Sparse Sensing for Statistical Inference

Theory, Algorithms, and Applications

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft, op gezag van
de Rector Magnificus prof. ir. K.Ch.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
xxdag dd month 2015 om hr:mm uur door

Sundeep Prabhakar Chepuri,

elektrotechnisch ingenieur,
geboren te Bangalore, India.

Dit proefschrift is goedgekeurd door de promotors:
Prof. dr. ir. G.J.T. Leus
Prof. dr. ir. A.-J. van der Veen

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. G.J.T. Leus,	Technische Universiteit Delft, promotor
Prof. dr. ir. A.-J. van der Veen,	Technische Universiteit Delft, promotor

ISBN xxx-xx-xxxx-xxx-x

To my parents, Souju, and Madhu, who made me who I am. To Kavya, for her ∞ love and patience.
To my advisors, for what you are reading.

Contents

Chapter 1

Introduction

Contents

1.1 Pervasive sensors and data deluge

Every day, we are generating data of the order of a billion gigabytes. This massive volume—exascale data comes from omnipresent sensors used in medical imaging (e.g., breast or fetal ultrasound), seismic processing (e.g., from oil or gas field exploration), environmental monitoring (e.g., pollution, temperature, precipitation sensing), radio astronomy (e.g., from radio telescopes like the square kilometre array), power networks (e.g., to monitor wind farms or other distribution grids), smart infrastructures (e.g., to monitor the condition of railway tracks or bridges), localization and surveillance platforms (e.g., security cameras or drones, indoor navigation), and so on.

The acquired data samples are stored locally and then transported to a central location (e.g., a server or cloud) to extract meaningful information (that is, for inference). Due to an unprecedented increase in the volume of the acquired data, it is becoming increasingly challenging to locally store and transport all the data samples to a central location for data/signal processing. This is because the amount of the sampled data quickly exceeds the storage and communication capacity by several orders of magnitude. Since the data processing is generally carried out at a server with ample computing power (e.g., Hadoop clusters), mainly the storage and transportation costs form the bottleneck of big data processing. To alleviate these bottlenecks, most of the data is blindly discarded without even being examined in order to limit the memory and communication requirements, causing a serious performance loss.

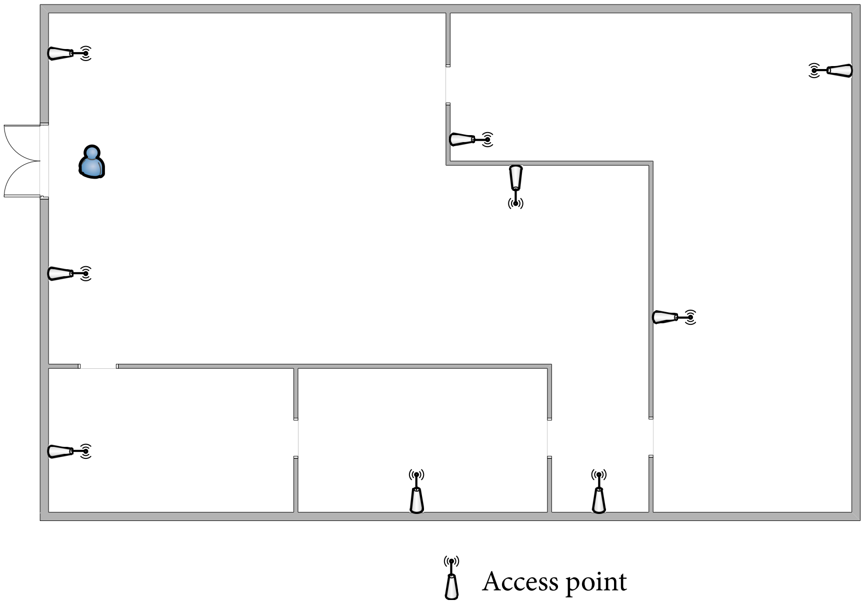
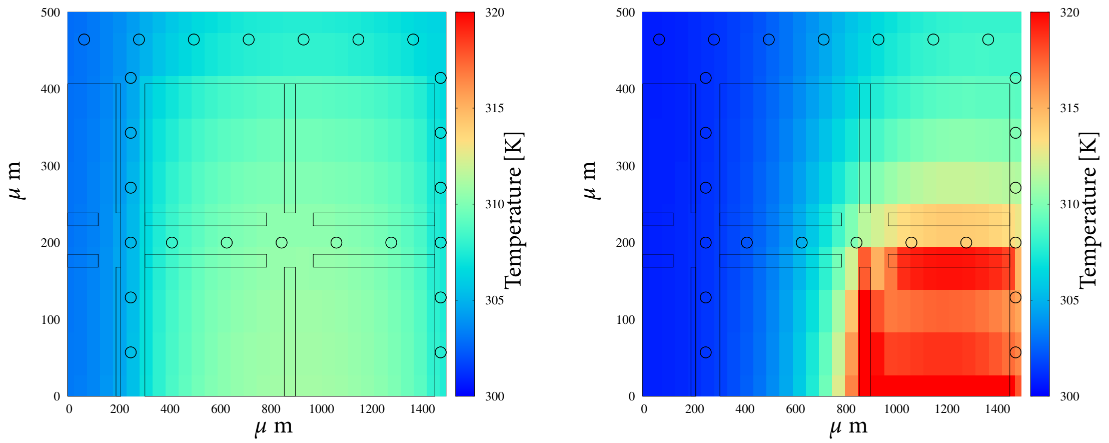


Figure 1.1: Illustration of an indoor localization setup. We show the floor plan of a building (e.g., museum) with candidate locations for installing the access points. The restriction on installing the access points in only certain areas might be for security or ambience purposes.

If we had some prior knowledge about the task we want to perform on the data samples, then just a small portion of that data might be sufficient to reach a desired inference accuracy, thereby significantly reducing the amount of sampled and transported data. That is to say, if the inference task is known beforehand, less data needs to be acquired, thereby significantly reducing the amount of sampled and transported data. Thus, the memory and bandwidth requirements can be seriously curtailed. In addition, the cost of data collection (or sensing) can be significantly reduced, where the major factors that determine the sensing costs are the number of physical sensors (thus, their economical and energy costs) and the physical space they occupy when installed. So, it is evident that there is an urgent need for developing unconventional and innovative sensing mechanisms tailored for specific inference tasks to extract as much information as possible yet collecting fewer data. This leads us to the first question:

Q1. *How can task-cognition be exploited to reduce the costs of sensing as well as the related storage and communications requirements?*

Given the central role of sampling in engineering sciences, answering this question will impact a wide range of applications. The basic question of interest for such applications is how the sensing systems should be designed to minimize the amount of data acquired yet reach a desired inference performance. In particular, the design questions that should be answered are related to the opti-



(a)

(b)

Figure 1.2: Heatmaps of a 32KB data cache (a) without and (b) with a hot spot. Black circles (○) denote the candidate temperature sensor locations— these are the areas with less or no active logic.

mal sensor placement in space and/or time, data rate, and sampling density. We next illustrate two specific examples of sensor placement for indoor localization and temperature sensing.

Example 1.1 (Target localization)**.** *Indoor localization is becoming increasingly important in many applications. Some examples include: locating people inside a building for rescue operations, monitoring logistics in a production plant, lighting control, and so on. In such environments, global positioning system (GPS) signals are typically unavailable. Thus, other types of measurements such as visual, acoustic or radio waves revealing information about range, bearing, and/or Doppler are used. These measurements are gathered by access points, like cameras, microphones, radars, or wireless transceivers. One such scenario is illustrated in Fig. 1.1, where we show an indoor localization setup for navigating a visitor inside a building. An interesting question is, instead of installing many such costly access points randomly, how can we minimize the number of access points (hence, the amount of data), by optimizing their characteristics (e.g., their spatial position) in such a way that a certain localization performance can be guaranteed.*

Example 1.2 (Field detection)**.** *Consider a multi-core processor with a hot spot. A historical question of interest is to estimate the thermal distribution, for instance, by interpolating noisy measurements. In some applications, though, a precise estimation of the temperature field might not be required, instead, detecting the hot spots (i.e., the areas where the temperature exceeds a certain threshold) would be sufficient for subsequent control actions. Such a scenario is illustrated in Fig. 1.2*[1]*, where the image on the*

[1] We would like to thank Sumeet Kumar for the heatmaps [Kumar et al., 2015] .

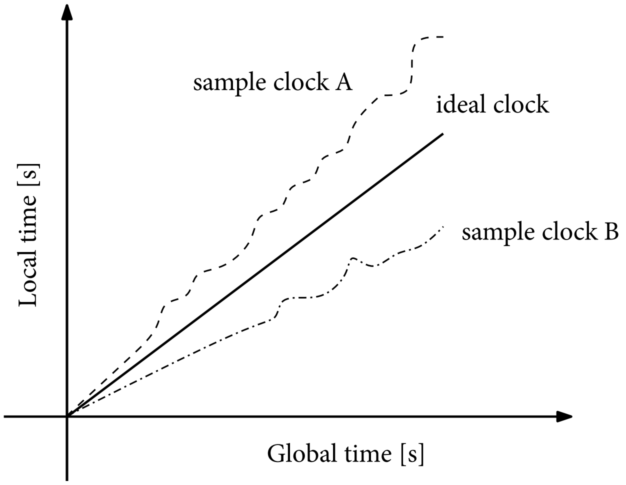


Figure 1.3: Illustration of two sample clocks drifting from each other with respect to an ideal clock.

right (left) shows a 32 KB data cache with (no) hotspots. An important question [Memik et al., 2008] of interest then is spatial sampling design, i.e., sensor placement, for such detection problems exploiting the knowledge of the underlying model, and accounting for the physical space and/or processing limitations.

Such optimally designed sensing systems can be used to perform a number of inference tasks. In the next section, we will introduce sensor networks as a spatial sampling device and discuss some related signal processing applications.

1.2 Distributed sensing: synchronization and localization

Over the past decade, advances in wireless sensor technology have enabled the usage of sensors to connect almost everything as a network. This so-called internet of things (IoT) is used for different purposes related to sensing, monitoring, and control. Such networks find applications ranging from monitoring natural ecosystems to buildings, industrial equipments, and vehicles, from military to civil localization applications, to name a few.

Due to the inherent discrete nature of the sensors—spatially localized objects, a sensor network performs sampling in space [Gastpar et al., 2006]. For many cases that we frequently encounter, a sensor network can be designed to faithfully represent distributed signals (e.g., a spatially varying phenomenon such as the temperature field). In addition, the distributed signals can be multidimensional, that is, they can exist in space and time. To acquire spatiotemporal variations of such distributed signals, we need to sample over both space and time, where the temporal sampling is achieved using the sensor's analog-to-digital converters (ADCs) or time-to-digital converters (TDCs), for example. Each sensor has an independent sample clock (i.e., oscillator), and its stability essentially determines the alignment of the temporal sampling grid across the sensors. This temporal sampling grid is perfectly aligned if all the sensors share a common clock. However, when uncommon, the sample clocks drift from each other due to imperfections in the oscillator, aging and other environmental factors (e.g., temperature variation, vibration). We illustrate two sample clocks drifting from an ideal (or a reference) clock in Fig. 1.3. This drift will result in the misalignment of the temporal sampling grid across the sensors. Therefore, we need to align them from time to time. In other words, we need to devise a mechanism to distribute the sample clock wirelessly. This brings us to the second question:

Q2. *How can wireless communications be exploited to synchronize spatially separated sample clocks?*

Answering this question impacts a range of other sensor network applications that demand for a common time frame for the entire network, such as sleep and wake-up coordination, time-based channel access, among others [Freris et al., 2010].

A vast number of the applications that use sensor networks rely on a fundamental aspect of either associating the location information to the data that is acquired by spatially distributed sensors (e.g., air quality measurements), or the acquired data is solely used to localize a target/source (e.g., indoor localization). One way to do this is to equip each sensor node with a GPS receiver, however, in many applications of interest the operating environment is harsh with GPS signals either being impaired or unavailable. Moreover, sensors are usually battery powered making GPS a less viable option. To facilitate low-power and efficient localization solutions especially in GPS-denied environments, there exists a plethora of algorithms based on two localization paradigms: absolute or relative localization. In absolute localization, the aim is to estimate the absolute position of the sensor(s) using a few reference nodes whose absolute positions are known. Hence, these reference nodes are commonly referred to as *anchors*. Absolute localization problems are typically solved using measurements related to the propagation of radio or acoustic waves, e.g., time-of-arrival (TOA), time-difference-of-arrival (TDOA), received signal strength (RSS), or angle-of-arrival (AOA), to list a few [Patwari et al., 2005, Gezici et al., 2005, Gustafsson and Gunnarsson, 2005]. Localization can also be relative, in which case the aim is to estimate the constellation of the sensors or the topology of the network, and determining the location of a sensor relative to the other sensors is sufficient. Classic solutions to relative localization are based on multi-dimensional scaling (MDS) using squared-range measurements [Cheung and So, 2005, Costa et al., 2006]. For relative localization, anchors are not needed.

The use of sensor networks to remotely monitor hazardous environments that are beyond human reach (e.g., leakage in oil pipes, surveillance of nuclear plants, or health of industrial machines) is gaining strong interest. Such tasks are generally performed using robots or drones (more generally, a sensing platform) with a number of sensors mounted on them. Consequently we now have to localize more than one sensor or even localize the whole sensing platform, and typically we know beforehand the sensor placement on the platform. This distributed setting creates a need to solve the following problem:

Q3. *How can we extend the classical localization paradigm to localize a sensing platform by exploiting the knowledge of the sensor placement on the platform?*

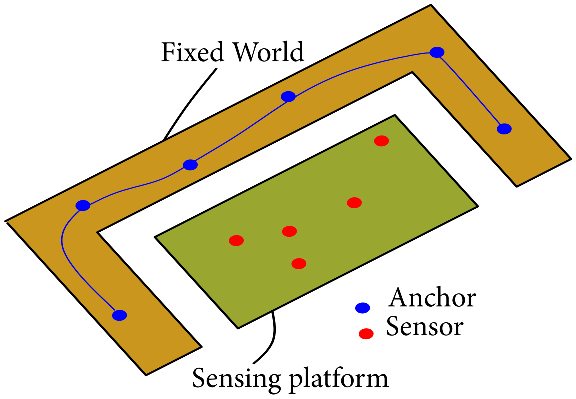


Figure 1.4: Illustration of distributed sensing with sensors mounted on a (moving) sensing platform. The sensor data has to be shipped to a central location with possibly multiple receivers mounted on a fixed world.

We next illustrate the aforementioned problems (i.e., Q2 and Q3) with the following scenario.

Example 1.3 (Sensors on a platform)**.** *Consider a number of sensors mounted on a moving platform as shown in Fig. 1.4. These autonomous sensors collect data related to different physical phenomena, like temperature, vibration, pressure, and so on. This data has to be shipped to a central location (having several wireless transceivers —anchors) on a fixed platform. This is a typical setup in industrial machines or robots, where we generally know the sensor placement on the platform. However, the absolute position of the platform might not be known. Now the questions of interest are: can we use the available wireless links between the sensors and anchors to (A) synchronize the sample clocks and (B) localize such rigid platforms?*

In the next section, we will discuss the context of this thesis and also pose the urgent questions that are addressed in this thesis.

1.3 Scope and context of this thesis

The research for this thesis was generously sponsored by the following two NWO/STW projects:

- VICI-SOWN: The VICI project on signal processing for self-organizing wireless networks (SOWN) aims at developing new mathematical signal processing tools for energy-efficient distributed information processing, spectral sensing, and localization in large sensor networks.

- FASTCOM: The reliable and fast wireless communication for lithography machines (FASTCOM) project aims to connect a sensor network on a moving platform to a control unit using high-speed links with low latency. To realize such a network, accurate sample clock synchronization and optimal sensing design to collect as little data as possible are needed.

Within the framework of the above two projects, we next pose the subquestions related to Q1, Q2, and Q3 that we have answered in this thesis. These subquestions are of general interest (impacts current hot topics like big data analytics, compressive sensing, internet of things) and goes far beyond the scope of VICI-SOWN and FASTCOM.

In order to reduce the sensing and other related costs, it is crucial to tailor the sensing mechanism for the specific inference task that will be performed on the acquired data samples. The tool that we will exploit in this thesis to reduce these costs is *sparse sensing*, which consists of an optimally designed structured and deterministic sparse (i.e., with many zeros and a few nonzeros) sensing function that is applied to the data in order to reach a desired inference performance. Here, the number of nonzeros determines the amount of data samples acquired (thus, determines the amount of data reduction). This naturally leads to a number of questions related to the definition of the inference task and the related performance metric, which we pose as the following subquestions of Q1:

Q1.1. How do we model sparse sensing functions to carry out fundamental signal processing tasks, like estimation, filtering, and detection?

