(\boldsymbol{Z}_1 + \boldsymbol{Z}_2 - \boldsymbol{\Delta}) \tag{5.9}$$

subject to

$$\begin{gathered} \begin{bmatrix} \boldsymbol{Z}_1 & \boldsymbol{\Delta}\boldsymbol{V}^T\boldsymbol{H}^T \\ \boldsymbol{H}\boldsymbol{V}\boldsymbol{\Delta} & \boldsymbol{C_w} + \boldsymbol{H}\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T\boldsymbol{H}^T \end{bmatrix} \succeq 0 \\ \begin{bmatrix} \boldsymbol{Z}_2 & \boldsymbol{P} \\ \boldsymbol{P} & \boldsymbol{\Lambda}^{1/2}\boldsymbol{T}^T\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T\boldsymbol{T}\boldsymbol{\Lambda}^{1/2} + \boldsymbol{I} \end{bmatrix} \succeq 0 \\ \ell_i \leq \delta_i \leq u_i \\ \boldsymbol{L} \leq \boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T \leq \boldsymbol{U} \end{gathered} \tag{5.10}$$

where $\boldsymbol{P}$ is defined as in Lemma 4.

2. *If $\boldsymbol{V} = \boldsymbol{T}$, then $\boldsymbol{\delta}$ can be obtained by solving the semidefinite program*

$$\min_{\boldsymbol{z}_1,\boldsymbol{z}_2,\boldsymbol{\delta}} \sum_{i=1}^{m} (z_1^{(i)}/\lambda_i + z_2^{(i)}) \tag{5.11}$$

subject to

$$\begin{aligned} \begin{bmatrix} z_1^{(i)} & 1 \\ 1 & 1+\lambda_i\delta_i \end{bmatrix} &\succeq 0 \\ \begin{bmatrix} z_2^{(i)} & \sqrt{\ell_i}+\kappa_i(\delta_i-\ell_i) \\ \sqrt{\ell_i}+\kappa_i(\delta_i-\ell_i) & 1+\lambda_i\delta_i \end{bmatrix} &\succeq 0 \\ \ell_i \leq \delta_i &\leq u_i \\ \boldsymbol{L} \leq \boldsymbol{V\Delta V}^T &\leq \boldsymbol{U} \end{aligned} \tag{5.12}$$

where $\kappa_i = (\sqrt{u_i}+\sqrt{\ell_i})^{-1}$.

Proof. In order to show that the estimator takes the form in (5.8) we note that $\Re_g(\boldsymbol{\Delta}, \boldsymbol{G})$ in (5.7) and the minimax problem (5.6) satisfy all the conditions of Lemma 3 and therefore the order of minimization and maximization can be interchanged. Minimizing (5.7) with respect to $\boldsymbol{G}$ leads to a solution in the form of the MMSE estimator with a covariance matrix given by $\boldsymbol{C_x} = \boldsymbol{V\Delta V}^T$ as is given in (5.8). Substituting (5.8) into (5.7) then leads to the objective for the maximization part, which is simply the difference between the MSE when using the MMSE estimator (3.26) with $\boldsymbol{C_x} = \boldsymbol{V\Delta V}^T$ and $f(\boldsymbol{\delta})$ in (5.3),

$$\max_{\boldsymbol{\delta}\in\Omega} \{Tr((\boldsymbol{H}^T\boldsymbol{C_w^{-1}H} + \boldsymbol{V\Delta}^{-1}\boldsymbol{V}^T)^{-1}) - Tr(\boldsymbol{P}^2(\boldsymbol{\Lambda}^{1/2}\boldsymbol{T}^T\boldsymbol{V\Delta V}^T\boldsymbol{T\Lambda}^{1/2}+\boldsymbol{I})^{-1})\} \tag{5.13}$$

Additionally we have $Tr((\boldsymbol{H}^T\boldsymbol{C_w^{-1}H} + \boldsymbol{V\Delta}^{-1}\boldsymbol{V}^T)^{-1}) = Tr(((\boldsymbol{HV})^T\boldsymbol{C_w^{-1}HV} + \boldsymbol{\Delta}^{-1})^{-1})$ and using the matrix inversion Lemma [32] we have

$$((\boldsymbol{HV})^T\boldsymbol{C_w^{-1}HV} + \boldsymbol{\Delta}^{-1})^{-1} = \boldsymbol{\Delta} - \boldsymbol{\Delta V}^T\boldsymbol{H}^T(\boldsymbol{C_w} + \boldsymbol{HV\Delta V}^T\boldsymbol{H}^T)^{-1}\boldsymbol{HV\Delta}. \tag{5.14}$$

We can now rewrite (5.13) as

$$\min_{\boldsymbol{Z}_1,\boldsymbol{Z}_2,\boldsymbol{\delta}} Tr(\boldsymbol{Z}_1 + \boldsymbol{Z}_2 - \boldsymbol{\Delta}) \tag{5.15}$$

subject to

$$\begin{aligned} \boldsymbol{\Delta V}^T\boldsymbol{H}^T(\boldsymbol{C_w} + \boldsymbol{HV\Delta V}^T\boldsymbol{H}^T)^{-1}\boldsymbol{HV\Delta} &\preceq \boldsymbol{Z}_1 \\ \boldsymbol{P}(\boldsymbol{\Lambda}^{1/2}\boldsymbol{T}^T\boldsymbol{V\Delta V}^T\boldsymbol{T\Lambda}^{1/2}+\boldsymbol{I})^{-1}\boldsymbol{P} &\preceq \boldsymbol{Z}_2 \\ \ell_i \leq \delta_i &\leq u_i \\ \boldsymbol{L} \leq \boldsymbol{V\Delta V}^T &\leq \boldsymbol{U} \end{aligned} \tag{5.16}$$

and using Lemma 1 we obtain the semidefinite program in (5.9), (5.10), which proves 1.

In order to prove 2 we use $\boldsymbol{V} = \boldsymbol{T}$ in (5.13) which simplifies to

$$\max_{\boldsymbol{\delta}\in\Omega} \sum_{i=1}^{m} \left(\frac{1}{\lambda_i} - \frac{1/\lambda_i}{1+\lambda_i\delta_i} - \frac{(\sqrt{\ell_i} + \kappa_i(\delta_i - \ell_i))^2}{1+\lambda_i\delta_i} \right) \tag{5.17}$$

By adding the inequalities:

$$\begin{gathered} \frac{1}{1+\lambda_i\delta_i} \leq z_1^{(i)} \\ \frac{(\sqrt{\ell_i} + \kappa_i(\delta_i - \ell_i))^2}{1+\lambda_i\delta_i} \leq z_2^{(i)} \end{gathered}$$

and using Lemma 1, it follows that the δ_i's are obtained using the semidefinite program given by (5.11), (5.12). □

The computational complexity of the semidefinite program in 1 is $O(m^4)$ whereas the computational complexity of the semidefinite program in 2 is $O(m^3)$ [33]. Therefore if joint diagonalizability holds it can be used to reduce the computational complexity. Furthermore the semidefinite program can be solved efficiently and accurately using standard toolboxes e.g. [55].

5.2 Relationship of the GDR to the difference regret estimator

In this section we study some properties of the GDR estimator which give some motivation for our choice of the admissible regret function that we use in the GDR criterion. We first show that when assuming joint diagonalizability and when considering only eigenvalue uncertainties then the GDR estimator that we obtain is identical to the difference regret estimator. Subsequently we show that the admissible regret function that we proposed in Section 5.1 is also a lower bound on the difference regret when using a commutability assumption which is a relaxed version of the joint diagonalizability assumption.

5.2.1 Equivalence of the GDR estimator with eigenvalue alone uncertainties to the difference regret estimator

Although a closed form solution of the difference regret estimator assuming that $\boldsymbol{H}^T\boldsymbol{C_w}^{-1}\boldsymbol{H} = \boldsymbol{V\Lambda V}^T$ and with eigenvalue alone uncertainty region was presented in [28], it is interesting to derive the closed form solution to the GDR estimator under the same assumptions since it provides us with important insight concerning our approach. In order to derive the closed form solution we maximize the objective in (5.17) with respect to $\boldsymbol{\delta}$ over the uncertainty set $\Omega = \{\boldsymbol{\delta} | \ell_i \leq \delta_i \leq u_i\}$. Since the objective is concave, if its maximum is obtained inside the uncertainty interval then it is also the solution to the constrained problem. Solving for the maximum of the unconstrained problem we have that the solution must satisfy the quadratic equation

$$\delta_i^2 + \frac{2}{\lambda_i}\delta_i + \frac{(\sqrt{\ell_i} - \kappa_i \ell_i)(2\kappa_i - \lambda_i\sqrt{\ell_i} + \lambda_i \ell_i \kappa_i) - 1}{\kappa_i^2 \lambda_i} = 0, \tag{5.18}$$

and its solution takes the form

$$\begin{aligned}\delta_i &= -\frac{1}{\lambda_i} + \frac{1}{\lambda_i}\sqrt{1 - \frac{\lambda_i}{\kappa_i^2}(\sqrt{\ell_i} - \kappa_i\ell_i)(2\kappa_i - \lambda_i\sqrt{\ell_i} + \lambda_i\ell_i\kappa_i) + \frac{\lambda_i}{\kappa_i^2}} \\ &= \frac{1}{\lambda_i}(\sqrt{1 + \lambda_i u_i + \lambda_i \ell_i + \lambda_i^2 \ell_i u_i} - 1). \qquad (5.19)\end{aligned}$$

It is straightforward to verify that (5.19) satisfies $\ell_i \leq \delta_i \leq u_i$ and therefore it is also the solution to the constrained problem. Furthermore, if we define $\zeta_i = (u_i + \ell_i)/2$ and $\epsilon_i = (u_i - \ell_i)/2$ then we obtain that

$$\delta_i = \frac{1}{\lambda_i}\Big(\sqrt{(1 + \lambda_i\zeta_i)^2 - \lambda_i^2\epsilon_i^2} - 1\Big), \tag{5.20}$$

which is identical to the solution that is obtained for the difference regret estimator (3.30).