Q1.2. What are the reasonable inference performance metrics for the above tasks?

Q1.3. Can we efficiently optimize (e.g., using a polynomial time algorithm) such inference performance metrics to obtain the sparse samplers of interest?

Next, for a network comprising of several sensor nodes with independent clock oscillators, we aim to distribute the clock signal wirelessly. In other words, we address the problem of synchronizing the sample clocks of the sensor nodes in a network. The assumption is that there are several sensor units at known relative locations (absolute locations are, however, not known) and one sensor unit at an unknown location. The sensors have unreliable and uncommon clocks, except for one of them, which has a relatively stable clock. The goal is to estimate the clock deviations using time-of-arrival measurements of messages. To this end, we pose the following subquestions of Q2:

Q2.1. What is a reasonable parametric representation for the clock deviations?

Q2.2. How can we fully exploit the broadcast nature of the wireless channel for clock synchronization?

Q2.3. Is there an efficient estimator (e.g., unbiased and linear) to resolve the clock parameters and what are the theoretical limits on the variation of the estimates, that is, what is the Cramér-Rao bound (CRB)?

Finally, we consider the problem of localizing a sensing platform using sensor networks. The assumption is that the fixed world has several transceivers with known locations (anchors) and the sensing platform has several sensor units at known relative locations, and the platform is rigid; cf. Fig 1.4. The aim is to localize, that is, to estimate the position and orientation of the rigid platform using distance measurements. We refer to this problem as *rigid body localization*. The related subquestions of Q3 are:

Q3.1. Is there a parametric representation for the rigid body localization problem?

Q3.2. What are the theoretical limits (e.g., CRB) on the variation of the position and orientation estimates computed using distance measurements?

Q3.3. Can we solve the rigid body localization problem if the known sensor topology is perturbed, that is, if the body is not fully rigid?

In the next section, we will discuss the main results and organization of this thesis.

1.4 Summary of results and outline of the thesis

This thesis is organized into three parts. In the first part of this thesis, the theory and algorithms of sparse sensing are discussed in depth. That is, we address subquestions Q1.1-Q1.3. In the second part of this thesis, applications of distributed sensing, more specifically, clock synchronization and localization of a rigid platform are studied (here, we address subquestions Q2.1-Q2.3 and Q3.1-Q3.3). Finally, the thesis concludes with the third part, where we pose some interesting open problems for future research. The content of Chapter 3 till Chapter 8 are published as papers, however with some new subtopics. The relation between these chapters and the publications is depicted in Table 1.1, where the list of publications is provided in Section 1.5.

Chapter 2 on *sparse sensing* fairly forms the backbone of the first part of this thesis. In this chapter, we will model the sparse sensing function as a linear projection operation, where the sensing function is parameterized by a sparse vector. This vector is basically a design parameter that is used as a handle to trade the amount of acquired data samples with the inference performance. We refer to this sparse sensing scheme as *discrete sparse sensing*, as the continuous observation domain is first discretized into grid points and we select (using the sparse vector) the best subset out of those grid points. To harness the full potential of sparse sensing, we need to sample in between the grid points and take samples anywhere in the continuous observation domain. We refer to such sensing mechanisms as *continuous sparse sensing*. We will discuss some applications of the proposed sparse sensing mechanisms and also list major differences with the state of the art in data reduction, that is, compressed sensing. Although the inference task is kept abstract in this chapter, the obtained novel unifying view allows us to jointly treat sparse sensing mechanisms for different inference tasks.

Chapter 3 focuses on discrete sparse sensing for a general *nonlinear estimation* problem. In particular, we solve the problem of choosing the best subset of observations that follow general nonlinear models (e.g., nonGaussian, nonadditive) with uncorrelated inputs. We also extend this framework to nonlinear Gaussian observations with correlated inputs. The data is acquired using the discrete sparse sensing function, which is guided by a sparse vector. The CRB is used as an inference performance metric and we derive several functions of the CRB that include the sparse vector. To compute the sparse samplers, we propose convex relaxations of the

	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8
J1	•					
J2			•			
J3				•		
J4					•	
J5						•
C1	•	•	•			
C2	•					
C3	•					
C4		•				
C5		•				
C6			•			
C7						•
C8					•	
C9						•

Table 1.1: Connection between the papers and chapters.

derived inference performance metric and also develop low-complexity solvers. In sum, the discrete sparse samplers for nonlinear inverse problems can be computed by solving a convex program.

Chapter 4 extends the theory developed in Chapter 3 to *nonlinear filtering* problems, that is, the focus will be on the design of discrete sparse sensing functions for systems that admit a nonlinear state-space representation. In particular, we solve the problem of choosing the best subset of time-varying observations based on the entire history of measurements up to that point. The posterior CRB is used as the inference performance metric to decide on the best subset of observations. Although this framework is valid for independent observations that follow general distributions (e.g., nonGaussian), we also extend it to Gaussian dependent observations. Further, we introduce some additional constraints to obtain smooth sensing patterns over time. Finally, we devise sparse sensing mechanisms for *structured* time-varying observations (e.g., for time-varying sparse signals). In all these cases, the discrete sparse samplers can be designed by solving a convex program.

Chapter 5 is dedicated to discrete sparse sensing for *statistical detection*. Specifically, the aim is to choose the best subset of observations that are conditioned on the hypothesis, which belongs to a binary set. Naturally, the best subset of the observations is the one that results in a desired global error probability. Since the numerical optimization of the error probabilities is difficult, we adopt simpler costs related to distance measures between the conditional distributions of the sensor observations. We design sparse samplers for the Bayesian and Neyman-Pearson setting, where we respectively use the Bhattacharyya distance and Kullback-Leibler distance (and J-divergence) as the inference performance metric. For conditionally independent observations, we give an explicit solution, which is optimal in terms of the error exponents. More specifically, the best subset of observations is the one with the smallest local average root-likelihood ratio and largest local average log-likelihood ratio in the Bayesian and Neyman-Pearson setting, respectively. We supplement the proposed framework with a thorough analysis for Gaussian observations, including the case when the sensors are conditionally dependent, and also provide examples for other observation distributions. One of the results shows that, for nonidentical Gaussian sensor observations with uncommon means and common covariances under both hypotheses, the number of sensors required to achieve a desired detection performance reduces significantly as the sensors become more coherent.

Chapter 6 contrasts with the discrete sparse sensing mechanisms that have been considered in Chapter 3 to Chapter 5, where the sparse sensing functions are parameterized by a discrete sparse vector that needs to be optimally designed. This basically means that the continuous observation domain is first discretized into grid points and we have to select the best subset out of those grid points. However, this discretization might be very coarse because of complexity reasons, preventing the system to achieve the best possible compression rates for the considered inference task. Therefore, in this chapter, we introduce *continuous sparse sensing* (or off-the-grid sparse sensing), where it is possible to sample in between the grid points and take samples anywhere in the continuous observation domain. The basic idea is to start from a discretized sampling space and to model every sampling point in the continuous sampling space as a discrete sampling point plus a perturbation. Then, the smallest set of possible discrete sampling points is searched for, along with the best possible perturbations, in order to reach the desired inference performance. We will demonstrate this approach for linear inverse problems, that is, for linear estimation problems with additive Gaussian noise.

Chapter 7 addresses subquestions of Q2 related to distributed sampling. More specifically, this chapter is dedicated to *wireless clock synchronization*. To realize this, we assume an affine clock model, that is, we approximate the clock deviations using phase offset (or clock offset) and frequency offset (or clock skew), where we ignore the higher order terms like the frequency drift and so forth. In other words, we approximate the sample clock curves in Fig. 1.3 with a piecewise straight line (within each observation interval) having slope and offset equal to the clock skew and clock offset, respectively. The assumption is that there are several anchor nodes with known relative locations and one sensor node with an unknown position. Further, all the nodes have unreliable and uncommon clocks, except for one node that has a relatively stable clock (that is, one of the nodes has no clock offset and a clock skew equal to one). We estimate these clock parameters using time-of-flight measurements. To fully harness the broadcast nature of the wireless medium, we allow all the nodes to passively listen to the messages and record time stamps. By doing so, we collect a significant amount of extra measurements, which we solve using a least squares estimator. Specifically, we solve for all the unknown clock skews and clock offsets along with the pairwise distances (i.e., ranges) of the sensor to each anchor. The proposed estimator is shown to be efficient, asymptotically meeting the theoretical CRB.

The proposed framework can be extended to jointly resolve unknown clock parameters and locations (instead of ranges) — a pertinent problem to time-based sensor network localization; see **C8** for the related results and the CRB. These results are, however, excluded in this thesis for the sake of conciseness.

Chapter 8 provides a framework for joint *position and orientation estimation* of a rigid platform. We consider a setup in which a few sensors are mounted on a rigid body. The absolute position of the rigid body is not known. However, we know how the sensors are mounted on the rigid body, i.e., the sensor topology is known. The rigid body is localized using noisy distance measurements between the sensors and a few anchors (nodes with known absolute positions), and without using any inertial measurements. We model the rigid body localization problem using an unknown rotation matrix and a translation vector that uniquely determine the orientation and position of the rigid platform, respectively. We propose a least squares, and a number of constrained least squares estimators, where the constrained estimators solve an optimization problem on the Stiefel manifold. As a benchmark, we derive a unitarily constrained CRB. Finally, the known topology of the sensors can be perturbed during fabrication or if the body is not entirely rigid. To take these perturbations into account, constrained total least squares estimators are also proposed.

We can further track the position and orientation of the rigid body using a state-space representation and a (constrained) Kalman filter; see **C9** for details. The results on tracking are, however, excluded here for the sake of conciseness.

Chapter 9 contains the conclusions and outlines a number of directions for future research along with some open problems.

1.5 List of publications and other contributions

The research work done for this thesis has resulted in the following journal papers, conference papers, and internal reports.

Included publications

J1 **S.P. Chepuri** and G. Leus, "Sparsity-Promoting Sensor Selection for Non-linear Measurement Models", *IEEE Trans. on Signal Processing*, vol. 63, no. 3, pp. 684–698, Feb. 2015,

J2 **S.P. Chepuri** and G. Leus, "Sparse Sensing for Distributed Detection", *Submitted to IEEE Trans. on Signal Processing*, Mar. 2015,

J3 **S.P. Chepuri** and G. Leus, "Continuous Sensor Placement", *IEEE Signal Processing Letters*, vol. 22, no. 5, pp. 544–548, May 2015,

J4 **S.P. Chepuri**, R.T. Rajan, G. Leus, and A.-J. van der Veen,“ Joint Clock Synchronization and Ranging: Asymmetrical Time-Stamping and Passive Listening”, *IEEE Signal Processing Letters*, vol. 20, no. 1, pp 51 - 54, Jan. 2013.

J5 **S.P. Chepuri**, G. Leus, and A.-J. van der Veen,“ Rigid Body Localization Using Sensor Networks”, *IEEE Trans. on Signal Processing*, vol. 62, no. 18, pp 4911 - 4924, Sep. 2013.

and the following conference papers

C1 **S.P. Chepuri** and G. Leus, “Sensor Selection for Estimation, Filtering, and Detection”, in *Proc. of the International Conference on Signal Processing and Communications (SPCOM 2014)*, Bangalore, India, Jul. 2014.

C2 **S.P. Chepuri**, G. Leus, and A.-J. van der Veen, “Sparsity-Exploiting Anchor Placement for Localization in Sensor Networks”, in *Proc. of the European Signal Processing Conference (Eusipco 2013)*, Marrakech, Morroco, Sep. 2013.

C3 **S.P. Chepuri** and G. Leus. Sparse Sensing for Estimation with Correlated Observations. To appear in proc. of Asilomar Conf. Signals, systems, and Computers (Asilomar 2015), Pacific Grove, California, USA, Nov. 2015

C4 **S.P. Chepuri** and G. Leus, “Compression schemes for time-varying sparse signals”, in *Proc. of Asilomar Conf. Signals, systems, and Computers (Asilomar 2014)*, Pacific Grove, California, USA, Nov. 2014.

C5 **S.P. Chepuri** and G. Leus, “Sparsity-Promoting Adaptive Sensor Selection for Non-linear Filtering”, in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, Florence, Italy, May 2014.

C6 **S.P. Chepuri** and G. Leus, “Sparse Sensing for Distributed Gaussian Detection”, in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2015)*, Brisbane, Australia, Apr. 2015, (**Best student paper award**.)

C7 **S.P. Chepuri**, G. Leus, and A.-J. van der Veen, “Position and orientation estimation of a rigid body: rigid body localization”, in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, Vancouver, Canada, May 2013.

Other related contributions

Journal papers

J6 S. Khademi, **S.P. Chepuri**, Z. Irahhauten, G.J.M. Janssen, and A.-J. van der Veen, “Channel Measurements and Modeling for a 60 GHz Wireless Link Within a Metal Cabinet”, *IEEE Trans. on Wireless Communications*, vol. 99, pp.1536-1276, May 2015.

J7 G. Kail, **S.P. Chepuri**, and G. Leus. Robust Censoring Using Metropolis-Hastings Sampling. *Submitted to IEEE Journal of Sel. Topics in Sig. Proc.*, Apr. 2015.

J8 S. Liu, **S.P. Chepuri**, M. Fardad, E. Masazade, G. Leus, and P.K. Varshney. Sensor Selection for Estimation with Correlated Measurement Noise. *Submitted to IEEE Transactions on Sig. Proc.*, Aug. 2015.

J9 S. Maleki, **S.P. Chepuri**, and G. Leus. Optimization of Hard Fusion Based Spectrum Sensing for Energy-Constrained Cognitive Radio Networks. *Elsevier Physical Communication*, ISSN 1874-4907, 10.1016/j.phycom.2012.07.003, Jun. 2012.

Conference papers

C8 **S.P. Chepuri**, G. Leus, and A.-J. van der Veen, "Joint Localization and Clock Synchronization for Wireless Sensor Networks", in *Proc. of Asilomar Conference on Signals, systems, and Computers (Asilomar 2012)*, Pacific Grove, California, USA, Nov. 2012.

C9 **S.P. Chepuri**, A. Simonetto, G. Leus, and A.-J. van der Veen, "Tracking Position and Orientation of a Mobile Rigid Body", in *Proc. of the IEEE Workshop on Comp. Adv. in Multi-Sensor Adaptive Proc. (CAMSAP 2013)*, Saint Martin, Dutch Antilles, Dec. 2013.

C10 K. Hu, **S.P. Chepuri**, and G. Leus, "Near-Field Source Localization Using Sparse Recovery Techniques", in *Proc. of the International Conference on Signal Processing and Communications (SPCOM 2014)*, Bangalore, India, Jul. 2014.

C11 K. Hu, **S.P. Chepuri**, and G. Leus, "Near-Field Source Localization: Sparse Recovery Techniques and Grid Matching", in *Proc. of the Eighth IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2014)*, A Corua, Spain, Jun. 2014 (**Finalist best student paper award contest**.)

C12 S. Khademi, **S.P. Chepuri**, Z. Irahhauten, G.J.M. Janssen, and A.-J. van der Veen, "Channel Characterization for Wideband 60 GHz Wireless Link Within a Metal Enclosure", in *Proc. of the 8th European Conference on Antennas and Propagation (EuCAP 2014)*, The Hague, Netherlands, Apr. 2014.

C13 V. Roy, **S.P. Chepuri**, and G. Leus, " Sparsity-Enforcing Sensor Selection for DOA Estimation", in *Proc. of the IEEE Workshop on Comp. Adv. in Multi-Sensor Adaptive Proc. (CAMSAP 2013)*, Saint Martin, Dutch Antilles, Dec. 2013.

C14 S. Khademi, **S.P. Chepuri**, G. Leus, and A.-J. van der Veen, "Zero-forcing pre-equalization with transmit antenna selection in MIMO systems", in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, Vancouver, Canada, May 2013.

C15 G. Kail, **S.P. Chepuri**, and G. Leus. Robust Censoring For Linear Inverse Problems. In Proc. of *IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2015)*, Stockholm, Sweden, Jun. 2015.

C16 **S.P. Chepuri**, Y. Zhang, G. Leus, and G.B. Giannakis. Big Data Sketching with Model Mismatch. To appear in proc. of *Asilomar Conf. Signals, systems, and Computers (Asilomar 2015)*, Pacific Grove, California, USA, Nov. 2015.

C17 S. Rao, **S.P. Chepuri**, and G. Leus. Greedy Sensor Selection for Non-Linear Models.To appear in proc. of *IEEE Workshop on Comp. Adv. in Multi-Sensor Adaptive Proc. (CAMSAP 2015)*, Cancun, Mexico, Dec. 2015.

C18 S. Rao, **S.P. Chepuri**, and G. Leus. DOA Estimation Using Sparse Vector Sensor Arrays. To appear in proc. of *IEEE Workshop Workshop on Comp. Adv. in Multi-Sensor Adaptive Proc. (CAMSAP 2015)*, Cancun, Mexico, Dec. 2015.

Internal reports

R1 **S.P. Chepuri** and A.-J. van der Veen, Clock requirements and synchronization for the FAST-COM project, March 25, 2014, 12 pages. (available on request.)

Part I

Theory and Algorithms

Chapter 2

Sparse Sensing

Contents

2.1 Introduction

Sparse sensing extends traditional sensing methods to more structured sensing mechanisms. The sensing functions are designed knowing beforehand the inference task we want to perform on the data. The statistical inference task could be as general as *estimation*, *filtering*, or *detection*, which are fundamental to statistical signal processing. The main aim of sparse sensing is therefore to exploit the knowledge of the inference task to be performed on the acquired data in order to significantly reduce the costs of sensing as well as storage and communications requirements yet assuring a desired inference quality.

In this chapter we provide the fundamental theory of sparse sensing. This includes modeling as well as designing the sensing function that is used for gathering data (or sensing) to carry out a specific signal processing task. In particular, we discuss the continuous and discrete sparse sensing problems. Sections 2.3 and 2.4 present some applications and benefits of sparse sensing, respectively. Section 2.5 gives the major differences of sparse sensing with compressed sensing.

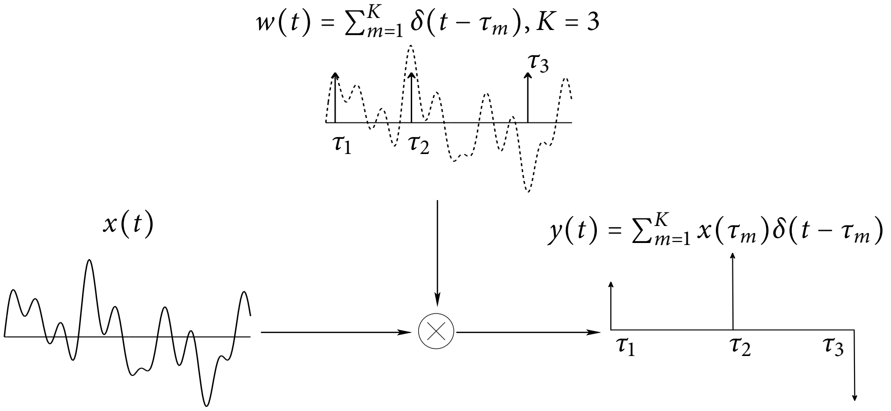

$w(t) = \sum_{m=1}^{K} \delta(t - \tau_m), K = 3$
τ_1
τ_2
τ_3
$x(t)$
$y(t) = \sum_{m=1}^{K} x(\tau_m)\delta(t - \tau_m)$
τ_1
τ_2
τ_3

Figure 2.1: Continuous sparse sensing scheme. Here, we show an illustration, where we acquire $K = 3$ samples from a one-dimensional signal $x(t)$.

2.2 Sparse sensing model

Let $x(t)$ be a continuous-domain signal, where $t \in [0, T]$ denotes the sampling domain. The sampling domain can be space, time, or space-time and can be even of higher dimensions. For example, t could be p-dimensional, where $p = 4$ represents a (three-dimensional) spatio-temporal sampling domain.

We assume that the observation signal $x(t)$ follows a known model, which relates the observation to the state of nature (e.g., through a parametric model or conditional distributions under different hypotheses) that we want to infer along with its statistical dependence on noise, if any.

In practice, the inference problem is typically solved in a digital fashion, that is, using a sampled version of the observation process. In this context, we are interested in the following fundamental question:

> *What are the best indices $\{\tau_m\}$ to optimally sample $x(t)$ to form $\{y_m = x(\tau_m)\}$ such that a desired inference performance is achieved?*

This is tantamount to applying a continuous-domain *sparse sensing* function (hence the name) $w(t)$, modeled as a sum of Diracs,

$$w(t) = \sum_{m=1}^{K} \delta(t - \tau_m) \tag{2.1}$$

that we apply on $x(t)$ to acquire

$$y(t) = w(t)x(t) = \sum_{m=1}^{K} x(\tau_m)\delta(t - \tau_m), \tag{2.2}$$

where we jointly design the *unknown* indices $\{\tau_m\}$, and the number of samples K, required in order to reach a desired inference performance. We label such a sensing mechanism as *continuous sparse*

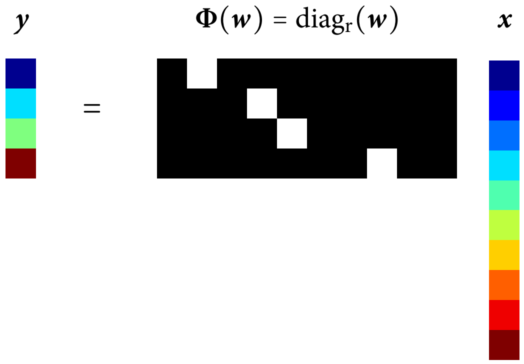


Figure 2.2: Discrete sparse sensing scheme. Here, a white (black) and colored square represents a one (zero) and an arbitrary value, respectively.

sensing; see the illustration in Fig. 2.1. The sensing operation $w(t)$ is designed keeping in mind the known inference task that needs to be performed, and it is related to it.

A way to design the continuous-domain sparse $w(t)$ is to discretize the sampling domain, and to assume that the indices $\{\tau_m\}_{m=1}^{K}$ lie on a discrete grid. In other words, we assume a set of $M \gg K$ candidate sampling locations $\{t_m\}_{m=1}^{M}$, and we alternatively model $y(t)$ as

$$y(t) = \sum_{m=1}^{M} w_m x(t_m)\delta(t - t_m), \tag{2.3}$$

where $w_m = (0)1$ indicates whether sample $x(t_m)$ is (not) selected. The vector $\boldsymbol{w}$ obtained by collecting $\{w_m\}_{m=1}^{M}$ as

$$\boldsymbol{w} = [w_1, w_2, \dots, w_M]^T \in \{0,1\}^M$$

is used to construct a sensing matrix $\boldsymbol{\Phi}(\boldsymbol{w}) = \text{diag}_\text{r}(\boldsymbol{w})$, which is then applied to the discrete signal $\boldsymbol{x} = [x(t_1), x(t_2), \dots, x(t_M)]^T$, to obtain a discrete-domain counterpart of (2.3) given as

$$\boldsymbol{y} = \boldsymbol{\Phi}(\boldsymbol{w})\boldsymbol{x} = \text{diag}_\text{r}(\boldsymbol{w})\boldsymbol{x}, \tag{2.4}$$

where $\boldsymbol{y} = [y(t_1), y(t_2), \dots, y(t_M)]^T$. We term such a sensing mechanism as *discrete sparse sensing*; see the illustration in Fig. 2.2. Thus, the design of a sparse function $w(t)$, can be simplified to the design of a sparse vector $\boldsymbol{w}$. Formally, we pose the question:

> *What is the sparsest* **w** *to optimally sense* **x** *to form* **y** *such that a desired inference performance is achieved?*

Before discussing the properties of the proposed sparse sensing scheme, we state the following remark.

Sparse sensing guided by a sparse vector has several interesting properties:

1. **Linearity**: The compression is linear, however the sampling is typically nonuniform or irregular (as in [Marziliano and Vetterli, 2000, Marvasti, 2001, Vaidyanathan, 2001], for instance).

2. **Deterministic and structured samplers**: Sparse sensing is close to traditional sampling. The samplers are easier to implement as compared to sampling via random projections, which is oftentimes not practical.

3. **Distributed sampling**: In contrast to a nonsparse linear compression, the construction of the sensing matrix $\mathbf{\Phi}(\boldsymbol{w})$ enables a fully distributed sampling scheme, which is central to distributed signal processing.

4. **Controllable**: Naturally, as with any subsampling scheme, sparse sensing also results in a reduction of the signal-to-noise ratio, by the compression factor, and leads to a loss in the inference performance. However, with sparse sensing, there exists a handle to trade this loss with the compression rate.

Other sub-optimal and trivial alternatives to the proposed sparse sensing mechanism are, for example,

1. **Uniform sensing:** A common practice is to use equally-spaced sampling indices, $t_m = -T/2 + (m-1)(T/K)$, $m = 1, 2, \ldots, K$.

2. **Random sensing:** Another approach would be to instead pick K indices uniformly at random, i.e., $t_m \sim \mathcal{U}[-T/2, T/2]$, $m = 1, 2, \ldots, K$.

These sensing schemes are suboptimal as they ignore the inference task at hand and might not always guarantee a desired inference performance or limits the compression rate.

2.3 Applications

The proposed sparse sensing model is universal in the sense that it can be used to optimally gather data or to optimally compress already acquired data to carry out a specific signal processing task. For example, the task could be estimation, learning (classification, clustering), control, or detection. This applies to a wide range of applications:

Sensor placement/selection/scheduling

Many real-world applications like field (temperature, pollution, precipitation, sound) inference, target localization and tracking, radar and sonar systems, video surveillance, imaging, spectral sensing, seismology, control, to list a few, are carried out using sensor networks —a spatial sampling modality. For such applications, sensor placement, sensor selection, and sensor scheduling are some of the important key design issues.

Choosing the best subset of sensors (or spatial locations) from a large set of candidate or available sensors (or spatial locations) such that a desired inference performance is achieved is referred to as *sensor selection (placement)*. Interpreting the entries of $\boldsymbol{x}$ as the observations from different sensors (or spatial locations), sensor selection/placement can be achieved through sparse sensing. Sparse sensing can also be used to select temporal samples or to schedule space-time sensor activations (e.g., antenna thinning). Sparse sensing can be used for *source placement*—a dual problem of sensor placement. Source placement can be interpreted as the problem of choosing the best subset of source (e.g., heater) locations from a large set of candidate source locations in order to generate a desired field (e.g., temperature).

Sparse sensing does not necessarily mean that the sensing matrix should itself be sparse. That is, sparse sensing can be used to pick the best subset of rows of matrix $\boldsymbol{A}$ that are applied to the signal $\boldsymbol{x}$. In other words, the signal $\boldsymbol{x}$ is acquired using a sensing function $\mathrm{diag}_{\mathrm{r}}(\boldsymbol{w})\boldsymbol{A}$. For example, the rows of $\boldsymbol{A}$ can represent different receive beamformers, filters, or (sparsifying) basis functions that are selected using a sparse $\boldsymbol{w}$.

Data/Graph sketching

The sparse sensing mechanisms developed in this thesis, find applications much beyond sensing a physical phenomenon. Examples include data from the internet, social networks, financial markets, genetics, consumer behavior, and so on. Data analytics (e.g., robust learning, classification, clustering) with such large-scale data is infeasible without dimensionality reduction due to limited computational capacity [Slavakis et al., 2014]. Sparse sensing guided by a sparse $\boldsymbol{w}$ can be used for dimensionality reduction of already acquired data by sketching (or throwing away) less informative samples again to ensure a desired inference quality. Data can also be represented as signals on graphs [Sandryhaila and Moura, 2014], where sparse sensing can be used to design optimal sampling of graph signals. In other words, which are the best graph nodes to observe for localizing/detecting a source (e.g., rumor, virus, failure) in a network (e.g., power grid, internet, social) admitting a graph representation.

2.4 Benefits

In this section, we will list the benefits of sparse sensing and answer the following questions: why and when is sparse sensing important?

1. **Economical constraints (hardware costs)**: In many of the practical applications such as environmental monitoring, radio astronomy, localization services, the sensing devices (including communications and signal processing hardware; their maintenance) are expensive. In such cases, it is of paramount importance to minimize the number of sensing devices itself yet achieving the best possible inference performance.

2. **Storage and physical space**: Sensors generate prohibitively large datasets. If the data is not acquired smartly keeping in mind the task to be performed, then there might not be sufficient memory to store the acquired data for subsequent processing. In some applications,

the physical space available for sensor placement might also be very restrictive and limited. For example, in thermal management of microprocessors there is not much physical space available for temperature sensor placement.

3. **Communications bandwidth**: Often the data acquired from the distributed sensors have to be transported to a central processing unit or a server. This consumes spectral resources, energy related to transmission and reception, and creates a need for a high data rate communication link. The communications requirements can be significantly reduced through sparse sensing.

4. **Processing and inference costs**: The data acquired has to be optimally processed to solve a specific interference task. Solving the inference problem becomes more and more difficult (e.g., increased latency, more computational capacity is required) as the data volume increases. Hence, through sparse sensing the processing requirements can be seriously diminished.

2.5 Contrast with compressive sensing

Sparse sensing differs from the broad research area of compressive sensing —state of the art in the field of sensing cost reduction [Donoho, 2006b, Candès and Wakin, 2008]. Compressive sensing is an elegant protocol for sensing and compressing data simultaneously. Although compressive sensing also aims at gathering fewer samples or measurements, there are a number of major differences with sparse sensing.

1. **Sparsity and signal processing task**: In compressive sensing, the signal of interest $x(t)$, is always considered sparse in some domain and the main goal is sparse signal reconstruction. On the other hand, for sparse sensing, the underlying signal does not necessarily have to be sparse and more general signal processing tasks can be considered. This can include sparse signal reconstruction, which we will discuss more in detail in Chapter 4, but is not limited to it.

2. **Samplers and compression**: The theory developed under the classical compressive sensing framework advocates random compression, which is essential to provide recovery algorithms, reconstruction guarantees, and performance analyses. Although random compression introduces robustness, it might limit the maximum amount of compression if a specific signal processing task needs to be carried out. Such random samplers are difficult to realize in practice, particularly for applications requiring spatial sampling such as source localization, field estimation, imaging, and cognitive radio sensing, to list a few. Sparse sensing, on the other hand, is a deterministic type of data compression, where the sparse vector $\boldsymbol{w}$ inside the sensing function gives a handle on the compression factor that can be used for optimally designing the sensing process.

3. **Inference quality**: The inference quality in compressive sensing, i.e., the reconstruction quality is generally characterized by a probabilistic measure on the space of random compression

	Compressed sensing	**Sparse sensing**
Sparse $x(t)$	needed	not needed
Signal processing task	sparse signal reconstruction	any statistical inference task
Samplers	random	structured and deterministic
Compression	robust, but not practical	practical and controllable

Table 2.1: Relation to compressive sensing.

matrices. This means that the sensing function has to be constantly changed to achieve a desired result. In contrast, in sparse sensing we use a fixed sensing function, which is designed based on the probabilistic nature of the noise. Thus, it is practically more meaningful.

These differences are summarized in Table 2.1.

2.6 Sparse sensing design

To design the optimal sensing operator (characterized by the vector $\boldsymbol{w}$), we need to know the model of the physical world, and the definition of the inference task that we are trying to solve from the acquired data. These models describe the uncertainty about the state of nature through a probability measure for the noise. This uncertainty or (task-specific) statistical risk denoted by a function $f : \{0,1\}^M \longmapsto \mathbb{R}$ quantifies the inference performance. In other words, by making observations $\boldsymbol{y}$ through a sensing operator $\boldsymbol{\Phi}(\boldsymbol{w})$, the uncertainty is reduced. Depending on the definition of the inference task, the statistical risk can either be the estimation error, prediction error, or detection probability, for example. Further, the risk can be ensemble, where the average is computed under the noise pdf or instantaneous leading to *model-driven* or *data-driven* sparse sensing, respectively. In model-driven sparse sensing, the sensing function is designed to guarantee an average inference performance (hence, it can be computed offline), whereas data-driven sparse sensing requires data to compute the sensing function (e.g., for sketching or censoring purposes). We will restrict ourselves to model-driven sparse sensing throughout this thesis.

We are interested in the design of the lowest-cost sensing structures that guarantee a desired inference performance. Generally, this corresponds to a sparsest $\boldsymbol{w}$ for a fixed statistical risk. Mathematically, it is a constrained cardinality minimization problem:

$$\begin{aligned} &\underset{\boldsymbol{w}\in\{0,1\}^M}{\arg\min} \quad \|\boldsymbol{w}\|_0 \\ &\text{s.to} \quad f(\boldsymbol{w}) \le \lambda, \end{aligned} \tag{P0}$$

where the threshold λ specifies the inference accuracy. Clearly, λ controls the sample size (and, hence the related sensing cost). Equivalently, the optimization problem in (P0) can also be formulated as

$$\begin{aligned} &\underset{\boldsymbol{w}\in\{0,1\}^M}{\arg\min}\ \ f(\boldsymbol{w}) \\ &\text{s.to} \quad \|\boldsymbol{w}\|_0 = K, \end{aligned} \tag{P1}$$

where K is the desired number of samples. The design problems (P0) and (P1) are equivalent in the sense that with some threshold say λ^*, K samples can be selected. The problem of the form (P0) might be appropriate for certain designs where the number K is not known, in which case λ should be known. In many applications, the number K might be known beforehand. This might happen, for example, when the sensors have already been purchased and we would want to use all of them. When K is *a priori* known, then the problem of the form (P1) is the obvious choice.