This result indicates that if the elementwise bounds are very loose (as may be the case in high SNR scenarios), and if the joint diagonalizable matrices assumption holds then the performance is going to be identical to that of the difference regret estimator. It also gives us insight into why the GDR criterion performs well experimentally, since it leads to the same solution as the difference regret criterion under the same assumptions in this case.

5.2.2 The GDR as a lower bound on the MSE obtained using the MMSE estimator

In this subsection we show that using a commutability assumption, which can be interpreted as a relaxed version of the joint diagonalizability assumption, the admissible regret function that we use in the admissible regret function that we use can also be obtained as a lower bound on the MSE when using the MMSE estimator matched to a covariance matrix $\boldsymbol{C}_{\boldsymbol{x}}$.

Let $\boldsymbol{Q} = \boldsymbol{\Delta}^{1/2}\boldsymbol{V}^T\boldsymbol{T}\boldsymbol{\Lambda}^{1/2}$ and assume that $\boldsymbol{Q}$ and $\boldsymbol{Q}^T$ are commutable i.e. they satisfy $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{Q}^T\boldsymbol{Q}$ (this clearly holds if $\boldsymbol{V} = \boldsymbol{T}$ but also for example if $\boldsymbol{\Delta} = \boldsymbol{\Lambda}$ and $\boldsymbol{V}^T\boldsymbol{T} = \boldsymbol{T}^T\boldsymbol{V}$). Using $\boldsymbol{C_x} = \boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^T$ and $\boldsymbol{H}^T\boldsymbol{C}_{\boldsymbol{w}}^{\boldsymbol{-1}}\boldsymbol{H} = \boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^T$ in (3.26) we have

$$\begin{aligned} Tr((\boldsymbol{H}^T\boldsymbol{C}_{\boldsymbol{w}}^{\boldsymbol{-1}}\boldsymbol{H} + \boldsymbol{C}_{\boldsymbol{x}}^{\boldsymbol{-1}})^{-1}) &= Tr((\boldsymbol{V}^T\boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}^T\boldsymbol{V} + \boldsymbol{\Delta}^{-1})^{-1}) = Tr(\boldsymbol{\Delta}(\boldsymbol{Q}\boldsymbol{Q}^T + \boldsymbol{I})^{-1}) \\ &= Tr(\boldsymbol{\Delta}(\boldsymbol{Q}^T\boldsymbol{Q} + \boldsymbol{I})^{-1}) \geq Tr(\boldsymbol{P}^2(\boldsymbol{Q}^T\boldsymbol{Q} + \boldsymbol{I})^{-1}) \qquad (5.21) \end{aligned}$$

where the equality in (5.21) follows from the commutability assumption. Since the lower bound in (5.21) is equal to (5.3), in order to prove our claim we will show that

$$Tr((\boldsymbol{\Delta} - \boldsymbol{P}^2)(\boldsymbol{Q}^T\boldsymbol{Q} + \boldsymbol{I})^{-1}) \geq 0, \ \forall\ \ell_i \leq \delta_i \leq u_i. \qquad (5.22)$$

Assume that $\boldsymbol{\Delta} - \boldsymbol{P^2} \succeq 0$ then we can define $\boldsymbol{A} = ((\boldsymbol{\Delta} - \boldsymbol{P}^2)^{1/2}(\boldsymbol{Q}^T\boldsymbol{Q} + \boldsymbol{I})^{-1/2}$ where $(\cdot)^{1/2}$ denotes the nonnegative symmetric square root of a matrix. We can now write the left hand side of (5.22) as

$$Tr((\boldsymbol{\Delta} - \boldsymbol{P}^2)(\boldsymbol{Q}^T\boldsymbol{Q} + \boldsymbol{I})^{-1}) = Tr(\boldsymbol{A}\boldsymbol{A}^T) \geq 0. \qquad (5.23)$$

Therefore all that there is left to show is that $\boldsymbol{\Delta} - \boldsymbol{P}^2$ is indeed a positive semidefinite matrix in the region of uncertainty. Since $\boldsymbol{P}$ and $\boldsymbol{\Delta}$ are diagonal matrices this is equivalent to

$$\delta_i - (\sqrt{\ell_i} + \kappa_i(\delta_i - \ell_i))^2 \geq 0, \ \forall \ell_i \leq \delta_i \leq u_i, \qquad (5.24)$$

and since δ_i is nonnegative it suffices to show that

$$p_i(\delta_i) = \sqrt{\ell_i} + \kappa_i(\delta_i - \ell_i) \leq \sqrt{\delta_i}, \ \forall \ell_i \leq \delta_i \leq u_i. \qquad (5.25)$$

We note that $\sqrt{\delta_i}$ is concave and since $p_i(\delta_i)$ is the line segment that connects the two points ℓ_i and u_i on the graph of $\sqrt{\delta_i}$, the line segment must lie bellow the graph which proves (5.25).

Our choice of admissible regret function can therefore be interpreted as relaxing a lower bound on the MSE when using the MMSE estimator matched to covariance matrix $\boldsymbol{C_x}$, which is only valid if the commutability assumption holds. It is important to emphasize however that since the solution of the minimax problem in Section 5.1 is obtained without any of the commutability or joint diagonalizability assumptions, the GDR estimator can be used generally also when these assumptions do not hold. This is also verified by the experimental results that are given in the next section.

5.3 Example of the GDR estimator

The example that we consider here is an estimation problem with the model given in (3.22), where $\boldsymbol{x}$ is a length m segment of a zero mean stationary first order autoregressive process with parameter ρ and where the covariance matrix of $\boldsymbol{w}$ is $\boldsymbol{C_w} = \sigma^2\boldsymbol{I}$. The autocorrelation function of $\boldsymbol{x}$ therefore takes the form

$$\mathbb{E}(x_i x_j) = \rho^{|j-i|}, \tag{5.26}$$

and where $\boldsymbol{w}$ is a Gaussian random vector independent of $\boldsymbol{x}$ with the covariance matrix $\boldsymbol{C_w} = \sigma^2\boldsymbol{I}$ where σ is assumed to be known. In the following subsection we discuss the estimation of covariance matrix and the uncertainty region for the problem.

5.3.1 Estimating the uncertainty region

The covariance matrix of $\boldsymbol{x}$, which is denoted by $\boldsymbol{C_x}$, is unknown and can be estimated from the available noisy measurements vector $\boldsymbol{y}$ using the estimator

$$\hat{\boldsymbol{C}}_{\boldsymbol{x}} = [\boldsymbol{H}^{\dagger}(\hat{\boldsymbol{C}}_{\boldsymbol{y}} - \boldsymbol{C_w})\boldsymbol{H}^{\dagger T}]_{+} = [\boldsymbol{H}^{\dagger}(\hat{\boldsymbol{C}}_{\boldsymbol{y}} - \sigma^2\boldsymbol{I})\boldsymbol{H}^{\dagger T}]_{+} \tag{5.27}$$

where the estimate of the covariance matrix of y takes the form [28]

$$\hat{\boldsymbol{C}}_{\boldsymbol{y}}(i,j) = \frac{1}{n}\sum_{k=1}^{n-|j-i|} y_k y_{k+|j-i|}, \tag{5.28}$$

and where $[\boldsymbol{C}]_{+}$ is obtained by replacing all the negative eigenvalues of $\boldsymbol{C}$ with zero. Specifically let $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{Q}\boldsymbol{U}^{-\mathbf{1}}$ where $\boldsymbol{Q}$ is a diagonal matrix, then $[\boldsymbol{C}]_{+} = \boldsymbol{U}\bar{\boldsymbol{Q}}\boldsymbol{U}^{-\mathbf{1}}$ where $\bar{\boldsymbol{Q}}$ is a diagonal matrix with the elements $\bar{q}_{ii} = \max(0, q_{ii})$. Since the estimators considered in this paper assume that the eigenvector matrix of the parameter's covariance $\boldsymbol{V}$ is known, we set it equal to the eigenvector matrix of $\hat{\boldsymbol{C}}_{\boldsymbol{x}}$ (more on the estimation of the eigenvectors of covariance matrices can be found in [53]). Let ζ_i denote the eigenvalues of $\hat{\boldsymbol{C}}_{\boldsymbol{x}}$ then similarly to [28, 29] we set the upper and lower bounds for the eigenvalues of the covariance matrix as $u_i = \zeta_i + \epsilon_i$, $\ell_i = \zeta_i - \epsilon_i$, where ϵ_i is proportional to the standard deviation of an estimate $\hat{\sigma}_x^2$ of the variance σ_x^2.

If $\boldsymbol{H} = \boldsymbol{I}$ then we have

$$\hat{\sigma}_x^2 = \frac{1}{n}\sum_{i=1}^{n} y_i^2 - \sigma^2, \tag{5.29}$$

and the variance of $\hat{\sigma}_x^2$ takes the form:

$$\begin{aligned}\mathbb{E}\Big\{(\hat{\sigma}_x^2-\sigma_x^2)^2\Big\} &= \mathbb{E}\Bigg\{\bigg(\frac{1}{n}\sum_{i=1}^{n}(y_i^2-\sigma_w^2-\sigma_x^2)\bigg)^2\Bigg\}\\ &= \frac{1}{n^2}\sum_{i,j=1}^{n}\mathbb{E}(t_it_j), \qquad (5.30)\end{aligned}$$

where $t_i = y_i^2 - \sigma^2 - \sigma_x^2$. Since $\boldsymbol{x}$ and $\boldsymbol{w}$ are Gaussian and independent we have

$$\mathbb{E}\Big\{(\hat{\sigma}_x^2-\sigma_x^2)^2\Big\} = \frac{2}{n^2}\sum_{i,j=1}^{n}(\boldsymbol{C_x}(i,j)+\sigma^2\delta_{ij})^2. \qquad (5.31)$$

The expression given in (5.31) for the variance of the estimate is slightly different from that given in [28] since we did not assume that the covariance matrix is circular which leads to the simplified expression given in [28] as this is only true in the limit when $n \to \infty$ [54].

If $\boldsymbol{H} \neq \boldsymbol{I}$ then we have the following estimator for the variance of the signal

$$\hat{\sigma}_x^2 = \frac{1}{n}(Tr(\boldsymbol{H}^\dagger\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{H}^{\dagger T}) - Tr(\boldsymbol{H}^\dagger\boldsymbol{C_w}\boldsymbol{H}^{\dagger T})), \qquad (5.32)$$

and therefore the variance of the estimator is

$$\begin{aligned}\mathbb{E}\Big\{(\hat{\sigma}_x^2-\sigma_x^2)^2\Big\} &= \frac{1}{n^2}\mathbb{E}\Big\{\Big(Tr(\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{H}^{\dagger T}\boldsymbol{H}^\dagger) - Tr(\boldsymbol{C_x}+\boldsymbol{H}^\dagger\boldsymbol{C_w}\boldsymbol{H}^{\dagger T})\Big)^2\Big\}\\ &= \frac{1}{n^2}\mathbb{E}\Big\{(Tr(\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{H}^{\dagger T}\boldsymbol{H}^\dagger))^2\Big\} - \frac{1}{n^2}(Tr(\boldsymbol{C_x}+\boldsymbol{H}^\dagger\boldsymbol{C_w}\boldsymbol{H}^{\dagger T}))^2. \qquad (5.33)\end{aligned}$$

Denoting $\mathscr{H} = \boldsymbol{H}^{\dagger T}\boldsymbol{H}^\dagger$ we have

$$\mathbb{E}\{(Tr(\boldsymbol{y}\boldsymbol{y}^T\mathscr{H}))^2\} = \sum_{i,j,k,\ell=1}^{n}\mathbb{E}\{y_iy_jy_ky_\ell\}\mathscr{H}_{i,j}\mathscr{H}_{k,\ell} = \sum_{i,j=1}^{n}\mathscr{H}_{i,j}Tr(\mathbb{E}\{y_iy_j\boldsymbol{y}\boldsymbol{y}^T\}\mathscr{H}). \qquad (5.34)$$

We use the following result from [56] that if $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$ then

$$\begin{aligned}E((\boldsymbol{y}^T\boldsymbol{A}\boldsymbol{y})\boldsymbol{y}\boldsymbol{y}^T) &= (\boldsymbol{\Sigma}+\boldsymbol{m}\boldsymbol{m}^T)(\boldsymbol{A}+\boldsymbol{A}^T)(\boldsymbol{\Sigma}+\boldsymbol{m}\boldsymbol{m}^T)\\ &+ \boldsymbol{m}^T\boldsymbol{A}\boldsymbol{m}(\boldsymbol{\Sigma}-\boldsymbol{m}\boldsymbol{m}^T) + Tr(\boldsymbol{A}\boldsymbol{\Sigma})(\boldsymbol{\Sigma}+\boldsymbol{m}\boldsymbol{m}^T). \qquad (5.35)\end{aligned}$$

Since $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{H}\boldsymbol{C_x}\boldsymbol{H}^T + \boldsymbol{C_w})$ we can use in (5.35) $\boldsymbol{m} = \boldsymbol{0}$, $\boldsymbol{\Sigma} = \boldsymbol{H}\boldsymbol{C_x}\boldsymbol{H}^T + \boldsymbol{C_w}$, and $\boldsymbol{A} = \boldsymbol{E}_{i,j}$ where $\boldsymbol{E}_{i,j}$ is an $n \times n$ matrix with all zero entries but for the i, j entry which is

1. Therefore we have

$$\mathbb{E}\{y_i y_j \boldsymbol{y}\boldsymbol{y}^T\} = \boldsymbol{\Sigma}(\boldsymbol{E}_{i,j} + \boldsymbol{E}_{i,j}^T)\boldsymbol{\Sigma} + Tr(E_{i,j}\boldsymbol{\Sigma})\boldsymbol{\Sigma}. \tag{5.36}$$

Summarizing (5.33), (5.34), and (5.36) we obtain that the variance of the estimator $\hat{\sigma}_x^2$ is

$$\begin{aligned}\mathbb{E}\Big\{(\hat{\sigma}_x^2 - \sigma_x^2)^2\Big\} &= \frac{1}{n^2}\sum_{i,j=1}^{n} \mathscr{H}_{i,j} Tr((\boldsymbol{\Sigma}(\boldsymbol{E}_{i,j} + \boldsymbol{E}_{i,j}^T)\boldsymbol{\Sigma} + Tr(E_{i,j}\boldsymbol{\Sigma})\boldsymbol{\Sigma})\mathscr{H}) \\ &- \frac{1}{n^2}(Tr(\boldsymbol{C}_{\boldsymbol{x}} + \boldsymbol{H}^\dagger \boldsymbol{C}_{\boldsymbol{w}} \boldsymbol{H}^{\dagger T}))^2. \end{aligned} \tag{5.37}$$

In order to ensure the nonnegativity of the eigenvalues, ϵ_i takes the form

$$\epsilon_i = \min\left(\zeta_i, A \cdot \sqrt{\mathbb{E}\Big\{(\hat{\sigma}_x^2 - \sigma_x^2)^2\Big\}}\right), \tag{5.38}$$

where the estimate $\hat{\boldsymbol{C}}_{\boldsymbol{x}}(i,j)$ is used instead of $\boldsymbol{C}_{\boldsymbol{x}}(i,j)$ in (5.31) or (5.37) in order to compute the variance of $\hat{\sigma}_x^2$, and where A is a proportionality constant chosen experimentally.

The elementwise bounds are chosen to be proportional to $\hat{\sigma}_x^2$, and inversely proportional to the standard deviation of $\hat{\sigma}_x^2$. Choosing the elements of the covariance matrix to be proportional to the variance is very intuitive since if the variance is large then the elements of the covariance matrix are expected to be larger, and alternatively if the variance is small then the elements of the covariance matrix are expected to be smaller. The motivation for choosing the elementwise to be inversely proportionality to the standard deviation of $\hat{\sigma}_x^2$ is less intuitive though. We argue that if the standard deviation of $\hat{\sigma}_x^2$ is small then we would like the bounds to be very loose so that we only employ the eigenvalue uncertainties, and on the other hand if the standard deviation of $\hat{\sigma}_x^2$ is large then we can not obtain a good estimate of σ_x and therefore the bounds should be very small such that the estimator is close to $G = 0$. We therefore set the elementwise bounds to

$$\boldsymbol{U}(i,j) = -\boldsymbol{L}(i,j) = B\frac{\hat{\boldsymbol{C}}_{\boldsymbol{x}}(1,1)}{\sqrt{\mathbb{E}\Big\{(\hat{\sigma}_x^2 - \sigma_x^2)^2\Big\}}}, \tag{5.39}$$

where B is a proportionality constant, and the estimate $\hat{\boldsymbol{C}}_{\boldsymbol{x}}(i,j)$ is used in (5.31) or (5.37) instead of $\boldsymbol{C}_{\boldsymbol{x}}(i,j)$ in order to compute the variance of $\hat{\sigma}_x^2$.

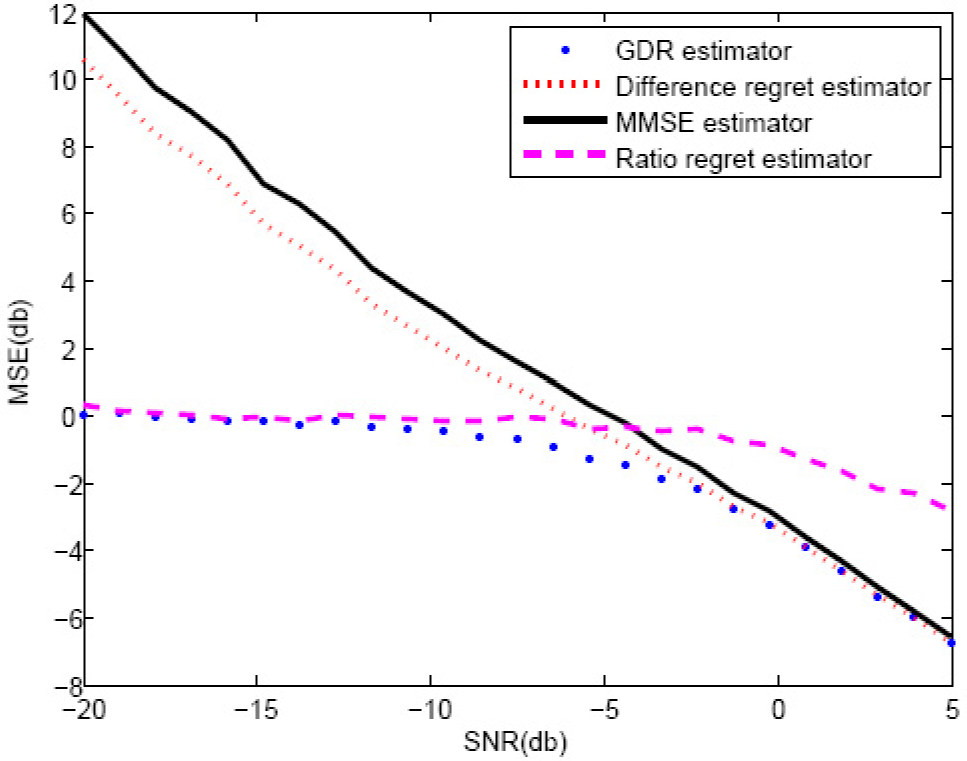

Figure 5.1: MSE vs. the SNR for the GDR estimator, difference and ratio regret estimators, and the MMSE estimator matched to the estimated covariance, for $\boldsymbol{H} = \boldsymbol{I}$