An optimal solution to (P0) and (P1), respectively, requires a combinatorial search over all the 2^M and $\binom{M}{K}$ possible combinations. This quickly becomes computationally intractable for modest values of M and K. For example, with $M = 100$ available potential sensors, there are in the order of 10^{30} possible choices whose direct enumeration is clearly impossible. This is essentially due to the Boolean constraint on the design variable $\boldsymbol{w}$. In addition, the cardinality function $\|\boldsymbol{w}\|_0$ is nonconvex in $\boldsymbol{w}$.

Depending on the shape of the statistical risk f, with respect to the selection variables, the above nonconvex Boolean optimization problem can be solved in the following two ways as discussed next.

2.6.1 Convex risk

The discrete combinatorial problems (P0) and (P1) can be approximately solved via convex optimization techniques assuming that there exists a risk $f(\cdot)$ that is a convex function of its argument. A convex function is formally defined as follows.

Definition 2.1 (Convex function). *Given a convex set $\mathcal{W}$, the function $f : \mathcal{W} \longmapsto \mathbb{R}$ is said to be convex, if it satisfies*

$$f(t\boldsymbol{w}_1 + (1-t)\boldsymbol{w}_2) \le f(t\boldsymbol{w}_1) + (1-t)f(\boldsymbol{w}_2)$$

$\forall \boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathcal{W}$ *and* $0 \le t \le 1$.

In order to solve (P0) and (P1) via convex optimization, we use standard convex relaxations: the discrete Boolean constraint $\boldsymbol{w} \in \{0,1\}^M$ is relaxed to a continuous set (also its convex hull) $0 \le w_m \le 1, m = 1,2,\ldots,M$, and the ℓ_0-(quasi) norm can be approximated with the ℓ_1-norm, its best convex approximation. By doing so, we can simplify the combinatorial problems (P0) and (P1) to the convex optimization problems

$$\begin{aligned} &\underset{\boldsymbol{w}}{\arg\min}\ \ \|\boldsymbol{w}\|_1 \\ &\text{s.to} \quad f(\boldsymbol{w}) \le \lambda, \\ &\qquad\ \ 0 \le w_m \le 1, m = 1,2,\ldots,M, \end{aligned} \tag{R0}$$

and

$$\begin{aligned} \arg\min_{\boldsymbol{w}} \quad & f(\boldsymbol{w}) \\ \text{s.to} \quad & \|\boldsymbol{w}\|_1 = K, \\ & 0 \leq w_m \leq 1, m = 1, 2, \ldots, M, \end{aligned} \tag{R1}$$

respectively. An approximate Boolean solution can then be recovered from the solution of the above convex optimization problem either by simple thresholding or randomized rounding. Alternatively, the ℓ_0-(quasi) norm can be approximated using the sum-of-logs $\sum_{m=1}^{M} \ln(w_m + \delta)$ with $\delta > 0$, which results in an iteratively weighted ℓ_1-norm optimization problem. Typically, log-based heuristics result in a sparser solution, and thus better approximate the ℓ_0-(quasi) norm.

Specific instances of relaxed problem (R1) has been proposed by [Joshi and Boyd, 2009] for estimation with linear Gaussian models and uncorrelated inputs. In this thesis, we provide a unifying theory and related algorithms for sparse sensing for more complicated (nonlinear) inference tasks. More specifically, we will seek statistical risk functions $f(\boldsymbol{w})$ that are *convex* functions on $\boldsymbol{w} \in [0,1]^M$ for fundamental statistical inference problems like estimation, filtering, and detection with uncorrelated (Gaussian and nonGaussian models) and correlated inputs.

2.6.2 Submodular risk

An alternative way to solve (P0) and (P1) is to look for a risk function that is submodular in nature. The notion of submodularity is based on the property of diminishing returns. That is, for example, adding an observation to a set $\mathcal{X}$ benefits less than or the same as adding the same observation to one of the subsets of $\mathcal{X}$. Mathematically, submodularity can be defined as follows.

Definition 2.2 (Submodular function). *Given two sets $\mathcal{X}$ and $\mathcal{Y}$ such that $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathcal{M}$ and $s \in \mathcal{M}\backslash\mathcal{Y}$, a function f is submodular if it satisfies*

$$f(\mathcal{X} \cup s) - f(\mathcal{X}) \geq f(\mathcal{Y} \cup s) - f(\mathcal{Y}).$$

Further, if the submodular function is monotone nondecreasing, i.e., $f(\mathcal{X}) \leq f(\mathcal{Y})$ for all $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathcal{M}$ and normalized (i.e., $f(\emptyset) = 0$), then a greedy maximization of such a function

Algorithm 2.1 Submodular sensing [Krause, 2008]

```
1. Require $\mathcal{X} = \emptyset, K$.
2. for $k = 1$ to $K$
3.         $s^* = \arg\max_{s \notin \mathcal{X}} f(\mathcal{X} \cup s)$
4.         $\mathcal{X} \leftarrow \mathcal{X} \cup s^*$
5. end
6. Return $\mathcal{X}$
```

is near-optimal and has a deterministic approximation factor of $(1 - 1/e)$, where e is the Euler's number [Nemhauser et al., 1978]. Hence, submodular risks are very useful to solve (discrete) combinatorial optimization problems using low-complexity greedy algorithms (see for e.g., [Krause, 2008, Ranieri et al., 2014]). Some examples of submodular functions that are used in sensing optimization for estimation and filtering problems are frame potential [Ranieri et al., 2014], mutual information [Krause, 2008], and entropy [Krause, 2008].

Let us define an index set $\mathcal{X}$ that is related to the sparse vector $\boldsymbol{w}$ as

$$\mathcal{X} = \{m \,|\, w_m = 1, m = 1, 2, \ldots, M\}.$$

Thus, the set $\mathcal{X}$ is analogous (and maps uniquely) to the sparse vector $\boldsymbol{w}$. Assuming that there exists a submodular (task-specific) risk $f(\cdot)$, we can solve the sparse sensing problem of the form (P1) using a greedy algorithm, which iteratively adds elements such that the uncertainty is reduced the most starting from an empty set; see the psuedo-code in Algorithm 2.1. Solving problems of the form (P0), i.e., minimizing the number of samples subject to a performance constraint is a straightforward adaptation of Algorithm 2.1, where the elements are added starting from an empty set until a desired performance is acheived.

Submodular sensing is only applicable for discrete sparse sensing and not for continuous sparse sensing. On the other hand, convex optimization techniques can be used to design discrete as well as continuous sparse sensing mechanisms. In this thesis, the focus will be on sparse sensing design with convex risks and we will not further discuss submodular sensing.

2.7 Discussion

In this chapter we have kept the definition of the signal processing task and hence the risk function abstract. We shall discuss more specifically different risk functions $f(\boldsymbol{w})$ for estimation, filtering, and detection in Chapter 3, Chapter 4, and Chapter 5, respectively. We will provide algorithms to solve the continuous sparse sensing problem in Chapter 6. The risk functions that we will provide quantify the ensemble inference performance. Thus, they are data-independent and are model-driven. As a result, the proposed sparse sensing mechanisms are more suitable for offline designs. Once the sensing functions are designed, solving the inference problem is not novel by itself and is based on classical signal processing tools.

Part of this chapter was published as: S.P. Chepuri and G. Leus. Sparsity-Promoting Sensor Selection for nonlinear Measurement Models. *IEEE Trans. on Signal Processing*, 63(3): 684-698, Feb. 2015 [Chepuri and Leus, 2015b].

Chapter 3

Sparse Sensing for Estimation

Contents

3.1 Introduction

Discrete sparse sensing mechanisms enable the design of sparse space-time samplers that guarantee a desired estimation accuracy. Such problems are encountered, for example, in sensor placement (or selection), where the best subset of sensor locations (or measurements) are to be selected from a large set of candidate sensor locations (or measurements) subject to a specific performance constraint. Sensor selection/placement is pertinent to various sensor network and data analysis applications like network monitoring (e.g., power grid, internet), location-aware services (e.g., target localization and tracking), field (e.g., heat, sound) estimation, and environmental (e.g., climate, precipitation) monitoring, to list a few. In such applications, we essentially solve an inverse problem, where the goal is to infer the parameters that describe the underlying physical phenomenon from a set of noisy measurements. These unknown parameters are related to the measurements through a model. In this chapter we are interested in designing sparse sensing mechanisms to gather only the most informative data being aware of the data model as well as the estimation task. In particular, the focus will be on nonlinear measurement models and developing risk functions that quantify the estimation accuracy.

3.1.1 Related prior works

A large volume of literature exists on sensor selection [Joshi and Boyd, 2009, and references therein]. The sensor selection problem is often formulated as an optimization problem based on some well-known performance measures from the optimal design of experiments [Ford et al., 1989, Pukelsheim, 1993]. For parameter estimation problems, the performance measures are related to the error covariance matrix denoted by $\boldsymbol{E} = \mathbb{E}\{(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})^T\}$, and they are optimized with respect to the selection variables. Here, $\boldsymbol{\theta}$ and $\widehat{\boldsymbol{\theta}}$ denote the unknown parameter and its estimate, respectively. Some of the popular choices for the performance measures are:

1. *A-optimality*: sum of eigenvalues of $\boldsymbol{E}$, i.e., $\text{tr}\{\boldsymbol{E}\}$.

2. *E-optimality*: maximum eigenvalue of $\boldsymbol{E}$, i.e., $\lambda_{\max}\{\boldsymbol{E}\}$.

3. *D-optimality*: determinant of $\boldsymbol{E}$, i.e., $\det\{\boldsymbol{E}(\boldsymbol{w})\}$.

All the above measures are equally reasonable, although neither of them completely characterizes the error covariance. There is no general answer to the question of how does one performance metric compare with the other.

Sensor selection for additive Gaussian linear models has been solved via convex relaxation techniques in [Joshi and Boyd, 2009], where the matrix $\boldsymbol{E}$ can be expressed in closed form (thus, can be optimized). However, this is not true in general (e.g., for nonlinear or nonGaussian measurement models). The solution from [Joshi and Boyd, 2009] has also been applied to sensor placement for power grid monitoring in [Kekatos et al., 2012].

Alternative greedy approaches exploiting the submodularity of the objective function [Krause et al., 2008b, Krause and Guestrin, 2007, Krause et al., 2008a, Shamaiah et al., 2010, Yao et al., 1993, Ranieri et al., 2014] are also proposed to solve the sensor selection for estimation (cf. Section 2.6.2 of Chapter 2).

Sensor selection for dynamical systems often referred to as sensor polling or scheduling, is studied in [Masazade et al., 2012, Carmi, 2010, Fu et al., 2012]. All the above literature (in general) deals with measurements that are related to additive Gaussian linear models. In [Kekatos and Giannakis, 2011], reliable sensor selection based on the actual measurements to identify the outliers is presented.

A different problem, yet related to sensor selection, is the problem of identifying source-informative sensors, which is studied in [Schizas, 2013].

3.1.2 Contributions

We consider general scenarios where the measurements of the unknown parameter follow a nonlinear model (unlike [Joshi and Boyd, 2009] for instance). nonlinear measurement models are frequently encountered in applications like source localization, field estimation, or phase retrieval, to list a few. The error covariance matrix for nonlinear models is not always available in closed form, and more importantly it depends on the unknown parameter. Our first contribution in the context of sensor selection is to leverage the additive property of the inverse Cramér-Rao bound (CRB) or the Fisher information matrix (FIM) for independent observations, and thus to express the performance requirement as a convex set. The CRB is a rigorous performance measure for optimality, and it generalizes very well for nonlinear measurement models (not necessarily in additive Gaussian noise). Although the first part of this chapter focuses on independent observations, we also extend the framework to the case of nonlinear measurements in additive *correlated* Gaussian noise. In order to design the sensing mechanism of interest, we do not need the actual measurements (i.e., data independent, but model driven), and hence, our framework is also well-suited for solving offline design problems.

The proposed sensor selection framework is generic and can be applied to any nonlinear estimation problem (linear being a special case). The selection problem is formulated as the design of a sparse selection vector, which is an ℓ_0-(quasi) norm nonconvex Boolean optimization problem. The nonconvex sensor selection problem is relaxed using standard convex relaxation techniques that can be efficiently solved in polynomial time.

A sparsity-enhancing concave surrogate for the ℓ_0-(quasi) norm is also proposed for sensor selection as an alternative to the traditional best convex relaxation. This is particularly advantageous when there are multiple (nearly) identical sensor measurements.

To cope with large-scale problems, we further present a projected subgradient algorithm. It is worth mentioning that the projected subgradient algorithm allows a very easy distributed implementation. In essence, we seek a sparse vector (i.e., a vector with many zeros and a few nonzero entries) that determines the sensing pattern. Sparse sensing leads to energy-efficient sampling schemes. We illustrate the sensor selection problem using examples of sensor placement for source localization.

3.2 Sensing nonlinear observations

We consider a generic nonlinear measurement model

$$x_m = h_m(\boldsymbol{\theta}, n_m), \; m = 1, 2, \ldots, M, \tag{3.1}$$

where x_m is the mth spatial or temporal sensor measurement, $\boldsymbol{\theta} \in \mathbb{R}^N$ is the unknown parameter, $\{n_m\}_{m=1,2,\ldots,M}$ describe the noise components, and the regressors $\{h_m\}_{m=1,2,\ldots,M}$ are (in general) nonlinear functionals. Let the vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_M]^T \in \mathbb{R}^M$ collect the measurements. The likelihood of the measurements $p(\boldsymbol{x}; \boldsymbol{\theta})$ is the probability density function (pdf) of $\boldsymbol{x}$ parameterized by the unknown vector $\boldsymbol{\theta}$.

We acquire the data $\boldsymbol{x}$ via the discrete sparse sensing mechanism that was introduced in Chapter 2. That is, we acquire data as

$$\boldsymbol{y} = \text{diag}_{\text{r}}(\boldsymbol{w})\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{w})\boldsymbol{x},$$

where $\boldsymbol{\Phi}(\boldsymbol{w}) = \text{diag}_{\text{r}}(\boldsymbol{w}) \in \{0,1\}^{K \times M}$ is the sensing matrix characterized by the selection vector

$$\boldsymbol{w} = [w_1, w_2, \ldots, w_M]^T \in \{0,1\}^M.$$

Here, the variable $w_m = (0)1$ indicates whether the mth sensor is (not) selected. Note that we are interested in cases where $K \ll M$ and K is not known. The reduced dimension data vector $\boldsymbol{y} \in \mathbb{R}^K$ is used instead of $\boldsymbol{x} \in \mathbb{R}^M$ to solve the estimation problem.

Our goal is now to select the best subset ($\geq N$) of the M available (or candidate) sensors, that is, to design the entries of $\boldsymbol{w}$ as sparse as possible, such that a certain accuracy on the estimate $\widehat{\boldsymbol{\theta}}$ is guaranteed. For nonlinear inverse problems, the risk functions $f(\boldsymbol{w})$ that quantify the estimation accuracy are discussed next.

3.3 $f(\boldsymbol{w})$ for estimation

For nonlinear estimation problems, the error covariance matrix does not admit a closed-form expression or their expressions might not be suitable for numerical optimization, in general. Therefore, we will discuss a simpler and weaker surrogate, which can be optimized instead of the error covariance matrix. More specifically, we will use the CRB as a substitute for the error covariance matrix, however, we will not restrict ourselves to any specific estimator. The motivation behind using the CRB is twofold:

1. The CRB is a measure for the (local) identifiability of the problem [Rothenberg, 1971]. More specifically, a nonsingular FIM implies (local) solvability and a unique estimate of $\boldsymbol{\theta}$, however, the converse is not necessarily true. The sensor selection problem presented in this chapter seeks a subset of sensors for which the FIM has full rank in some domain such that the solvability of the problem in that domain is always ensured.

2. Typically, the subset of selected sensors that yields a lower CRB also yields a lower MSE, and thus improves the performance of any practical system.

The CRB also has a very attractive mathematical structure resulting in a selection problem that can be efficiently solved using numerical optimization techniques. Before we formally introduce the CRB, we make the following assumption:

Assumption 3.1 (Regularity condition)**.** *The log-likelihood of the measurements satisfies the regularity condition, that is,* $\mathbb{E}\{\frac{\partial \ln p(y;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\} = \boldsymbol{0}$.