5.3.2 Experimental results

Figure 5.1 shows the MSE vs. SNR for $\boldsymbol{H} = \boldsymbol{I}$. This model satisfies the constraint $\boldsymbol{H}^T \boldsymbol{C}_{\boldsymbol{w}}^{-1} \boldsymbol{H} = \boldsymbol{V} \boldsymbol{\Lambda} \boldsymbol{V}^T$, which is required by the difference and ratio regret estimators, for any orthonormal matrix $\boldsymbol{V}$. Furthermore we can use the more computationally efficient implementation given in Theorem 3 for this case. The parameters that we used were $n = 10$, $\sigma = 1$, $A = 4$, $B = 1$, $\rho = 0.8$, and the MSE was averaged over 2000 independent experiments for each SNR value. It can be seen that the GDR estimator can improve the MSE compared to all the other estimators. Since the joint digonalizable matrices assumption holds for this example it follows from Section 5.2.1 that the results obtained using the GDR estimator with eigenvalue alone uncertainties are the same as those obtained using the difference regret estimator. This explains the convergence of the GDR estimator with the joint elementwise and eigenvalue uncertainties to the difference regret estimator in high SNRs, since the elementwise uncertainty was chosen to be very large for high SNRs. It can also be seen that the GDR estimator converges to the ratio regret estimator in low SNRs, which can be explained as an effect of the elementwise bounds. Since the elements of the covariance

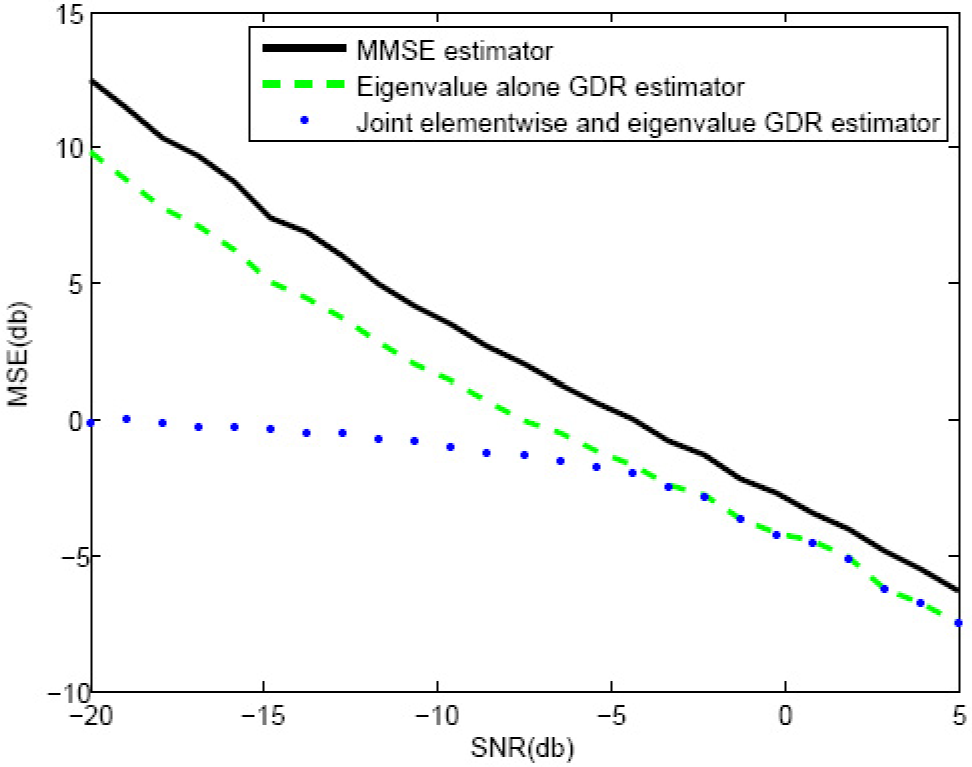

Figure 5.2: MSE vs. SNR for the GDR estimator and for the MMSE estimator matched to the estimated covariance, with $\boldsymbol{H}$ in a toeplitz form

matrix are bounded, then it can be seen from (5.8) that as the variance of the noise increases the estimator converges to $\boldsymbol{G} = 0$.

Figures 5.2 and 5.3 show the MSE vs. SNR when $\boldsymbol{H}$ is a toeplitz matrix and a diagonal matrix respectively such that the joint diagonalizable matrices assumption does not hold. Specifically in Figure 5.2 we use a toeplitz matrix which implements a linear time invariant filter with 4 taps given by $h[0] = 1$, $h[1] = 0.4$, $h[2] = 0.2$, $h[3] = 0.1$, and in Figure 5.3 we use the diagonal matrix $\boldsymbol{H} = \text{diag}([1, 0.8, 1, 0.5, 1.3, 1.2, 1.5, 0.7, 2, 1.5]^T)$ where the diagonal elements were chosen arbitrarily. In both Figures we used the parameters $n = 10$, $\sigma = 1$, $A = 4$, $B = 2$, $\rho = 0.8$, and the GDR eigenvalue alone estimator was obtained by removing the elementwise uncertainty constraint from (5.10). It can be seen from both of the Figures that the MSE can be improved significantly when using the GDR estimator compared to using the MMSE estimator.

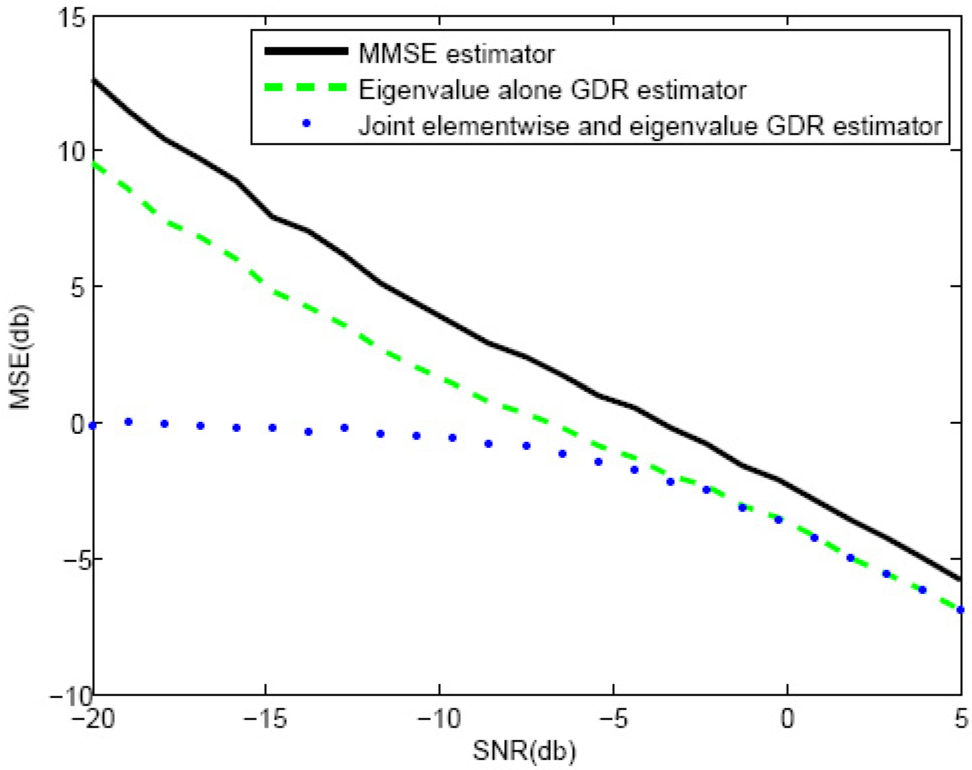

Figure 5.3: MSE vs. SNR for the GDR estimator and for the MMSE estimator matched to the estimated covariance, with $\boldsymbol{H}$ in a diagonal form

5.4 Robust estimation in a sensor network

As explained in Chapter 3 when performing estimation in a sensor network, power conservation is one of the major concerns. The more energy efficient approaches transmit messages between the sensor nodes and have the sensors perform the estimation collectively, such as the decentralized estimation algorithms presented in [37,38]. Nevertheless these distributed estimation algorithms depend on an estimate of the covariance or inverse covariance matrix, and therefore in practice require an initial stage where many samples are transmitted to a centralized location so that the covariance matrix or inverse covariance matrix matrix can be estimated. The results presented in this chapter can be used to improve the estimation performance for a given number of samples that are transmitted to the centralized location and used in order to obtain the estimator. Furthermore since in the GDR estimator has the same form as the MMSE estimator then one can use the same methods presented in [37,38] to perform distributed estimation.

The estimation model for the sensor network case is

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{w}, \tag{5.40}$$

where we assume that each node's signal is a scalar (extension to the vector case is straightforward) and the Gaussian random vector $\boldsymbol{x}$ is composed of all the sensors' signals. Similarly the vector $\boldsymbol{y}$ is composed of all the sensors' noisy observations. The Gaussian random noise vector $\boldsymbol{w}$ where the covariance matrix of $\boldsymbol{w}$ is $\boldsymbol{C_w} = \sigma^2\boldsymbol{I}$. This model is identical to (3.22) with $\boldsymbol{H} = \boldsymbol{I}$, and therefore satisfies the constraint $\boldsymbol{H}^T\boldsymbol{C_w^{-1}}\boldsymbol{H} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ which is required by the difference and ratio regret estimators for any orthonormal matrix $\boldsymbol{V}$. Unlike the previous examples, in this example we use a different set of samples for finding the estimator and for testing its performance and therefore the elementwise bounds used in the previous example do not apply in this case. However since in a sensor network the variance at each sensor can be estimated without transmitting any data (assuming that the observation noise is i.i.d.), we can assume that it is known and use the bound for the elements of the covariance matrix $\boldsymbol{C_x}$ [52]

$$|\boldsymbol{C_x}(i,j)| \leq \sigma_{x,i}\sigma_{x,j} \ \forall \ i,j \in 1,\ldots,n, \tag{5.41}$$

where $\sigma_{x,i}$ denotes the true standard deviation of sensor i, in order to obtain the required elementwise bounds.

In order to simulate the sensors' signals we assume that the covariance matrix is obtained from a Gaussian process (GP) [44,47] as such modeling is common in sensor networks e.g. [45]. We use a zero mean GP with a neural network covariance function [44] that takes the form

$$k(\boldsymbol{s},\boldsymbol{s}') = \frac{2}{\pi}\sin^{-1}\left(\frac{2\tilde{\boldsymbol{s}}^T\boldsymbol{\Sigma}\tilde{\boldsymbol{s}}'}{\sqrt{(1+2\tilde{\boldsymbol{s}}^T\boldsymbol{\Sigma}\tilde{\boldsymbol{s}})(1+2\tilde{\boldsymbol{s}}'^T\boldsymbol{\Sigma}\tilde{\boldsymbol{s}}')}}\right), \tag{5.42}$$

where $\tilde{\boldsymbol{s}} = [\ 1 \quad \boldsymbol{s}^T\ ]^T$, and we used $\boldsymbol{\Sigma} = \text{diag}([10,10,10]^T)$. We generate the positions of 20 sensors $\{\boldsymbol{s}_j, j = 1,\ldots,20\}$ by sampling a uniform distribution over $[-2,2]$ for both of the axes. The covariance matrix of the signal vector $\boldsymbol{x}$ is then obtained by $\boldsymbol{C_x}(i,j) = k(\boldsymbol{s}_i,\boldsymbol{s}_j)$, and the measurement vectors $\boldsymbol{y}^{(i)}$, $i = 1,\ldots,n$ available at the centralized location are generated using (5.40). The covariance matrix is then estimated from the available samples using

$$\hat{\boldsymbol{C}}_{\boldsymbol{x}} = \left[\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i\boldsymbol{y}_i^T - \sigma^2\boldsymbol{I}\right]_+, \tag{5.43}$$

where σ^2 denotes the variance of the noise which is assumed known, and $[\boldsymbol{C}]_+$ is obtained

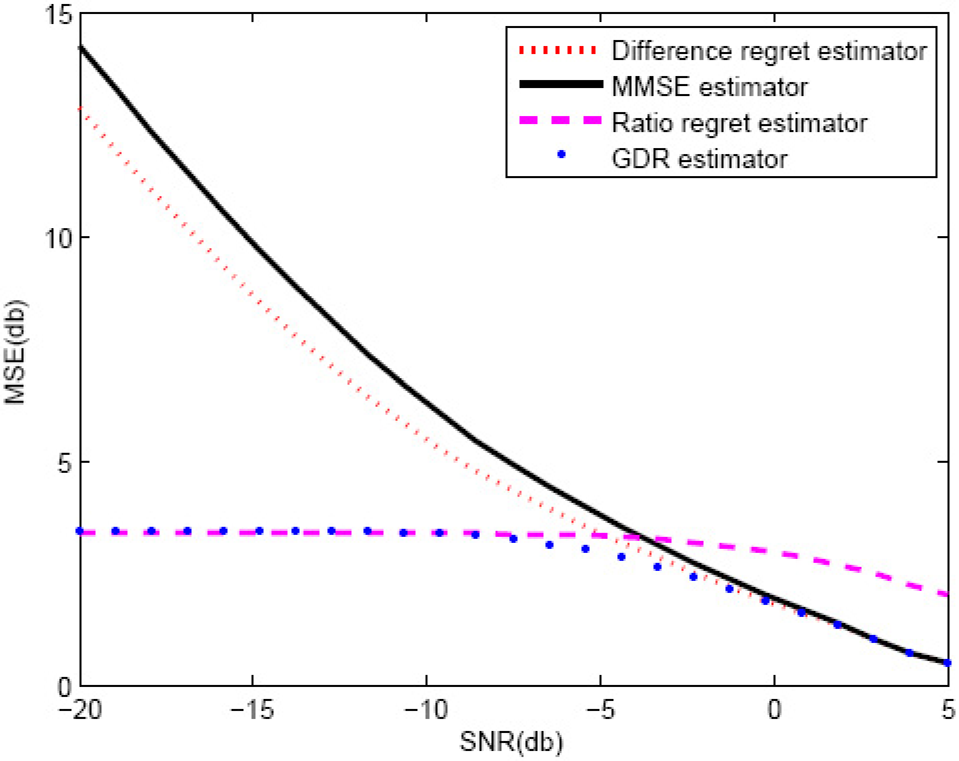

Figure 5.4: MSE vs. SNR for different estimators

by replacing the negative eigenvalues of $\boldsymbol{C}$ with zero. Let ζ_i denote the eigenvalues of $\hat{\boldsymbol{C}}_{\boldsymbol{x}}$ then we set the bounds on the eigenvalues to be $\ell_i = 0$, and $u_i = 2\zeta_i$. The bounds on the elements of the covariance matrix are set using (5.41) to $\boldsymbol{U}(i,j) = \sqrt{\boldsymbol{C}_{\boldsymbol{x}}(i,i)\boldsymbol{C}_{\boldsymbol{x}}(j,j)}$ and $\boldsymbol{L}(i,j) = -\sqrt{\boldsymbol{C}_{\boldsymbol{x}}(i,i)\boldsymbol{C}_{\boldsymbol{x}}(j,j)}$, where $\boldsymbol{C}_{\boldsymbol{x}}(i,i)$ denotes the true variance of the signal at sensor node i which as mentioned previously is assumed to be known.

In order to show the usefulness of the GDR estimator for the sensor network problem we assume that we have only $n = 5$ measurement vectors at the centralized location using which we can obtain the robust estimator for $\boldsymbol{x}$. We averaged the MSE shown in Figure 5.4 over 2000 experiments, where in each experiment we first generated $n = 5$ measurements from the linear Gaussian model which were used to obtain the robust estimator, and subsequently we computed the MSE using 2000 measurements which were different from those that were used to find the robust estimator. It can be seen that the GDR estimator either improves or performs equally as well as the other estimators. Furthermore since the joint diagonalizable matrices assumption holds for this example, for high SNRs when the elementwise bounds are very loose we have that the performance of the GDR estimator with joint elementwise and eigenvalue uncertainties converges to that of the difference regret estimator, as is shown in

Section 5.2.1. Similarly to the example in the previous section it can be seen that the GDR converges to the ratio regret estimator for low SNRs, which is the effect of the elementwise bounds on the covariance matrix.

5.5 Conclusion

We presented a new minimax estimator that is robust to an uncertainty region that is described using bounds on the eigenvalues and bounds on the elements of the covariance matrix. The estimator is based on a new criterion which is called the generalized difference regret (GDR) and can be obtained efficiently using semidefinite programming. Furthermore the GDR estimator avoids the joint diagonalizable matrices assumption that is required by both the difference and ratio regret estimators and can therefore be used in more general cases. We also showed that when the joint diagonalizable matrices assumption holds and when there are only eigenvalue uncertainties, then the GDR estimator is identical to the difference regret estimator. This result gives motivation into why the proposed criterion is successful, and explains the convergence of the GDR estimator with joint elementwise and eigenvalue uncertainties to the difference regret estimator in high SNRs when the joint diagonalizable matrices assumption holds. The experimental results show that the GDR estimator can improve the MSE over the MMSE estimator and the difference and ratio regret estimators. When considering model matrices that do not satisfy the joint diagonalizable matrices assumption we also showed significant MSE improvement compared to the MMSE estimator.

Chapter 6

Robust Parameter Estimation with Sensor Positional Uncertainty

Whereas in the previous chapter we considered robust estimation of a random parameter vector, in this chapter we turn to estimating a deterministic parameter. There has been much recent work on robust estimation of a deterministic parameter in the linear Gaussian model e.g. [26], [27], however in many cases the observation model represents some physical system which is not linear. In this chapter we consider robust parameter estimation with sensor positional uncertainty, where the measurements are obtained from electromagnetic induction (EMI) sensors. These parameters are an integral part of the buried unexploded ordnance (UXO) classification schemes used in [57], [58], and therefore as was reported in [57], obtaining better parameter estimates by addressing the sensor positional uncertainties is necessary in order to improve the classifier's performance.

The approach that we present in this chapter for the parameter estimation with sensor positional uncertainty is an extension of the method presented in [57] where the parameter estimation is formulated as a minimax problem. Specifically, using the robust optimization tools discussed in Chapter 3 we propose a nonlinear programming (NLP) approach instead of the dynamic programming (DP) approach that was presented in [57]. The DP approach assumes a discretization of the uncertainty set, and can be shown to converge to the optimal solution if the discretization is fine enough. Nevertheless, as the discretization becomes finer the computational complexity increases significantly and practical considerations necessitate using an approximate DP approach. The NLP approach avoids the discretization stage which is necessary in the DP approach to solving the minimax problem, and therefore has the potential to improve the performance in practice.

In the next section we describe the measurement model which was used in [57,58], and is also known as the *dipole model.* We then review the robust parameter estimation approach that was used in [57], and finally present the new nonlinear programming approach to the problem. Future work will include simulation and validation of the new approach.