The regularity condition in general holds for observations that belong to the family of exponential pdfs, and it already includes a large number of distributions. Thus, the proposed sparse sensing framework is valid as long as the above assumption is true.

Under Assumption 3.1 —a well-known condition for the CRB to exist [Kay, 1993], the covariance of any unbiased estimate $\widehat{\boldsymbol{\theta}} \in \mathbb{R}^N$ of the unknown parameter satisfies the well-known inequality [Kay, 1993]

$$\mathbb{E}\{(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}})^T\} \geq \boldsymbol{C}(\boldsymbol{w}, \boldsymbol{\theta}) = \boldsymbol{F}^{-1}(\boldsymbol{w}, \boldsymbol{\theta}),$$

where the Fisher information matrix (FIM) is given by [Kay, 1993]

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{w}, \boldsymbol{\theta}) &= -\mathbb{E}\left\{ \frac{\partial^2}{\partial \boldsymbol{\theta}} \left(\frac{\ln p(\boldsymbol{y}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^T \right\} \\ &= \mathbb{E}\left\{ \left(\frac{\partial \ln p(\boldsymbol{y}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right) \left(\frac{\partial \ln p(\boldsymbol{y}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^T \right\} \in \mathbb{R}^{N \times N}, \end{aligned} \tag{3.2}$$

and $\boldsymbol{C}(\boldsymbol{w}, \boldsymbol{\theta})$ is the CRB matrix.

We now introduce another assumption.

Assumption 3.2 (Independent observations). *The observations $\{x_m\}_{m=1,2,\ldots,M}$ are a sequence of independent random variables.*

Under Assumption 3.2, the selection variable w_m modifies the log-likelihood of the selected measurements[1] as

$$\ln p(\boldsymbol{y}; \boldsymbol{\theta}) = \ln \prod_{m=1}^{M} p(x_m; \boldsymbol{\theta})^{w_m} = \sum_{m=1}^{M} w_m \ln p(x_m; \boldsymbol{\theta}). \tag{3.3}$$

Using (3.3) in (3.2), the FIM $\boldsymbol{F}(\boldsymbol{w}, \boldsymbol{\theta})$, can be explicitly expressed as a *linear function* of $\boldsymbol{w}$ as

$$\boldsymbol{F}(\boldsymbol{w}, \boldsymbol{\theta}) = \sum_{m=1}^{M} w_m \boldsymbol{F}_m(\boldsymbol{\theta}), \tag{3.4}$$

where

$$\begin{aligned} \boldsymbol{F}_m(\boldsymbol{\theta}) &= -\mathbb{E}\left\{ \frac{\partial^2}{\partial \boldsymbol{\theta}} \left(\frac{\ln p(x_m; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^T \right\} \\ &= \mathbb{E}\left\{ \left(\frac{\partial \ln p(x_m; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right) \left(\frac{\partial \ln p(x_m; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right)^T \right\}, \end{aligned} \tag{3.5}$$

is the $N \times N$ FIM of the mth (local) measurement. In other words, (3.4) means that every independent measurement contributes to the information measure and we use the Boolean selection

[1] The pdf of the selected measurements is of reduced dimension, i.e., it does not include the measurements that are set to zero.

parameter to choose the most informative sensors (or measurements). Note that the FIM for nonlinear models depends on the unknown vector $\boldsymbol{\theta}$.

One specific example that often occurs in practice is the case where the observations $x_m, m = 1, 2, \ldots, M$, are related through the following additive Gaussian nonlinear model given by

$$x_m = h_m(\boldsymbol{\theta}) + n_m, \; m = 1, 2, \ldots, M, \tag{3.6}$$

for which it is easy to verify that (3.5) simplifies to

$$\boldsymbol{F}_m(\boldsymbol{\theta}) = \frac{1}{\sigma_m^2}\left(\frac{\partial h_m(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial h_m(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^T.$$

Remark 3.1 (Additive Gaussian linear model)**.**
As a special case, when the measurement process is linear, we have $x_m = \boldsymbol{h}_m^T\boldsymbol{\theta} + n_m, m = 1, 2, \ldots, M$, *i.e.,* $h_m(\boldsymbol{\theta}, n_m) := \boldsymbol{h}_m^T\boldsymbol{\theta} + n_m$ *with* $\boldsymbol{h}_m \in \mathbb{R}^N$ *being the regressor. The computation of the FIM for a linear model is straightforward, and is given by*

$$\boldsymbol{F}(\boldsymbol{w}) = \sum_{m=1}^{M} \frac{w_m}{\sigma_m^2}\boldsymbol{h}_m\boldsymbol{h}_m^T.$$

The CRB for linear models in additive Gaussian noise is equal to the mean squared error (MSE), and more importantly it is independent of the unknown vector $\boldsymbol{\theta}$.

In what follows, we will develop several scalar risk functions $f(\boldsymbol{w})$ together with the corresponding accuracy threshold λ that can be used in (P0) to design the sparse sensing operator as discussed in Chapter 2.

We constrain the estimation error $\boldsymbol{\varepsilon} = \widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}$ to be within an origin-centered circle of radius R_e with a probability higher than P_e, i.e.,

$$\Pr(\|\boldsymbol{\varepsilon}\|_2 \le R_e) \ge P_e, \tag{3.7}$$

where the values of R_e and P_e define the accuracy required and are assumed to be known. A higher accuracy level is obtained by reducing R_e and/or increasing P_e. This metric is used in several occasions as an accuracy measure (e.g., see [Cover and Thomas, 2012, Gustafsson and Gunnarsson, 2005, Wang et al., 2009]). We next discuss two popular performance measures from the design of experiments that satisfy the above requirement.

Trace constraint

The risk function that satisfies the accuracy requirement in (3.7) is

$$f(\boldsymbol{w}) := \operatorname{tr}\{\boldsymbol{C}(\boldsymbol{w}, \boldsymbol{\theta})\} = \operatorname{tr}\left\{\left(\sum_{m=1}^{M} w_m\boldsymbol{F}_m(\boldsymbol{\theta})\right)^{-1}\right\}$$

with a sufficient condition (see Appendix 3.A)

$$\operatorname{tr}\left\{\left(\sum_{m=1}^{M} w_m \boldsymbol{F}_m(\boldsymbol{\theta})\right)^{-1}\right\} \leq \lambda_{\mathrm{tr}} = (1 - P_e)R_e^2. \tag{3.8}$$

This measure is related to the *A-optimality* or the average-variance criterion, which restricts the sum of the semi-axes of the confidence ellipsoid to λ_{tr}.

Minimum eigenvalue constraint

Another risk function that satisfies the accuracy requirement in (3.7) is

$$f(\boldsymbol{w}) := \lambda_{\max}\{\boldsymbol{C}(\boldsymbol{w}, \boldsymbol{\theta})\}$$

with a sufficient condition

$$\lambda_{\min}\{\boldsymbol{F}(\boldsymbol{w}, \boldsymbol{\theta})\} \geq \lambda_{\mathrm{eig}} = \frac{N}{R_e^2}\left(\frac{1}{1 - P_e}\right),$$

where λ_{eig} is derived in [Wang et al., 2009] (see also Appendix 3.A). This measure is related to the *E-optimality* or the worst-case error, which restricts the semi-major axis of the confidence ellipsoid to λ_{eig}. The inequality constraint $\lambda_{\min}\{\boldsymbol{F}(\boldsymbol{w}, \boldsymbol{\theta})\} \geq \lambda_{\mathrm{eig}}$ can be equivalently expressed as the following linear matrix inequality (LMI):

$$\sum_{m=1}^{M} w_m \boldsymbol{F}_m(\boldsymbol{\theta}) - \lambda_{\mathrm{eig}} \boldsymbol{I}_N \geq \boldsymbol{0}_N. \tag{3.9}$$

In other words, we put a lower bound on each eigenvalue of the matrix $\boldsymbol{F}$.

The above performance measures depend on the unknown parameter $\boldsymbol{\theta}$. In practice, the unknown parameter $\boldsymbol{\theta}$ has a physical meaning and takes values within a certain domain denoted by $\mathcal{U}$. For example, in the case of direction-of-arrival estimation, $\mathcal{U}$ is the sector where the source is expected or for target localization it is the surveillance area where the target resides. Since the FIM for nonlinear models depends on the unknown $\boldsymbol{\theta}$, we propose to constrain every point within the domain $\mathcal{U}$.

Remark 3.2. *The trace constraint, which can also be represented by LMIs, has a larger feasible set as compared to the minimum eigenvalue constraint. However, the resulting sensor selection problem is computationally less attractive compared to the minimum eigenvalue constraint (as we show later on in Section 3.5.5).*

For the aforementioned reason, we focus on the minimum eigenvalue (LMI) constraint from now on. However, either one of the two performance constraints can be used.

3.4 Problem statement

Having introduced the risk functions we can now formally state the problem.

Problem 3.1 (Discrete sparse sensing for estimation).
Given the likelihoods of the measurements, $p(x_m; \boldsymbol{\theta})$, $m = 1, 2, \ldots, M$, and a desired inference performance λ_{eig}, find a sparsest vector $\boldsymbol{w} \in \{0,1\}^M$ that selects the minimum number of most informative sensors satisfying the performance measure $\sum_{m=1}^{M} w_m \boldsymbol{F}_m(\boldsymbol{\theta}) - \lambda_{\text{eig}} \boldsymbol{I}_N \succeq \boldsymbol{0}_N, \ \forall \boldsymbol{\theta} \in \mathcal{U}$.

Mathematically, the discrete sparse sensing problem (P0) introduced in Chapter 2 for estimation task specializes to

$$\boldsymbol{w}^\star = \arg\min_{\boldsymbol{w}} \quad \|\boldsymbol{w}\|_0 \tag{3.10a}$$

$$\text{s.t.} \sum_{m=1}^{M} w_m \boldsymbol{F}_m(\boldsymbol{\theta}) - \lambda_{\text{eig}} \boldsymbol{I}_N \succeq \boldsymbol{0}_N, \forall \boldsymbol{\theta} \in \mathcal{U}, \tag{3.10b}$$

$$\boldsymbol{w} \in \{0,1\}^M. \tag{3.10c}$$

The threshold λ_{eig} imposes the accuracy requirement. The threshold λ_{eig} is also the sparsity-inducing parameter, where $\lambda_{\text{eig}} \to 0$ implies a sparser solution. Alternatively, the sensor selection problem can also be expressed as (P1) described in Chapter 2 when K is known.

Suppose the domain $\mathcal{U}$ consists of D points, obtained by gridding the entire domain (where the parameter is expected) at a certain resolution. The resulting multiple LMI constraints can then be stacked together as a single LMI constraint. Let us consider the domain $\mathcal{U} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_D\}$ with $|\mathcal{U}| = D$. The constraints in (3.10b) can then be equivalently expressed as a single LMI constraint written as $\sum_{m=1}^{M} w_m \boldsymbol{F}_m - \lambda_{\text{eig}} \boldsymbol{I}_{DN} \succeq \boldsymbol{0}_{DN}$, where

$$\boldsymbol{F}_m = \text{diag}(\boldsymbol{F}_m(\boldsymbol{\theta}_1), \boldsymbol{F}_m(\boldsymbol{\theta}_2), \cdots, \boldsymbol{F}_m(\boldsymbol{\theta}_D)) \in \mathbb{S}^{DN}$$

for $m = 1, 2, \ldots, M$. Note that the FIM after gridding is independent of $\boldsymbol{\theta}$. Henceforth, we denote this simply by $\boldsymbol{F}_m$ (not explicitly as a function of $\boldsymbol{\theta}$). The computational complexity of the resulting solvers depends on the number of grid points, which is due to the fact that we do not exactly know where the true parameter is located and because we are dealing with a nonlinear system model. We make the following remarks to indicate some scenarios where the computational burden due to gridding can be reduced.

Remark 3.3 (Worst-case constraints).
If for every $\boldsymbol{\theta} \in \mathcal{U}$, there exists some $\tilde{\boldsymbol{\theta}} \in \mathcal{U}_{\text{worst}} \subset \mathcal{U}$ such that

$$\lambda_{\min}\left\{\boldsymbol{F}(\boldsymbol{w}, \tilde{\boldsymbol{\theta}})\right\} \leq \lambda_{\min}\left\{\boldsymbol{F}(\boldsymbol{w}, \boldsymbol{\theta})\right\}, \ \forall \boldsymbol{w} \in \{0,1\}^M,$$

then it is sufficient to constrain the performance for only the worst-case set $\mathcal{U}_{\text{worst}}$ instead of $\mathcal{U}$. This property can be used as a guideline for gridding.

Remark 3.4 (Bayesian CRB constraint)**.**
In a Bayesian setting, when prior information of the unknown parameter $\boldsymbol{\theta}$ is available, this additional knowledge can be incorporated in the CRB. The related information matrix is often called the Bayesian information matrix (BIM) [Van Trees, 2004], and it is independent of the unknown parameter (hence, gridding is not needed). The BIM is given by

$$\boldsymbol{F}_{\mathrm{B}}(\boldsymbol{w}) = \boldsymbol{J}_{\mathrm{p}} + \mathbb{E}_{\boldsymbol{\theta}}\{\boldsymbol{F}(\boldsymbol{w},\boldsymbol{\theta})\},$$

where $\boldsymbol{J}_{\mathrm{p}}$ is the prior information matrix $\boldsymbol{J}_{\mathrm{p}} = -\mathbb{E}_{\boldsymbol{\theta}}\left\{\frac{\partial}{\partial\boldsymbol{\theta}}\left(\frac{\ln p(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right)^T\right\}$ with the (log) prior $\ln p(\boldsymbol{\theta})$, and the expectation $\mathbb{E}_{\boldsymbol{\theta}}\{\cdot\}$ is under the pdf $p(\boldsymbol{\theta})$. The LMI constraint in (3.9) *for the Bayesian setting will then be*

$$\boldsymbol{J}_{\mathrm{p}} + \sum_{m=1}^{M} w_m \mathbb{E}_{\boldsymbol{\theta}}\{\boldsymbol{F}_m(\boldsymbol{\theta})\} \succeq \lambda_{\mathrm{eig}}\boldsymbol{I}_N. \tag{3.11}$$

The prior information typically comes from the dynamics, previous measurements, or combining other available measurements.

In order to optimize the Bayesian CRB, we need to know the distribution of the unknown parameter.

3.5 Solvers

As discussed in Chapter 2, the optimization problem (3.10) that is of the form (P0) is nonconvex in $\boldsymbol{w}$. We next present a number of solvers based on the relaxed convex problem (R0) presented in Chapter 2, which can be solved efficiently in polynomial time.

3.5.1 Convex approximation based on ℓ_1-norm

A computationally tractable (suboptimal) solution is to use the traditional best convex surrogate for the ℓ_0-(quasi) norm, namely, the ℓ_1-norm. The ℓ_1-norm is known to represent an efficient heuristic for the ℓ_0-(quasi) norm optimization with convex constraints especially when the solution is sparse [Polyak et al., 2013]. Such relaxations are well-studied for problems with linear constraints in the context of compressed sensing (CS) and sparse signal recovery [Donoho, 2006a]. The nonconvex Boolean constraint in (3.10c) is further relaxed to the convex box constraint $[0,1]^M$. The constraint (3.10b) is convex on $\boldsymbol{w} \in [0,1]^M$.

The relaxed optimization problem is given as the following problem

$$\widehat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^M} \quad \|\boldsymbol{w}\|_1 \tag{3.12a}$$

$$\text{s.t.} \sum_{m=1}^{M} w_m\boldsymbol{F}_m - \lambda_{\mathrm{eig}}\boldsymbol{I}_{DN} \succeq \boldsymbol{0}_{DN}, \tag{3.12b}$$

$$0 \le w_m \le 1,\ m = 1,2,\ldots,M. \tag{3.12c}$$

Due to the positivity constraint, the objective function $\|\boldsymbol{w}\|_1$ will simply be an affine function $\mathbf{1}_M^T\boldsymbol{w}$. The optimization problem in (3.12) is a standard semidefinite programming (SDP) problem in the inequality form, which can be efficiently solved in polynomial time using interior-point methods [Boyd and Vandenberghe, 2004], for instance. An implementation of the interior-point method for solving SDP problems in the inequality form is typically based on Newton's method using an approximating barrier function. A brief description of the *projected Newton's method* is provided in Appendix 3.B, which we use to analyze the computational complexity of the relaxed convex problem (3.12).