6.1 The measurement model and nonlinear least squares parameter estimation

The dipole model that was given in [57], [58] is a truncated version of the EMI physical model in [59], [60]. According to this model the measurements collected at M frequency or time steps, and at N locations with nominal coordinates $\bar{\boldsymbol{r}}_i = (x_i, y_i, z_i)$, take the form

$$d_{i,j} = \boldsymbol{g}_i(\boldsymbol{r}_0)^T \boldsymbol{R}(\boldsymbol{\alpha}) \boldsymbol{\Lambda}_j \boldsymbol{R} \boldsymbol{f}_i(\boldsymbol{r}_0) + n_{i,j} = s_{i,j} + n_{i,j},\ i = 1, \ldots, N,\ j = 1, \ldots, M \tag{6.1}$$

where $\boldsymbol{g}_i(r_0)$ and $\boldsymbol{f}_i(r_0)$ are 3×1 vectors holding the x, y, z components of the transmitted field and the sensitivity function of the receiver [58] respectively, both of which are related to the coordinates of the UXO object given by $\boldsymbol{r}_0 = [x_0, y_0, z_0]^T$. The matrix $\boldsymbol{R}$ is a 3×3 rotation matrix that is related to the 3×1 vector of Euler angles, and $\boldsymbol{\Lambda}_j$ is a 3×3 diagonal matrix with the diagonal elements

$$\lambda_k(\omega_j) = \sum_{\ell=1}^{\infty} \frac{a_{k,\ell}\omega_j\sqrt{-1}}{p_{k,\ell} + \omega_j\sqrt{-1}},\ k = 1, \ldots, 3 \tag{6.2}$$

$$\lambda_k(t_j) = -\sum_{\ell=1}^{\infty} a_{k,\ell} p_{k,\ell} e^{-p_{k,\ell}t} u(t_j),\ k = 1, \ldots, 3 \tag{6.3}$$

where $a_{k,\ell}$ is the expansion coefficient for the ℓth term corresponding to the kth axis, $p_{k,\ell}$ is the ℓth pole of the kth axis, and $u(t)$ is a unit step function. Equations (6.2) and (6.3) apply only to the case of nonferrous objects. When the object is ferrous a DC term must be added to (6.2), and a Dirac delta to (6.3). The final term in (6.1) is an additive Gaussian noise with variance σ_i^2.

The measurement model given in (6.1) is therefore related to the following parameters: object coordinates, object orientation angles, expansion coefficients, and the poles. The UXO classification schemes in [58] only use the poles as the features to the classification algorithms. Furthermore they make use of a single pole for each axis, and these poles are estimated using nonlinear least squares. Let $\boldsymbol{\nu} = [\boldsymbol{a}^T, \boldsymbol{r}_0^T, \boldsymbol{\alpha}^T]^T$ denote the nuisance parameters which are not used in the UXO classification scheme then the nonlinear least squares parameter estimation is obtained using

$$\hat{\boldsymbol{p}}_j = \arg\min_{\boldsymbol{p},\boldsymbol{\nu}} \sum_{i=1}^{N}\sum_{j=1}^{M} (d_{i,j} - s_{i,j}(\boldsymbol{p},\boldsymbol{\nu}))^2/\sigma_i^2, \tag{6.4}$$

which can be solved iteratively using any gradient based optimization algorithm [43], though the solution is only guaranteed to be locally optimal.

6.2 Previous work on parameter estimation with positional uncertainty with the dipole model

A major drawback of the nonlinear least squares approach given in (6.4) is that it assumes that the sensors' locations are known perfectly, whereas in practice the sensors' positions are only known to be in some nominal location. Often dead reckoning or inertial navigation may be used to estimate the sensors' locations, however this process is clearly error prone. Even sensor locations that were obtained using a global positioning system (GPS) are subject to uncertainty which may influence the poles' estimates, and have a negative effect on the classification performance. Parameter estimation in the dipole model with sensor positional uncertainty was considered in [57]. Specifically, a robust approach was proposed in which the poles are estimated by solving a minimax optimization problem that minimizes over the estimates of the poles and the nuisance parameters, and maximizes over the sensors' locations in some region of uncertainty. The minimax problem was solved under several positional uncertainty scenarios, which can also be considered as having some prior model for the positional error accumulation along the course. The fist observation that is used to define the uncertainty region is that they can be bounded by some prior knowledge about the measurement error of the positional measurement device (e.g. GPS). Furthermore, since the physical process of obtaining the measurements involves a human operator that in general can be expected to move in a straight line, one can expect the positional errors to be correlated as well. Let $\boldsymbol{r}_i = \bar{\boldsymbol{r}}_i + \delta\boldsymbol{r}_i$ denote the real coordinates vector of the ith sensor, where $\delta\boldsymbol{r}_i$ is the perturbation from the nominal position $\boldsymbol{r}_i$, then the uncertainty region involves some form of constraints on the perturbation errors $\delta\boldsymbol{r}_i$. Let Ω denote the uncertainty region of the sensors' positions, and assuming that the measurement model (6.1) can be approximated using a first order Taylor series expansion around the nominal sensor position $\bar{\boldsymbol{r}}$, then the parameter estimation takes the form:

$$\min_{\boldsymbol{p},\boldsymbol{\nu}} \max_{\delta \boldsymbol{r}\in\Omega} \sum_{i=1}^{M}\sum_{j=1}^{N} (d_{i,j} - s_{i,j}(\boldsymbol{p},\boldsymbol{\nu},\bar{\boldsymbol{r}}_i) - \boldsymbol{A}_{i,j}\delta\boldsymbol{r}_i)^2 = \min_{\boldsymbol{p},\boldsymbol{\nu}} \max_{\delta \boldsymbol{r}\in\Omega} \sum_{i=1}^{M} \|\bar{\boldsymbol{d}}_i - \bar{\boldsymbol{A}}_i \delta\boldsymbol{r}\|^2 \tag{6.5}$$

where $\boldsymbol{A}_{i,j}$ denotes the 1×3 vector whose elements are the first order term in the Taylor series expansion of $s_{i,j}$ with respect to the sensor coordinates vector $\boldsymbol{r}$, $\bar{\boldsymbol{A}}_i = [\boldsymbol{A}_{i,1}^T, \boldsymbol{A}_{i,2}^T, \dots, \boldsymbol{A}_{i,N}^T]^T$, and $\bar{\boldsymbol{d}}_i$ is the appropriately chosen vector such that the equality holds.

The following uncertainty regions were considered in [57]:

- *Polyhedral and independent:* Under this model the positional error for each sensor is independent of the other sensors, and each sensor is assumed to be within a known polyhedral region around the nominal sensor position. The optimal solution of the maximization part in such a case is achieved at one of the corners of the polyhedral region, and can be found by testing each of the corners to find the one that maximizes the objective function.

- *Ellipsoidal-shaped and independent:* In this case the positional errors are also independent of each other, and the uncertainty region is on the boundary of the ellipsoid $\delta\boldsymbol{r}^T \mathrm{diag}([\ell_x^{-1}, \ell_y^{-1}, \ell_z^{-1}]^T)\delta\boldsymbol{r} = 1$, where ℓ_x, ℓ_y, ℓ_z are the parameters that define the ellipsoid and are obtained from some prior knowledge about the uncertainty region. The closed form solution to the maximization problem under the ellipsoidal uncertainty region was also presented in [57].

- *Box-shaped and dependent:* In this model the positional perturbations of the sensor are assumed to satisfy the constraint $\delta\boldsymbol{r}_i = \alpha\delta\boldsymbol{r}_{i-1} + \delta\boldsymbol{r}_i'$ where $\delta\boldsymbol{r}_i'$ is the innovation. If the innovations were a realization of a Gaussian noise then this model would be identical to a first order autoregressive process, however in this case it is only assumed that $\delta\boldsymbol{r}_i$ are bounded in a box (which also means that the innovation can not be Gaussian since they are bounded as well). The solution to the maximization problem using this uncertainty region can be obtained using DP, where first the innovations $\delta\boldsymbol{r}_i'$ are discreteized, and subsequently the DP algorithm is used to find optimal solution. Clearly as the discretization becomes finer the computational complexity increases significantly, and therefore an approximate DP algorithm was used in practice.

Let $c(\boldsymbol{p},\boldsymbol{\nu})$ denote the optimal solution to the maximization problem using any of the above models, then the approach used in [57] to solve the minimax problem was to use a nonlinear gradient based algorithm in order to find the solution that minimizes $c(\boldsymbol{p},\boldsymbol{\nu})$.

In the next section we present a new approach which extends the box-shaped and dependent perturbation model that was discussed previously.

6.3 A nonlinear programming approach to robust parameter estimation with sensor positional uncertainty

The uncertainty model that we propose follows the dependent perturbations model described in the previous section, where instead of assuming that the innovations are bounded in a box we assume that they are bounded in an ellipsoid. The uncertainty region in this case can be written as

$$\Omega = \{\delta\boldsymbol{r} \Big| \delta\boldsymbol{r}_i^{\prime T}\boldsymbol{R}\delta\boldsymbol{r}_i' \leq L^2/n,\ \delta\boldsymbol{r}_i = \alpha\delta\boldsymbol{r}_{i-1} + \delta\boldsymbol{r}_i'\} \tag{6.6}$$

where n denotes the number of sensors, $\boldsymbol{R} = \text{diag}([\ell_x^{-1}\ \ell_y^{-1}\ \ell_z^{-1}]^T)$, and L is some constant.

Let $\boldsymbol{\Gamma}_{\alpha,i}$ be a matrix such that $\boldsymbol{\Gamma}_{\alpha,i}\delta\boldsymbol{r} = \delta\boldsymbol{r}' \Leftrightarrow \delta\boldsymbol{r}_i = \alpha\delta\boldsymbol{r}_{i-1} + \delta\boldsymbol{r}_i',\ \forall i = 2,\ldots,M$, then the uncertainty region is equivalent to

$$\Omega = \{\delta\boldsymbol{r} \Big| \delta\boldsymbol{r}^T\boldsymbol{\Gamma}_{\alpha,i}^T\boldsymbol{R}\boldsymbol{\Gamma}_{\alpha,i}\delta\boldsymbol{r} \leq L^2/n\} \tag{6.7}$$

The Lagrangian for the inner maximization takes the form

$$\boldsymbol{\mathcal{L}}(\delta\boldsymbol{r},\boldsymbol{\lambda}) = (\tilde{\boldsymbol{d}} - \tilde{\boldsymbol{A}}\delta\boldsymbol{r})^T(\tilde{\boldsymbol{d}} - \tilde{\boldsymbol{A}}\delta\boldsymbol{r}) + \sum_{i=1}^{n}\lambda_i(L^2/n - \delta\boldsymbol{r}^T\boldsymbol{\Gamma}_{\alpha,i}^T\boldsymbol{R}\boldsymbol{\Gamma}_{\alpha,i}\delta\boldsymbol{r}). \tag{6.8}$$

where $\tilde{\boldsymbol{A}} = \text{blkdiag}(\bar{\boldsymbol{A}}_1,\ldots,\bar{\boldsymbol{A}}_M)$, $\tilde{\boldsymbol{d}} = [\bar{\boldsymbol{d}}_1^T,\ldots,\bar{\boldsymbol{d}}_M^T]^T$. Maximizing with respect to $\delta\boldsymbol{r}$ we have

$$\frac{\partial\boldsymbol{\mathcal{L}}(\delta\boldsymbol{r},\boldsymbol{\lambda})}{\partial\delta\boldsymbol{r}} = -2\tilde{\boldsymbol{A}}^T(\tilde{\boldsymbol{d}} - \tilde{\boldsymbol{A}}\delta\boldsymbol{r}) - 2\sum_{i=1}^{n}\lambda_i\boldsymbol{\Gamma}_{\alpha,i}^T\boldsymbol{R}\boldsymbol{\Gamma}_{\alpha,i}\delta\boldsymbol{r} = 0, \tag{6.9}$$

and therefore the solution for $\delta\boldsymbol{r}$ is

$$\delta\boldsymbol{r}^* = -(\sum_{i=1}^{n}\lambda_i\boldsymbol{\Gamma}_{\alpha,i}^T\boldsymbol{R}\boldsymbol{\Gamma}_{\alpha,i} - \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}})^{-1}\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{d}}. \tag{6.10}$$

Using (6.10) in (6.8) we obtain the dual function

$$g(\boldsymbol{\lambda}) = \tilde{\boldsymbol{d}}^T\tilde{\boldsymbol{d}} + \tilde{\boldsymbol{d}}^T\tilde{\boldsymbol{A}}(\sum_{i=1}^{n}\lambda_i\boldsymbol{\Gamma}_{\alpha,i}^T\boldsymbol{R}\boldsymbol{\Gamma}_{\alpha,i} - \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}})^{-1}\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{d}} + L^2/n\sum_{i=1}^{n}\lambda_i. \tag{6.11}$$

By minimizing the dual function we obtain the dual problem for the inner maximization which takes the form

$$\min_{\boldsymbol{\lambda}}\left\{\tilde{\boldsymbol{d}}^T\tilde{\boldsymbol{d}} + (\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{d}})^T(\sum_{i=1}^{n}\lambda_i\boldsymbol{\Gamma}_{\alpha,i}^T\boldsymbol{R}\boldsymbol{\Gamma}_{\alpha,i} - \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}})^{-1}(\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{d}}) + L^2/n^2\sum_{i=1}^{n}\lambda_i\right\} \tag{6.12}$$

subject to

$$\begin{array}{c} \sum_{i=1}^{n}\lambda_i\boldsymbol{\Gamma}_{\alpha,i}^T\boldsymbol{R}\boldsymbol{\Gamma}_{\alpha,i} - \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}} \succeq 0 \\ \boldsymbol{\lambda} \geq 0 \end{array} \tag{6.13}$$

Using Lemma 1 it is straightforward to show that the optimization problem (6.12) subject to (6.13) is convex, and can be written as

$$\min_{\boldsymbol{\lambda},\tau}\{\tilde{\boldsymbol{d}}^T\tilde{\boldsymbol{d}} + \tau + L^2/n\sum_{i=1}^{n}\lambda_i\} \tag{6.14}$$

subject to

$$\begin{array}{c} \left[\begin{array}{cc} \tau & (\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{d}})^T \\ \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{d}} & \sum_{i=1}^{n}\lambda_i\boldsymbol{\Gamma}_{\alpha,i}^T\boldsymbol{R}\boldsymbol{\Gamma}_{\alpha,i} - \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}} \end{array}\right] \succeq 0 \\ \boldsymbol{\lambda} \geq 0 \end{array} \tag{6.15}$$

The dual problem for the inner maximization is therefore a semidefinite program. Since the primal problem is not convex (it is a maximization problem over a convex function), Slater's conditions are not satisfied and strong duality is not guaranteed. The optimal value for the dual is an upper bound on the optimal value for the primal. Hence solving the dual problem provides an upper bound on the primal. Moreover it can easily be shown to be finite. We show this by considering the dual problem of (6.5) with the uncertainty region given by

$$\Omega = \{\delta\boldsymbol{r}\Big|\sum_{i=1}^{n}\delta\boldsymbol{r}^T\boldsymbol{\Gamma}_{\alpha,i}^T\boldsymbol{R}\boldsymbol{\Gamma}_{\alpha,i}\delta\boldsymbol{r} \leq L^2\}. \tag{6.16}$$

In this case we obtain that the dual problem is also a semidefinite program that takes the form

$$\min_{\lambda,\tau}\{\tilde{\boldsymbol{d}}^T\tilde{\boldsymbol{d}} + \tau + \lambda L^2\} \tag{6.17}$$

subject to

$$\begin{gathered}\begin{bmatrix} \tau & (\tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{d}})^T \\ \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{d}} & \lambda\sum_{i=1}^{n}\boldsymbol{\Gamma}_{\alpha,i}^T\boldsymbol{R}\boldsymbol{\Gamma}_{\alpha,i} - \tilde{\boldsymbol{A}}^T\tilde{\boldsymbol{A}} \end{bmatrix} \succeq 0 \\ \boldsymbol{\lambda} \geq 0 \end{gathered} \tag{6.18}$$

Let λ^* denote the optimal solution to the problem (6.17), (6.18) then we can obtain the same objective value in (6.14), (6.15) simply by taking $\lambda_i = \lambda^*/n$, and therefore the solution to the second problem is an upper bound on the solution to the first problem. Furthermore since in the second problem we have an optimization problem over a quadratic objective with a single quadratic inequality, strong duality holds even though the problem is not convex. The solution to the second problem is bounded, and since the second problem is an upper bound on the first we have that the solution to the first problem is bounded as well.

By replacing the maximization problem in (6.5) with its dual problem, which is a minimization problem, we can solve for the parameters by minimizing (6.14), (6.15) with respect to the parameters $\boldsymbol{p}$ and $\boldsymbol{\nu}$ as well. The problem is no longer convex in this case, however there exist techniques that solve such nonlinear semidefinite programs iteratively [61], [62] and are guaranteed to converge to a locally optimal solution.

Chapter 7

Conclusions and Future Work

In this dissertation we studied difference estimation scenarios, and presented new methods that address important issues and challenges in the areas of nonlinear and robust estimation. The major contributions of this dissertation in the area of nonlinear filtering are the introduction of a new proposal distribution, the AMPF-IS which was shown to improve the AMPF proposal distribution significantly, and the new performance bounds for the IFGT which facilitate the choice of the parameters of the IFGT and enable its use within the filtering framework. The significance of the new IFGT performance bounds extends beyond the area of nonlinear filtering, and is also important to many other fields that rely on fast KDE such as computer vision [3] and machine learning [63]. The new bounds can also be easily used to develop a hybrid dual tree IFGT algorithm for fast kernel density estimation, building on the strengths of each of these algorithms. Such an algorithm was developed for the FGT [64] however it was not developed for the IFGT because of the several issues that existed with the previously existing bounds. The minor contribution of this dissertation in the area of nonlinear filtering is the application of the IFGT to the particle filtering problem. Previous work has employed the FGT for this purpose [2], however in this work we used the IFGT which performs better than the FGT in high dimensions.

In the area of robust estimation we considered robust estimation in the linear Gaussian model. The major contribution of this dissertation in this field was the introduction of the GDR estimator which extends the difference regret estimator presented in [28] such that the limiting assumption of joint diagonalizability is no longer required, and to define the region of uncertainty as the intersection between an eigenvalue and an elementwise uncertainty set. Both of these extensions were demonstrated to improve the results obtained using the difference regret estimator and the MMSE estimator. We also showed that when joint diagonalizability holds and when considering only eigenvalue uncertainties then the GDR estimator is identical to the difference regret estimator. The most interesting possible extension to this line of work is to find the GDR estimator with uncertainties on the elements of the inverse covariance matrix. Furthermore, since we showed that the GDR criterion that we used is an upper bound on the difference regret provided that joint diagonalizability holds, it is interesting to develop other bounds on the MSE using more principled methods such as the Weiss-Weinstein bound [65] [66], and see if they are useful for developing new criteria for the minimax problem. This may also be a good approach in order to solve the problem when there are elementwise uncertainties in the inverse covariance matrix.

Finally we considered the use of the robust optimization techniques that were discussed in this dissertation to robust parameter estimation under sensor positional uncertainty. The robust parameter estimation problem is motivated by a UXO classification problem which uses these parameters as the features for the classification problem. A new nonlinear programming approach was proposed to replace the approximate dynamic programming approach that was considered in [57]. Future work will include the simulation and validation of the nonlinear programming approach to the problem.

Bibliography

[1] B. D. O. Anderson, and J. B. Moore, *Optimal Filtering*, Prentice-Hall, 1979.

[2] M. Klaas, N. de Freitas and A. Doucet, "Toward practical N^2 Monte Carlo: the marginal particle filter," *21st Conference on Uncertainty in Artificial Intelligence*, 2005.

[3] A. Elgammal, R. Duraiswami, and L. S. Davis, "Efficient Kernel density estimation using the fast Gauss transform with applications to color modeling and tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1499–1504, Nov. 2003.