Remark 3.5 (Complexity per iteration).
The computational cost involved during each iteration is as follows [Boyd and Vandenberghe, 2004, p. 619]. The matrices $\{\boldsymbol{F}_m\}_{m=1,2,\ldots,M}$ have a block-diagonal structure with D blocks. Forming the matrix $\boldsymbol{S} = \sum_{m=1}^{M} w_m\boldsymbol{F}_m - \lambda_{\text{eig}}\boldsymbol{I}_{DN}$ costs $\mathcal{O}(DMN^2)$ flops; computing $\boldsymbol{S}^{-1}\boldsymbol{F}_i\ \forall i$ via Cholesky factorization costs $\mathcal{O}(MDN^3)$ flops; the Hessian matrix is computed via the inner product of the matrices $\boldsymbol{S}^{-1}\boldsymbol{F}_i$ and $\boldsymbol{S}^{-1}\boldsymbol{F}_j$, which costs $\mathcal{O}(DM^2N^2)\ \forall i,j$. Finally, the Newton step is computed via Cholesky factorization costing $\mathcal{O}(DM^3)$ flops, and the projection costs $\mathcal{O}(M)$ flops. Assuming that $M \gg N$, the overall computational complexity per iteration of the projected Newton's algorithm is then $\mathcal{O}(DM^3)$.

Implementations of the interior-point methods are easily available in the form of well-known toolboxes like YALMIP [Lofberg, 2004], SeDuMi [Sturm, 1999], and CVX [Grant and Boyd, 2012].

3.5.2 Projected subgradient algorithm

The second-order Newton's method (cf. Appendix 3.B) is computationally intensive when the number of candidate sensors is very large ($M \gg 1000$, for example). To circumvent this problem, we propose a subgradient algorithm. The projected subgradient algorithm is a first-order method that is attractive for large-scale problems as each iteration is much cheaper to process.

The subgradient method is typically used for optimizations involving nondifferentiable functions [Boyd et al., 2003, Bertsekas, 1999]. The subgradient method is a generalization of the gradient method for nonsmooth and nondifferentiable functions, such as, the ℓ_1-norm and the minimum eigenvalue constraint functions. We next derive the projected subgradient algorithm.

The relaxed sensor selection problem in (3.12) can be equivalently expressed as

$$\underset{\boldsymbol{w}}{\arg\min} \quad \mathbf{1}_M^T\boldsymbol{w} \tag{3.13a}$$

$$\text{s.t. } f_{\text{eig}}(\boldsymbol{w}) \geq \lambda_{\text{eig}}, \tag{3.13b}$$

$$\boldsymbol{w} \in \mathcal{W}, \tag{3.13c}$$

where $f_{\text{eig}}(\boldsymbol{w}) := \lambda_{\min}\{\sum_{m=1}^{M} w_m\boldsymbol{F}_m\}$ is the constraint function in (**??**), and the set

$$\mathcal{W} = \{\boldsymbol{w} \in \mathbb{R}^M \mid 0 \leq w_m \leq 1, m = 1, 2, \ldots, M\}$$

denotes the box constraints in (3.12c).

The objective $\mathbf{1}_M^T\boldsymbol{w}$ is affine, so a subgradient of the objective is the all-one vector $\mathbf{1}_M$. Let $\boldsymbol{g}^k \in \partial f_{\text{eig}}(\boldsymbol{w}^k)$ denote a subgradient of the constraint function $f_{\text{eig}}(\boldsymbol{w})$ evaluated at $\boldsymbol{w} = \boldsymbol{w}^k$. Here, the set $\partial f_{\text{eig}}(\boldsymbol{w}^k)$ denotes the subdifferential of $f_{\text{eig}}(\boldsymbol{w})$ towards $\boldsymbol{w}$ evaluated at $\boldsymbol{w} = \boldsymbol{w}^k$. To compute $\boldsymbol{g}^k$, we express the constraint function $f_{\text{eig}}(\boldsymbol{w}^k)$ as

$$f_{\text{eig}}(\boldsymbol{w}^k) = \inf_{\|\boldsymbol{v}\|\le 1} \quad \boldsymbol{v}^T\left(\sum_{m=1}^{M} w_m^k \boldsymbol{F}_m\right)\boldsymbol{v}.$$

The computation of a subgradient is straightforward, and is given by

$$\boldsymbol{g}^k = [(\boldsymbol{v}_{\text{min}}^k)^T\boldsymbol{F}_1\boldsymbol{v}_{\text{min}}^k, \cdots, (\boldsymbol{v}_{\text{min}}^k)^T\boldsymbol{F}_m\boldsymbol{v}_{\text{min}}^k]^T \in \partial f_{\text{eig}}(\boldsymbol{w}^k), \tag{3.14}$$

where $\boldsymbol{v}_{\text{min}}^k$ is the eigenvector corresponding to the minimum eigenvalue $\lambda_{\text{min}}\{\sum_{m=1}^M w_m^k\boldsymbol{F}_m\}$. The minimum eigenvalue and the corresponding eigenvector can be computed using a low-complexity iterative algorithm, for example, the power method (see Appendix 3.C) or using the standard eigenvalue decomposition [Golub and Van Loan, 1996]. Let the projection of a point onto the set $\mathcal{W}$ be denoted by the operator $\mathcal{P}_{\mathcal{W}}(\cdot)$, which can be expressed elementwise as

$$[\mathcal{P}_{\mathcal{W}}(\boldsymbol{w})]_m = \begin{cases} 0 & \text{if } w_m \le 0, \\ w_m & \text{if } 0 < w_m < 1, \\ 1 & \text{if } w_m \ge 1. \end{cases} \tag{3.15}$$

The projected subgradient algorithm then proceeds as follows:

$$\boldsymbol{w}^{k+1} = \begin{cases} \mathcal{P}_{\mathcal{W}}(\boldsymbol{w}^k - \alpha^k\mathbf{1}_M) & \text{if } f_{\text{eig}}(\boldsymbol{w}^k) \ge \lambda_{\text{eig}}, \\ \mathcal{P}_{\mathcal{W}}(\boldsymbol{w}^k + \alpha^k\boldsymbol{g}^k) & \text{if } f_{\text{eig}}(\boldsymbol{w}^k) < \lambda_{\text{eig}}. \end{cases} \tag{3.16}$$

In other words, if the current iterate $\boldsymbol{w}^k$ is feasible (i.e., $f_{\text{eig}}(\boldsymbol{w}^k) \ge \lambda_{\text{eig}}$), we update $\boldsymbol{w}$ in the direction of a negative objective subgradient, as if the LMI constraints were absent; if the current iterate $\boldsymbol{w}^k$ is infeasible (i.e., $f_{\text{eig}}(\boldsymbol{w}^k) < \lambda_{\text{eig}}$), we update $\boldsymbol{w}$ in the direction of a subgradient $\boldsymbol{g}^k$ associated with the LMI constraints. After the update is computed, the iterate is projected onto the constraint set $\mathcal{W}$ using $\mathcal{P}_{\mathcal{W}}(\cdot)$. When the kth iterate is feasible, a diminishing nonsummable step size $\alpha^k = 1/\sqrt{k}$ is used. When the iterate is not feasible Polyak's step size $\alpha^k = (f_{\text{eig}}(\boldsymbol{w}^k) + \alpha_0)/\|\boldsymbol{g}^k\|_2^2$ is used, where we adopt the optimal value for $\alpha_0 := \mathbf{1}_M^T\boldsymbol{w}^\star$ when $\|\boldsymbol{w}\|_0$ is known (i.e., the number of sensors to be selected is known). If this is not known, then we approximate it with $\alpha_0 := f_{\text{best}}^k + \gamma$, where $\gamma = 10/(10+k)$, and $f_{\text{best}}^k = \min\{f_{\text{best}}^{k-1}, \mathbf{1}_M^T\boldsymbol{w}^k\}$ [Boyd et al., 2003]. The algorithm is terminated after a specified maximum number of iterations k_{max}. Finally, the estimate is denoted by $\widehat{\boldsymbol{w}} = \boldsymbol{w}^{k_{\text{max}}}$.

The convergence results of the subgradient method for the constrained optimization (i.e., without the projection step) are derived in [Boyd et al., 2003]. Since the projection onto a convex set is nonexpansive [Bertsekas, 1999], it does not affect the convergence. The projected subgradient algorithm is summarized as Algorithm 3.1.

Algorithm 3.1 Projected subgradient algorithm

```
**Initialize** iteration counter $k = 0$, $\boldsymbol{w}^k = \mathbf{1}_M$, $\boldsymbol{g}^k = \mathbf{0}$, $k_{\max}$, $\alpha_0$, and $\lambda_{\text{eig}}$.
**for** $k = 0$ to $k_{\max}$
    **compute** $f_{\text{eig}}(\boldsymbol{w}^k) = \lambda_{\min}\{\sum_{m=1}^{M} w_m^k \boldsymbol{F}_m\}$
    **update**
        **if** $f_{\text{eig}}(\boldsymbol{w}^k) \geq \lambda_{\text{eig}}$
            $\boldsymbol{w}^{k+1} = \mathcal{P}_{\mathcal{W}}(\boldsymbol{w}^k - (1/\sqrt{k})\mathbf{1}_M)$
        **elseif** $f_{\text{eig}}(\boldsymbol{w}^k) < \lambda_{\text{eig}}$
            compute $\boldsymbol{g}^k$ according to (3.14)
            $\boldsymbol{w}^{k+1} = \mathcal{P}_{\mathcal{W}}(\boldsymbol{w}^k + \frac{f_{\text{eig}}(\boldsymbol{w}^k)+\alpha_0}{\|\boldsymbol{g}^k\|_2^2}\boldsymbol{g}^k)$
        **end**
**end**
$\widehat{\boldsymbol{w}} = \boldsymbol{w}^{k_{\max}}$
```

Remark 3.6 (Complexity per iteration)**.**
We first form the matrix $\sum_{m=1}^{M} w_m \boldsymbol{F}_m$, which costs $\mathcal{O}(DMN^2)$ flops. The minimum eigenvalue and the corresponding eigenvector can be computed using the power method at a cost of $\mathcal{O}(DN^2)$ flops [Golub and Van Loan, 1996]. Forming the vector $\boldsymbol{g}$ *costs $\mathcal{O}(DMN^2)$ flops, computing its norm costs $\mathcal{O}(M)$ flops, and the update and projection together cost $\mathcal{O}(M)$ flops. Assuming that $M \gg N$ as earlier, the computational cost of the projected subgradient algorithm is $\mathcal{O}(DMN^2)$, which is much lower than the complexity of the projected Newton's method.*

A distributed implementation of the projected subgradient algorithm is very easy. A simple distributed averaging algorithm (e.g., [Xiao and Boyd, 2004]) can be used to compute the sum of matrices $\sum_{m=1}^{M} w_m \boldsymbol{F}_m$. The minimum eigenvalue and the corresponding eigenvector can then be computed using power iterations at each node independently. The update step (3.16), the subgradient vector $\boldsymbol{g}$, and the projection are computed coordinatewise and are already distributed.

Subgradient methods are typically very slow compared to the interior-point method involving Newton iterations, and subgradient methods typically require a few hundred iterations. Newton's method typically requires in the order of ten steps. On the other hand, unlike the projected subgradient method, Newton's method cannot be easily distributed, and incurs a relatively high complexity per iteration due to the computation and storage of up to second-order derivatives. Depending on the scale of the problem and the resources available for processing one could choose between the subgradient or Newton's algorithm.

3.5.3 Concave surrogate based on sum-of-logarithms

The ℓ_1-norm is customarily used as the best convex relaxation for the ℓ_0-(quasi) norm. However, the intersection of the ℓ_1-norm ball (or an affine subspace) with the positive semi-definite cone (i.e., the

LMI constraint) is not always a unique point as shown in the following Theorem.

Theorem 3.1 (Uniqueness). *The projection of a point $\boldsymbol{w} \in [0,1]^M$ onto a convex LMI constraint set $\sum_{m=1}^{M} w_m \boldsymbol{F}_m - \lambda_{\text{eig}} \boldsymbol{I}_{DN} \succeq \boldsymbol{0}_{DN}$ under the ℓ_1-norm is not always unique.*

Proof. The proof follows from the fact that the ℓ_1-norm is not strictly convex, and from the linearity of the constraint set. Let us consider an example with $M = 2$ (w.l.o.g.), and $\boldsymbol{F}_1 = \boldsymbol{F}_2 \succeq \lambda_{\text{eig}} \boldsymbol{I}_{DN}$. In other words, the observations are identical. In this case, the extreme points of the ℓ_1-norm ball, i.e., $\widehat{\boldsymbol{w}}_1 = [1,0]^T$ and $\widehat{\boldsymbol{w}}_2 = [0,1]^T$ are two example solutions. Moreover, since the solution set of a convex minimization problem is convex, $\tau\widehat{\boldsymbol{w}}_1 + (1-\tau)\widehat{\boldsymbol{w}}_2$ is also a solution for *any* $0 < \tau < 1$, which gives an infinite number of solutions to the relaxed optimization problem (3.12). For such cases, the ℓ_1-norm relaxation might not result in a sparse solution. □

To improve upon the ℓ_1-norm solution due to its nonuniqueness following from Theorem 3.1, we propose an alternative approximation to the ℓ_0-(quasi) norm which also results in fewer selected sensors. Instead of relaxing the ℓ_0-(quasi) norm with the ℓ_1-norm, using a nonconvex surrogate function can yield a better approximation. It is motivated in [Candès et al., 2008] that the logarithm of the geometric mean of its elements can be used as an alternative surrogate function for linear inverse problems in CS. Adapting this to our sensor selection problem, we arrive at the optimization problem

$$\underset{\boldsymbol{w} \in \mathbb{R}^M}{\arg\min} \quad \sum_{m=1}^{M} \ln(w_m + \delta) \tag{3.17a}$$

$$\text{s.t.} \sum_{m=1}^{M} w_m \boldsymbol{F}_m - \lambda_{\text{eig}} \boldsymbol{I}_{DN} \succeq \boldsymbol{0}_{DN}, \tag{3.17b}$$

$$0 \le w_m \le 1, \; m = 1, 2, \ldots, M, \tag{3.17c}$$

where $\delta > 0$ is a small constant that prevents the cost from tending to $-\infty$. The cost (3.17a) is concave, but since it is smooth w.r.t. $\boldsymbol{w}$, iterative linearization can be performed to obtain a local minimum [Candès et al., 2008]. The first-order approximation of $\ln(w_m + \delta)$ around $w_m[i-1] + \delta$ results in

$$\ln(w_m + \delta) \le \ln(w_m[i-1] + \delta) + \frac{w_m - w_m[i-1]}{w_m[i-1] + \delta}.$$

Here, i denotes the iteration index. Instead of minimizing the original cost, the majorizing cost (second term on the right-hand side of the above inequality) can be optimized to attain a local minimum. More specifically, the optimization problem (3.17) can be iteratively driven to a local minimum using the iterations

$$\begin{aligned} \widehat{\boldsymbol{w}}[i] = \underset{\boldsymbol{w} \in \mathbb{R}^M}{\arg\min} & \quad \sum_{m=1}^{M} \frac{w_m}{\widehat{w}_m[i-1] + \delta} && (3.19a) \\ \text{s.t.} & \sum_{m=1}^{M} w_m \boldsymbol{F}_m - \lambda_{\text{eig}} \boldsymbol{I}_{DN} \geq \boldsymbol{0}_{DN}, && (3.19b) \\ & 0 \leq w_m \leq 1, \; m = 1, 2, \dots, M. && (3.19c) \end{aligned}$$

The iterative algorithm is summarized as Algorithm 3.2. Each iteration in (3.19) solves a weighted ℓ_1-norm optimization problem. The weight updates force the small entries of the vector $\widehat{\boldsymbol{w}}[i]$ to zero and avoid inappropriate suppression of larger entries. The parameter δ provides stability, and guarantees that the zero-valued entries of $\widehat{\boldsymbol{w}}[i]$ do not strictly prohibit a nonzero estimate at the next step. Finally, the estimate is given by $\widehat{\boldsymbol{w}} = \widehat{\boldsymbol{w}}[i_{\max}]$, where $i_{\max}$ is the specified maximum number of iterations.