[4] K. Deng, and A. W. Moore, "Multiresolution instance-based learning," *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pp. 1233–1239, 1995.

[5] L. Greengard, and J. Strain, "The fast Gauss transform," *SIAM J. Scientific Computing*, vol. 12, no. 1, pp. 79–94, 1991.

[6] B. J. C. Baxter, and G. Roussos, "A new error estimate of the fast Gauss transform," *SIAM J. Scientific Computing*, vol. 24, no. 1, pp. 257–259, 2002.

[7] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis, "Improved fast Gauss transform and efficient Kernel density estimation," *IEEE International Conference on Computer Vision*, pp. 464–471, 2003.

[8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[9] A. G. Gray, and A. W. Moore, "Rapid evaluation of multiple density models," *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Jan 2003.

[10] C. P. Robert, and G. Casella, "Monte Carlo Integration," *Monte Carlo Statistical Methods*, Springer-Verlag, 2004.

[11] D. Guo, and X. Wang, "Quasi-Monte Carlo filtering in nonlinear dynamic systems," *IEEE Trans. on Signal Processing*, vol. 54, no. 6, pp. 2087–2098, June. 2006.

[12] P. Hellekalek, and G. Larcher, ed., *Random and Quasi-Random Point Sets*, , Lecture Notes in Statistics, Springer-Verlag, 1998.

[13] M. K. Pitt, and N. Shephard, "Filtering via simulation: auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.

[14] D. Lang, M. Klaas and N. de Freitas, *Empirical testing of Fast Kernel density estimation algorithms*, UBC TR-2005-03.

[15] N. I. Fisher, *Statistical Analysis of Circular Data*, Cambridge University Press, 1993.

[16] J. H. Kotecha, P. M. Djurić, "Gaussian particle filtering," *IEEE Trans. on Signal Processing*, Vol 51, No. 10, 2003.

[17] V. C. Raykar, C. Yang, R. Duraiswami, and N. Gumerov, *Fast computation of sums of Gaussians in high dimensions*, CS-TR-4767, Department of computer science, University of Maryland, Collegepark, 2006.

[18] R. Mittelman, and E. L. Miller, "Fast Gauss transforms based on a high order singular value decomposition for nonlinear filtering," *IEEE Statistical Signal Processing Workshop* 2007.

[19] F. Campillo, and V. Rosi, "Convolution particle filtering for parameter estimation in general state-space models," *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006.

[20] C. Chang and R. Ansari, "Kernel particle filter for visual tracking," *IEEE Signal Processing Letters*, Vol. 12, No. 3, 2005.

[21] *Sequential Monte Carlo Methods in Practice*, Ed. A. Doucet N. de Freitas, and N. J. Gordon, 2001.

[22] A. T. Ihler, J. W. Fisher, and A. S. Willsky, "Particle filtering under communications constraints," *IEEE Statistical Signal Processing Workshop* 2005.

[23] J. H. Friedman, J. L. Bently, and R. A. Finkel, "An algorithm for finding the best matches in logarithmic expected time", *ACM Transactions on Mathematical Software*, Vol. 3 No. 3, 1977.

[24] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, *The unscented particle filter*, Technical report CUED/F-INFENG/TR 280, Cambridge University Engineering Department.

[25] *Convex Analysis*, R. T. Rockafellar, Princeton University Press, 1970.

[26] Y. C. Eldar, A. Ben-Tal, and A. Nemirovsky, "Robust mean-squared error estimation in the Presence of Model Uncertainties," *IEEE Trans. on Signal Processing*, Vol. 53, No. 1, Jan 2005.

[27] Y. C. Eldar, A. Ben-Tal and A. Nemirovski, "Linear minimax regret estimation of deterministic parameters with bounded data uncertainties," *IEEE Trans. Signal Processing*, Vol. 52, No. 8, Aug. 2004.

[28] Y. C. Eldar, and N. Merhav "A competitive minimax approach to robust estimation of random parameters," *IEEE Trans. on Signal Processing*, Vol. 52, No. 7, July 2004.

[29] Y. C. Eldar, and N. Merhav "Minimax MSE-ratio estimation with signal covariance uncertainties," *IEEE Trans. on Signal Processing*, Vol. 53, No. 4, April 2005.‘

[30] S. Verdu, and H. V. Poor, "On minimax robustness: a general approach and applications," *IEEE Trans. on Inform. Theory*, Vol. 30, No. 2, March 1984.

[31] S. A. Kassam, and H. V. Poor, "Robust techniques for signal processing: a survey," *Proceedings of the IEEE*, Vol. 73, No. 3, March 1985.

[32] S. Boyd, and L. Vandenberghe, *Convex Oprimization*, Cambridge University Press, 2004.

[33] L. Vandenberghe, and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 40–95, March 1996.

[34] Y. Nesterov, and A. Nemirovsky, *Interior-Point Polynomial Algorithms in Convex Programming*, Philadelphia, PA:SIAM, 1994.

[35] M. J. Wainwright and M. .I. Jordan, "Log-determinant relaxation for approximate inference in discrete Markov random fields," *IEEE Trans. on Signal Processing*, Vol. 54, No. 6, June 2006.

[36] S. L. Lauritzen, *Graphical models*, Oxford University Press, 1996.

[37] E. B. Sudderth, M. J. Wainwright and A. .S. Willsky, "Embedded trees: estimation of Gaussian processes on graphs with cycles," *IEEE Trans. on Signal Processing*, Vol. 54, No. 6, June 2006.

[38] V. Delouille, R. Neelamani and R. .G. Baraniuk, "Robust distributed estimation using the embedded subgraph algorithm," *IEEE Trans. on Signal Processing*, Vol. 54, No. 8, August 2006.

[39] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*, John Wiley and Sons Inc., 1968.

[40] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*, Addison Wesley Publishing Company, 1991.

[41] C. Y. Chong, and S. .P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, Vol. 91, No. 8, August 2003.

[42] H. A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, January 2004.

[43] D. G. Luenberger *Linear and Nonlinear Programming.* Reading, MA. : Addison-Wesley, 1984.

[44] C. E. Rasumussen, and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[45] A. Krause, A. Singh, and C. Guestrin, "Near optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*: 9, 2008.

[46] R. Mittelman, and E. L. Miller, "Nonlinear filtering using a new proposal distribution and the improved fast Gauss transform with tighter performance bounds," *IEEE Trans. on Signal Processing*, Vol. 56, No. 12, December 2008.

[47] M. Seeger, "Gaussian processes for machine learning," *International Journal of Neural Systems*, 14(2):69–106, 2004.

[48] D. J. Nott, and W. T. M. Dunsmuir, "Estimation of nonstationary spatial covariance structure," *Biometrica*, 89(4):819–829, 2002.

[49] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems*, pp. 50-61, 2004.

[50] M. Yuan, and Y. Lin, "Model selection and estimation in the Gaussian graphical model," *Biometrika*, February 28, 2007.

[51] E. B. Sudderth, *Embedded trees: Estimation of Gaussian processes on graphs with cycles*, Masters thesis, Mass. Inst. Technol.,Feb 2002. [online] Available: http://ssg.mit.edu/ esuddert Biometrika, February 28, 2007.

[52] A. Papoulis, *Probability, random variables, and stochastic processes.* Singapore: McGraw-Hill, 1991.

[53] X. Mestre, "Improved estimation of eigenvalues and eigenvectors of covariance matrices using their sample estimates," *IEEE Trans. on Inform. Theory*, Vol. 54, No. 11, November 2008.

[54] R. M. Gray, "Toeplitz and Circulant Matrices: a Review," *Now Publishers Inc.*, Boston-Delft, 2005.

[55] M. Yamashita, K. Fujisawa, and M. Kojima, "Implementation and evaluation of SDPA 6.0 (SemiDefinite Programming Algorithm 6.0)," *Optimization Methods and Software* 18, 491–505, 2003.

[56] M. Brookes, *The Matrix Reference Manual*, [online] http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html, 2005.

[57] A. B. Tarokh, and E. L. Miller, "Subsurface sensing under sensor positional uncertainty," *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 45, No. 3, March 2007.

[58] A. Aliamiri, J. Stalnaker, and E. L. Miller, "Statistical classification of buried unexploded ordnance using nonparametric prior models," *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 45, No. 9, September 2007.

[59] L. Carin, H. T. Yu, and C. E. Baum, “On the wideband electromagnetic inductance signature of conducting and permeable targets,” *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 39, No. 6, pp. 1206–1213, June 2001.

[60] N. Geng, C. E. Baum, and L. Carin, “On the low-frequency natural response of conducting and permeable targets,” *IEEE Trans. on Geoscience and Remore Sensing*, Vol. 37, No. 1, pp. 347–359, January 1999.

[61] C. Kanzow, C. Nagel, H. Kato. and M. Fukushima, “Successive linearization methods for nonlinear semidefinite programs,” *Computational Optimization and Applications*, 31, pp. 251–273, 2005.

[62] R. Correa, and H. Ramirez C., “A global algorithm for nonlinear semidefinite programming,” *SIAM Journal of Optimization*, Vol. 15, No. 1, pp. 303–318, 2004.

[63] A. Gray, “Fast Kernel Matrix-Vector Multiplication with Application to Gaussian Process Regression,” *Carnegie Mellon technical report* CMU-CS-04-110, 2004).

[64] D. Lee, A. Gray, and A. Moore, “Dual tree fast Gauss transforms,” *NIPS*, 2005.

[65] A. J. Weiss and E. Weinstein, “A lower bound on the mean-square error in random parameter estimation,” *IEEE Trans. on Information Theory*, Vol. 31, No. 5, pp. 680–682, September 1985.

[66] A. Renaux, P. Forster, P. Larzabal, C. D. Richmond, and A. Nehorai, “A fresh look at the Bayesian bounds of the Weiss-Weinstein family,” *IEEE Trans. on Signal Processing*, Vol. 56, No. 11, pp. 5334–5352, November 2008.