Remark 3.7 (Sparsity-enhancing projected subgradient algorithm)**.**
The projected subgradient algorithm can be adapted to fit into the sparsity-enhancing iterative algorithm as well. The optimization problem (3.18) *is then replaced with the following update equations:*

$$\boldsymbol{w}^{k+1}[i] = \begin{cases} \mathcal{P}_{\mathcal{W}}(\boldsymbol{w}^k[i] - \alpha^k \boldsymbol{u}[i]) & \textit{if } f_{\text{eig}}(\boldsymbol{w}^k[i]) \geq \lambda_{\text{eig}}, \\ \mathcal{P}_{\mathcal{W}}(\boldsymbol{w}^k[i] + \alpha^k \boldsymbol{g}^k[i]) & \textit{if } f_{\text{eig}}(\boldsymbol{w}^k[i]) < \lambda_{\text{eig}}, \end{cases}$$

where we solve a number of iterations (inner loop) of the projected subgradient algorithm within the

Algorithm 3.2 Sparsity-enhancing iterative algorithm

```
**Initialize** the iteration counter $i = 0$, the weight vector $\boldsymbol{u}[0] = [u_1[0], u_2[0], \dots, u_M[0]]^T = \boldsymbol{1}_M$, $\delta$, and $i_{\max}$.
**for** $i = 0$ to $i_{\max}$
    **solve**

    $\widehat{\boldsymbol{w}}[i] = \underset{\boldsymbol{w} \in \mathbb{R}^M}{\arg\min} \quad \boldsymbol{u}[i]^T \boldsymbol{w}$    (3.18a)
    $\text{s.t.} \sum_{m=1}^{M} w_m \boldsymbol{F}_m - \lambda_{\text{eig}} \boldsymbol{I}_{DN} \geq \boldsymbol{0}_{DN},$    (3.18b)
    $0 \leq w_m \leq 1, m = 1, 2, \dots, M.$    (3.18c)

    **update** $u_m[i+1] = (\delta + \widehat{w}_m[i])^{-1}$, for each $m = 1, 2, \dots, M$.
**end**
$\widehat{\boldsymbol{w}} = \widehat{\boldsymbol{w}}[i_{\max}]$.
```

Algorithm 3.3 Randomized rounding algorithm

```
1. Generate $L$ candidate estimates of the form $w_{m,l} = 1$ $(l = 1, 2, \dots, L)$ with a
   probability $\widehat{w}_m$ (or $w_{m,l} = 0$ with a probability $1 - \widehat{w}_m$) for $m = 1, 2, \dots, M$.
2. Define $\boldsymbol{w}_l = [w_{1,l}, \dots, w_{M,l}]^T$ and the index set of the candidate estimates
   satisfying the constraints as
       $\Omega \triangleq \{l \mid \lambda_{\min}\{\boldsymbol{F}(\boldsymbol{w}_l, \boldsymbol{\theta})\} \geq \lambda_{\text{eig}}, \forall \boldsymbol{\theta} \in \mathcal{U}, \; l = 1, 2, \dots, L\}.$
3. If the set $\Omega$ is empty, go back to step 1.
4. The suboptimal Boolean estimate is the solution to the optimization problem
       $\widehat{\boldsymbol{w}}_{\text{bp}} = \arg\min_{l \in \Omega} \|\boldsymbol{w}_l\|_1.$
```

ith iteration (outer loop) of Algorithm 3.2. Here, the kth iterate of the inner loop in the ith outer loop is denoted as $(\cdot)^k[i]$.

From the solution of the relaxed optimization problem, the approximate Boolean solution can be obtained using randomization techniques, as described next.

3.5.4 Randomized rounding

The solution of the relaxed optimization problem is used to compute the suboptimal Boolean solution for the selection problem. A straightforward technique that is often used is the simple rounding technique, in which the Boolean estimate is given by $\text{round}(\widehat{w}_m)$, $m = 1, 2, \dots, M$, where we define $\widehat{\boldsymbol{w}} = [\widehat{w}_1, \widehat{w}_2, \dots, \widehat{w}_M]^T$, and the $\text{round}(\cdot)$ operator rounds its arguments towards the nearest integer. However, there is no guarantee that the Boolean estimates obtained from the rounding technique always satisfy the performance constraints. Hence, we propose a randomized rounding technique, where the suboptimal Boolean estimates are computed based on random experiments guided by the solution from the SDP problem (3.12) or the iterative version in (3.19). The randomized rounding technique is summarized as Algorithm 3.3.

3.5.5 Trace and determinant constraints

In this section, we will discuss the relaxed sensor selection problem based on the optimization criteria related to A-optimality and D-optimality.

Trace constraint

The relaxed sensor selection problem with the scalar trace constraint is given as follows

$$
\begin{aligned}
&\underset{\boldsymbol{w}\in\mathbb{R}^M}{\arg\min} \quad \|\boldsymbol{w}\|_1 \\
&\text{s.t. } \operatorname{tr}\left\{\left(\sum_{m=1}^{M} w_m \boldsymbol{F}_m(\boldsymbol{\theta})\right)^{-1}\right\} \le \lambda_{\text{tr}}, \quad \forall\boldsymbol{\theta}\in\mathcal{U}, \\
&0 \le w_m \le 1,\ m = 1,2,\dots,M.
\end{aligned}
\tag{3.20}
$$

The trace constraint in (3.20) is convex on $\boldsymbol{w}\in[0,1]^M$; this is easier to verify when the above trace constraint is expressed as an LMI [Boyd and Vandenberghe, 2004, p. 387]. The optimization problem (3.20) is a convex problem, and can be cast as an SDP:

$$
\begin{aligned}
&\underset{\boldsymbol{w}\in\mathbb{R}^M,\,\boldsymbol{x}\in\mathbb{R}^N}{\arg\min} \quad \|\boldsymbol{w}\|_1 \\
&\text{s.t. } \begin{bmatrix} \sum_{m=1}^{M} w_m \boldsymbol{F}_m(\boldsymbol{\theta}) & \boldsymbol{e}_n \\ \boldsymbol{e}_n^T & x_n \end{bmatrix} \succeq \boldsymbol{0}_{N+1}, n = 1,2,\dots,N,\ \forall\boldsymbol{\theta}\in\mathcal{U}, \\
&\boldsymbol{1}_N^T\boldsymbol{x} \le \lambda_{\text{tr}}, \\
&x_n \ge 0, n = 1,2,\dots,N, \\
&0 \le w_m \le 1,\ m = 1,2,\dots,M,
\end{aligned}
\tag{3.21}
$$

where $\boldsymbol{x} = [x_1, x_2, \dots, x_N]^T \in \mathbb{R}^N$ is the auxiliary variable and $\boldsymbol{e}_n$ is the nth unit vector in $\mathbb{R}^N$.

In addition to the box constraints, the optimization problem (3.21) has N LMI constraints (of size $N+1$) for every point in $\mathcal{U}$ and $N+1$ inequality constraints, while (3.12) has only one LMI constraint (of size N) for every point in $\mathcal{U}$. Hence, solving (3.21) is computationally more intense than solving (3.12).

Determinant constraint

Another popular scalar performance measure that quantifies the estimation accuracy is the determinant (product of eigenvalues) constraint. This measure is related to the D-optimality. The relaxed sensor selection problem with the determinant constraint is given as follows

$$
\begin{aligned}
&\underset{\boldsymbol{w}\in\mathbb{R}^M}{\arg\min} \quad \|\boldsymbol{w}\|_1 \\
&\text{s.t. } \ln\det\{\sum_{m=1}^{M} w_m \boldsymbol{F}_m(\boldsymbol{\theta})\} \ge \lambda_{\text{det}}, \forall\boldsymbol{\theta}\in\mathcal{U}, \\
&0 \le w_m \le 1,\ m = 1,2,\dots,M,
\end{aligned}
\tag{3.22}
$$

where the threshold λ_{det} specifies the desired mean radius of the confidence ellipsoid (see Appendix 3.A). The log-determinant constraint is a concave function of $\boldsymbol{w}\in[0,1]^M$. Note that although the constraint $\ln\det\{\sum_{m=1}^{M} w_m \boldsymbol{F}_m(\boldsymbol{\theta})\} \geq \lambda_{\text{det}}$ is an indication of the performance of the estimator, it is not a sufficient condition for (3.7).

The relaxed sensor selection problem with the scalar (trace or determinant) constraints can be solved with either one of the two proposed approximations of the cardinality cost, i.e., the ℓ_1-norm or log-based concave surrogate.

3.6 Numerical example: sensor placement

Localization is an important and extensively studied topic in wireless sensor networks (WSNs). Target localization can be performed using a plethora of algorithms [Gustafsson and Gunnarsson, 2005, Gezici et al., 2005, Patwari et al., 2005] (and references therein), which exploit inter-sensor measurements like time-of-arrival (TOA), time-difference-of-arrival (TDOA), angle-of-arrival (AOA), or received signal strength (RSS). The performance of any location estimator depends not only on the algorithm but also on the placement of the anchors (sensors with known locations). Sensor placement is a key challenge in localization system design, as certain sensor constellations not only deteriorate the performance, but also result in ambiguity or identifiability issues [Chepuri et al., 2013b].

The sensor placement problem can be interpreted as the problem where we divide a specific sensor area $\mathcal{S}$ into M grid points and select the best subset of locations from these grid points. Here, the selected sensors are deemed the best, if they guarantee a certain prescribed accuracy on the location estimates for a target within a specific target area $\mathcal{U}$. We consider a two-dimensional network with one target located in the target area $\mathcal{U}$ and M possible sensors located at the M grid points.

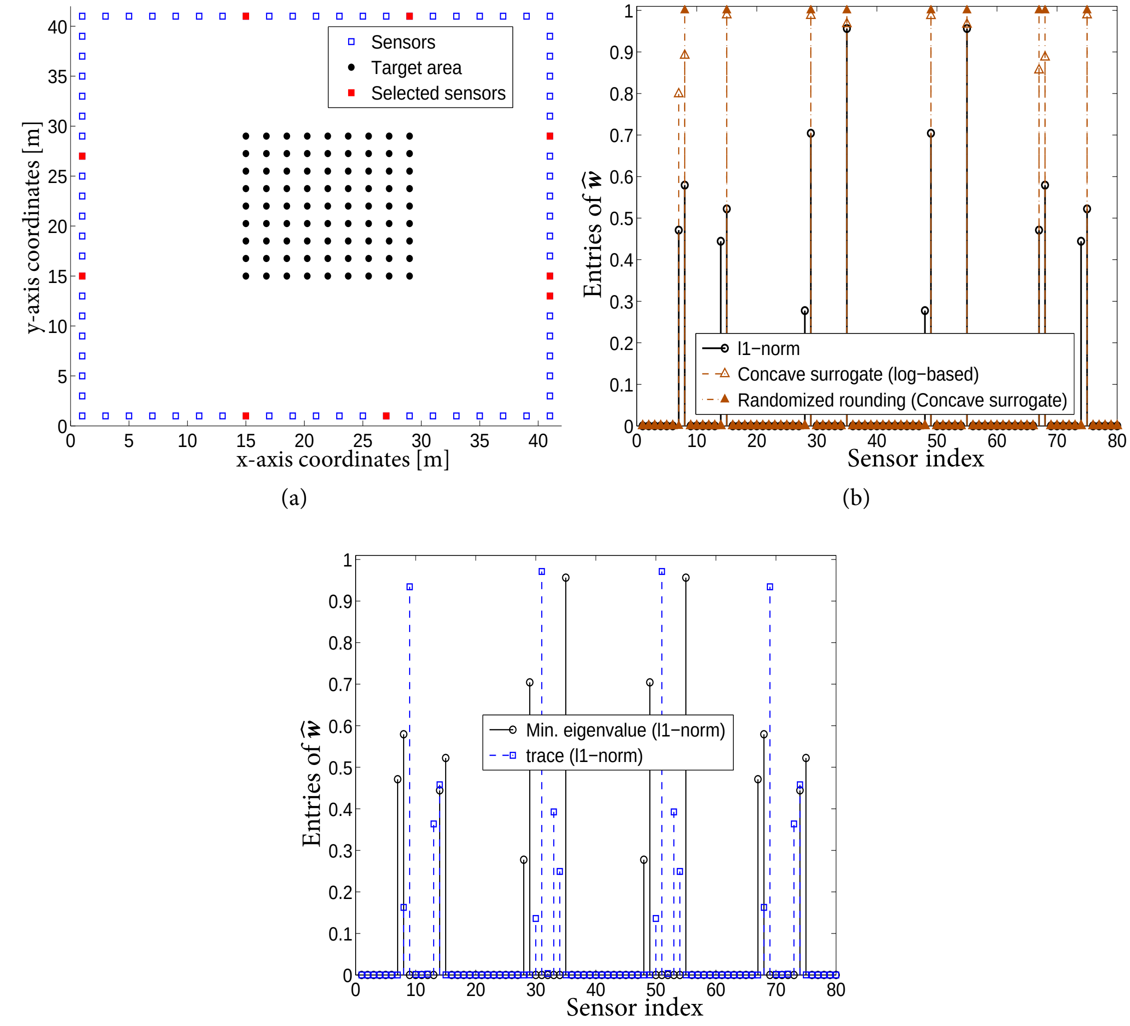
The absolute positions of the sensor grid points are known, hence, the considered sensors are commonly referred to as anchor nodes. Let the coordinates of the target and the mth anchor be denoted by the 2×1 vectors $\boldsymbol{\theta} = [\theta_1, \theta_2]^T$ and $\boldsymbol{a}_m = [a_{m,1}, a_{m,2}]^T$, respectively. Here, $\boldsymbol{\theta}$ is assumed to be unknown, but known to be within $\mathcal{U}$. We next illustrate the developed theory with an example.

Let the pairwise distance between the target and the mth anchor be denoted by $d_m = \|\boldsymbol{\theta} - \boldsymbol{a}_m\|_2$. In practice, the pairwise distances are obtained by ranging and they are noisy. The range measurements typically follow an additive Gaussian nonlinear model, as given by

$$y_m = d_m + n_m, \; m = 1, 2, \ldots, M, \tag{3.23}$$

where $n_m \sim \mathcal{N}(0, \sigma_m^2)$ is the noise with $\sigma_m^2 = \sigma^2 / d_m^{-\eta}$. Here, σ^2 is the nominal noise variance, and η is the path-loss exponent. We can now write the FIM for the localization problem as $\boldsymbol{C}^{-1} = \boldsymbol{F}(\boldsymbol{w}, \boldsymbol{\theta}) = \sum_{m=1}^{M} w_m \boldsymbol{F}_m(\boldsymbol{\theta})$, where using (3.5) we can compute

$$\boldsymbol{F}_m(\boldsymbol{\theta}) := \frac{(\boldsymbol{\theta} - \boldsymbol{a}_m)(\boldsymbol{\theta} - \boldsymbol{a}_m)^T}{\sigma_m^2 \|\boldsymbol{\theta} - \boldsymbol{a}_m\|_2^2}.$$



(a)

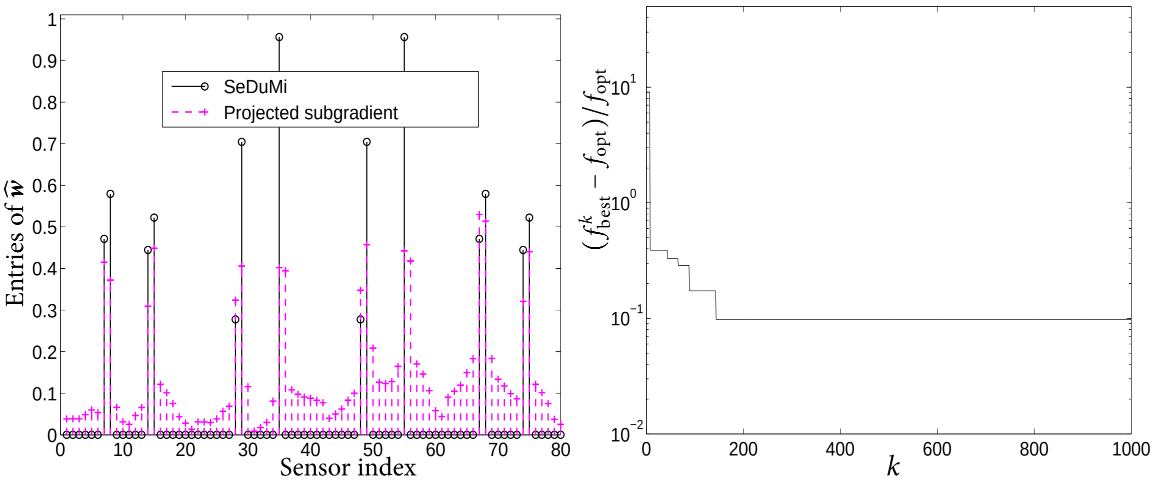
(b)

(c)

Figure 3.1: Sensor placement for target localization with $M = 80$ candidate sensor positions. The thresholds are computed using $R_e = 20$ cm and $P_e = 0.9$. (a) Selection based on sparsity-enhancing iterations with minimum eigenvalue constraints. The Boolean solution is recovered using randomized rounding. (b) Minimum eigenvalue constraints with ℓ_1-norm and concave surrogate based relaxations. Randomized rounding is applied on the concave surrogate based solution. (c) ℓ_1-norm based selection with the trace constraints.

We apply the proposed sensor selection problem to sensor placement design for target localization. To test the proposed algorithms, we use CVX [Grant and Boyd, 2012]. CVX internally calls SeDuMi [Sturm, 1999], a MATLAB implementation of the second-order interior-point methods.

We consider the scenario shown in Fig. 3.1(a) with $M = 80$ sensors to illustrate the sensor place-



(a)

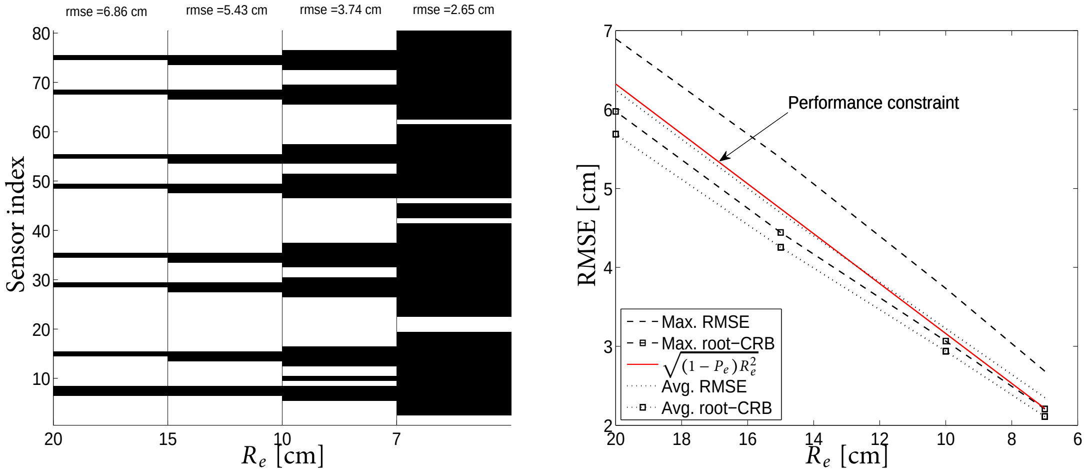
(b)

Figure 3.2: (a) The projected subgradient algorithm used to solve (3.12). (b) Performance of the projected subgradient algorithm. The thresholds are computed using $R_e = 20$ cm and $P_e = 0.9$.

ment problem. Recall that the problem here is to choose the best sensor positions out of $M = 80$ available ones, such that a certain specified localization accuracy is achieved. We grid the target (or surveillance) area of 15×15 m^2 uniformly with a resolution of 1.75 m along both horizontal and vertical directions as shown in Fig. 3.1(a) to obtain the set $\mathcal{U}$ containing $D = 81$ points.

The original nonconvex optimization problem is relaxed to an ℓ_1-norm optimization problem. Alternatively, a concave surrogate function can be used to further enhance the sparsity. The optimization problem with the concave surrogate cost function is iteratively solved by affinely scaling the objective based on the solution from the previous iteration. For the sparsity-enhancing iterative Algorithm 3.2, we use $i_{\max} = 10$ and $\delta = 10^{-8}$. The number of randomizations used in the randomized rounding Algorithm 5.1 is $L = 100$. As observed in the simulations, a solution is typically found in the first batch itself, and a few tens of candidate entries are sufficient. We use the following parameters for the simulations: $\eta = 2$, $\sigma^2 = 2 \times 10^{-5}$, and $P_e = 0.9$.

Fig. 3.1 shows the sensor selection for the distance (range) measurement model. The thresholds are computed with $R_e = 20$ cm and $P_e = 0.9$. The selection shown in Fig. 3.1(a) is based on Algorithm 3.2 with randomized rounding to recover the approximate Boolean solution. The selection results based on the ℓ_1-norm cost with the minimum eigenvalue constraint is shown in Fig. 3.1(b). Fig. 3.1(b) also shows that the solution based on the concave surrogate cost function with the minimum eigenvalue constraint leads to a sparser solution. The selection results based on the trace constraint obtained by solving (3.20) are illustrated in Fig. 3.1(c). The sensors from the same region (close to the red filled boxes in Fig. 3.1(a)) are selected with either one of the two constraints. The sufficient trace constraint has a larger feasible set compared to the stronger sufficient minimum eigenvalue constraint. As a result, for the considered scenario, the minimum eigenvalue constraint



(a)

(b)

Figure 3.3: (a) Solution path of the sensors selected for different values of R_e and $P_e = 0.9$. Maximum RMSE based on selected sensors can be seen on the top of this plot. (b) Maximum and average RMSE of the location estimates based on Gauss-Newton's method, the corresponding maximum and average root-CRB, and the performance constraint in (3.31) for different values of R_e, and $P_e = 0.9$.

leads to a slightly larger ℓ_1-norm compared to the trace constraint.

The optimization problem (3.12) is also solved using the projected subgradient method summarized in Algorithm 3.1 with $k_{\max} = 1000$ iterations. The solution of the projected subgradient is shown in Fig. 3.2(a). The performance of the projected subgradient algorithm is compared to the solution from SeDuMi denoted by f_{opt}, i.e., the convergence $(f_{\text{best}}^k - f_{\text{opt}})/f_{\text{opt}}$ is shown in Fig. 3.2(b). Even though the convergence of the projected subgradient algorithm is very slow, the estimated support after a few hundred iterations can be used along with randomized rounding to further refine the solution. The computation time on the same computer for the projected subgradient algorithm that solves (3.12) is around 8.84 seconds for 1000 iterations while SeDuMi takes around 4.03 seconds to solve the SDP problem in (3.12).

A practical estimator does not meet the CRB in some cases (for instance at low SNRs or finite data records). Therefore, the sensors selected with a specific R_e would lead to an underestimate of the desired MSE. We can account for this gap by choosing R_e appropriately. To this end, we give the entire solution path of the selected sensors for different values of R_e in Fig. 3.3(a). The solution path can be efficiently computed by increasing R_e. The sensors corresponding to some R_e can then be used to meet the desired MSE requirement. The nonlinear model in (3.23) is solved in the least-squares sense iteratively using Gauss-Newton's method with 10 iterations [Kay, 1993]. The maximum root-MSE (RMSE), maximum root-CRB, average RMSE, and average root-CRB of the location estimates of a target within the target area using the selected sensors (as shown in the solution path) for different values of R_e are shown in Fig. 3.3(b). For the considered scenario, both the maximum and average root-CRB satisfy the performance constraint which is given by the inequality in (3.31). The performance constraint is shown as a red solid line in Fig. 3.3b. The maximum RMSE does not satisfy the accuracy requirement specified by a certain R_e, and this can be corrected by using an appropriate (lower) R_e. Moreover, for the considered scenario, the gap between the average RMSE and the performance constraint is still reasonable. We also show the maximum RMSE on top of Fig. 3.3(a).

The proposed framework can be applied to a variety of data models as long as (a1) and (a2) are valid. This is illustrated in [Chepuri and Leus, 2015b] for different models based on bearing measurements, received signal strength, and energy measurements.

Although the illustrated example is related to passive sensing, we underline that the proposed sensor selection framework is not limited to passive sensors.

Remark 3.8 (Active sensor selection).
The sensor selection problem can also be formulated for active sensing. In active sensing, the sensors transmit probing signals (e.g., radar, sonar, remote sensing). The selection parameter w_m for active sensing is a soft parameter used for joint selection and resource allocation [Chepuri et al., 2013b], i.e., $w_m \in [0,1]$ is a resource (e.g., transmit energy) normalized to the maximum prescribed value, and hence, it is dimensionless. The relaxed active sensor selection problem takes the same form as in (3.12). *In fact, minimizing the ℓ_1-norm in active sensor selection minimizes the overall network resources (e.g., overall transmit energy).*

3.7 Dependent observations

Throughout this chapter so far, we have assumed that the observations are a sequence of independent random variables. This assumption is reasonable if the sensors are solely responsible for the noisy observations, for example, due to the internal thermal noise. If the observation signal itself is stochastic in nature or if the observations are subject to external noises or interference, then Assumption 3.2 will be too idealistic. As a consequence of relaxing Assumption 3.2, the FIM will not be linear (or convex) in $\boldsymbol{w}$ anymore. Nevertheless, the FIM will still be a function of $\boldsymbol{w}$ and sparse samplers can be designed using nonlinear and often nonconvex optimization techniques. In some cases, the solution can be computed using convex optimization techniques as illustrated in the following example.

Suppose the unknown vector $\boldsymbol{\theta} \in \mathbb{R}^N$ is related to the observations according to

$$\boldsymbol{x} \sim \mathcal{N}\left(\boldsymbol{h}(\boldsymbol{\theta}), \boldsymbol{\Sigma}\right),$$

where $\boldsymbol{h}(\cdot) : \mathbb{R}^N \longmapsto \mathbb{R}^M$ is a nonlinear function and $\boldsymbol{\Sigma} \in \mathbb{S}^N$ is the noise covariance matrix. As before, we acquire the data using the sparse sensing function $\boldsymbol{\Phi}(\boldsymbol{w})$ to obtain $\boldsymbol{y} = \boldsymbol{\Phi}(\boldsymbol{w})\boldsymbol{x}$.

Using (3.2), we can compute the FIM, and it is given by

$$\boldsymbol{F}(\boldsymbol{w}, \boldsymbol{\theta}) = \left[\boldsymbol{\Phi}\boldsymbol{J}(\boldsymbol{\theta})\right]^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{w}) \left[\boldsymbol{\Phi}\boldsymbol{J}(\boldsymbol{\theta})\right], \tag{3.24}$$

where for the sake of brevity we write $\boldsymbol{\Phi}(\boldsymbol{w}) = \boldsymbol{\Phi}$. Here, $\boldsymbol{J}(\boldsymbol{\theta}) = \partial \boldsymbol{h}(\boldsymbol{\theta})/\partial \boldsymbol{\theta}^T \in \mathbb{R}^{M \times N}$ and

$$\boldsymbol{\Sigma}(\boldsymbol{w}) = \boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T \in \mathbb{R}^{K \times K}$$

is the submatrix of $\boldsymbol{\Sigma}(\boldsymbol{w})$, which includes only the entries corresponding to the selected measurements. Clearly, the FIM (3.24) is no more additive or linear in $\boldsymbol{w}$. Consequently, the constraint $\lambda_{\min}\{\boldsymbol{F}(\boldsymbol{w},\boldsymbol{\theta})\} \geq \lambda_{\text{eig}}$ in its current form is not convex in $\boldsymbol{w}$. This is also true for the trace and determinant constraints.

In what follows, we will provide some simplifications to express the minimum eigenvalue constraint as a convex constraint on $\boldsymbol{w}$. Firstly, we write the noise covariance matrix $\boldsymbol{\Sigma}$ as

$$\boldsymbol{\Sigma} = a\boldsymbol{I} + \boldsymbol{S}, \tag{3.25}$$

where a nonzero $a \in \mathbb{R}$ is chosen such that $\boldsymbol{S} \in \mathbb{R}^{M\times M}$ is invertible and well-conditioned. Using (4.16) in (3.24), we obtain

$$\boldsymbol{F}(\boldsymbol{w},\boldsymbol{\theta}) = \boldsymbol{J}^T(\boldsymbol{\theta})\boldsymbol{\Phi}^T\left(a\boldsymbol{I} + \boldsymbol{\Phi}\boldsymbol{S}\boldsymbol{\Phi}^T\right)^{-1}\boldsymbol{\Phi}\boldsymbol{J}(\boldsymbol{\theta}).$$

We now state the following property.

Property 3.1. *If* $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \text{diag}(\boldsymbol{w})$, *then*

$$\boldsymbol{\Phi}^T\left(a\boldsymbol{I} + \boldsymbol{\Phi}\boldsymbol{S}\boldsymbol{\Phi}^T\right)^{-1}\boldsymbol{\Phi} = \boldsymbol{S}^{-1} - \boldsymbol{S}^{-1}\left[\boldsymbol{S}^{-1} + a^{-1}\text{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}^{-1}. \tag{3.26}$$

Proof. By definition $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \text{diag}(\boldsymbol{w})$. Applying the matrix inversion lemma [Kay, 1993]

$$\boldsymbol{C}(\boldsymbol{B}^{-1}+\boldsymbol{C}^T\boldsymbol{A}^{-1}\boldsymbol{C})^{-1}\boldsymbol{C}^T = \boldsymbol{A} - \boldsymbol{A}(\boldsymbol{A} + \boldsymbol{C}\boldsymbol{B}\boldsymbol{C}^T)^{-1}\boldsymbol{A},$$

with $\boldsymbol{C} = \boldsymbol{\Phi}^T$, $\boldsymbol{B}^{-1} = a\boldsymbol{I}$, and $\boldsymbol{A} = \boldsymbol{S}^{-1}$, it is easy to verify (3.26). □

Therefore, from Property 3.1, we can simplify $\boldsymbol{F}(\boldsymbol{w},\boldsymbol{\theta})$ to

$$\boldsymbol{F}(\boldsymbol{w},\boldsymbol{\theta}) = \boldsymbol{J}^T(\boldsymbol{\theta})\boldsymbol{S}^{-1}\boldsymbol{J}(\boldsymbol{\theta}) - \boldsymbol{J}^T(\boldsymbol{\theta})\boldsymbol{S}^{-1}\left[\boldsymbol{S}^{-1} + a^{-1}\text{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}^{-1}\boldsymbol{J}^T(\boldsymbol{\theta}), \tag{3.27}$$

where by definition $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \text{diag}(\boldsymbol{w})$. In contrast to (3.24), the design parameter $\boldsymbol{w}$ appears only once in (3.27), which makes the problem much easier. Using the Schur complement, the constraint $\lambda_{\min}\{\boldsymbol{F}(\boldsymbol{w},\boldsymbol{\theta})\} \geq \lambda_{\text{eig}}$ can now be equivalently expressed under $a > 0$ as an LMI:

$$\begin{bmatrix} \boldsymbol{S}^{-1} + a^{-1}\text{diag}(\boldsymbol{w}) & \boldsymbol{S}^{-1}\boldsymbol{J}(\boldsymbol{\theta}) \\ \boldsymbol{J}^T(\boldsymbol{\theta})\boldsymbol{S}^{-1} & \boldsymbol{J}^T(\boldsymbol{\theta})\boldsymbol{S}^{-1}\boldsymbol{J}(\boldsymbol{\theta}) - \lambda_{\text{eig}}\boldsymbol{I}_N \end{bmatrix} \geq \boldsymbol{0}_{M+N}, \tag{3.28}$$

which is linear (thus, convex) in $\mathbf{w}$. The above LMI is of size $M + N$ and it is larger than the size-N LMI (3.9), which is related to the independent observation case. Note that the constraint (3.28) also depends on the unknown parameter vector $\boldsymbol{\theta}$. We remark here that for linear measurement models, the above constraint is independent of the unknown parameter vector $\boldsymbol{\theta}$. In other words, in that case, $\boldsymbol{J}(\boldsymbol{\theta})$ will be independent of $\boldsymbol{\theta}$ and will simply be the regression matrix itself.

3.7.1 Convex relaxation

The discrete sparse sensing design problem for nonlinear dependent Gaussian observations is obtained by replacing the LMI constraint (3.10b) with (3.28). The resulting nonconvex optimization problem can then be relaxed to a convex optimization problem along similar lines as explained in Section 3.5.1 or Section 3.5.3. For example, the sparsity-enhancing iterations, i.e., the re-weighted ℓ_1-norm optimization problem (cf. (3.19) and Algorithm 3.2) for the dependent case is given by

$$\begin{aligned} \widehat{\boldsymbol{w}}[i] = \underset{\boldsymbol{w} \in \mathbb{R}^M}{\arg\min} \quad & \sum_{m=1}^{M} \frac{w_m}{\widehat{w}_m[i-1] + \delta} \\ \text{s.to} \quad & \begin{bmatrix} \boldsymbol{S}^{-1} + a^{-1}\text{diag}(\boldsymbol{w}) & \boldsymbol{S}^{-1}\boldsymbol{J}(\boldsymbol{\theta}) \\ \boldsymbol{J}^T(\boldsymbol{\theta})\boldsymbol{S}^{-1} & \boldsymbol{J}^T(\boldsymbol{\theta})\boldsymbol{S}^{-1}\boldsymbol{J}(\boldsymbol{\theta}) - \lambda_{\text{eig}}\boldsymbol{I}_N \end{bmatrix} \succeq \boldsymbol{0}_{M+N}, \; \forall \boldsymbol{\theta} \in \mathcal{U}, \\ & 0 \le w_m \le 1, \quad m = 1, \ldots, M. \end{aligned} \tag{3.29}$$

In what follows, we will illustrate the sparse sensing design for dependent Gaussian observations applied to sensor placement for source localization.

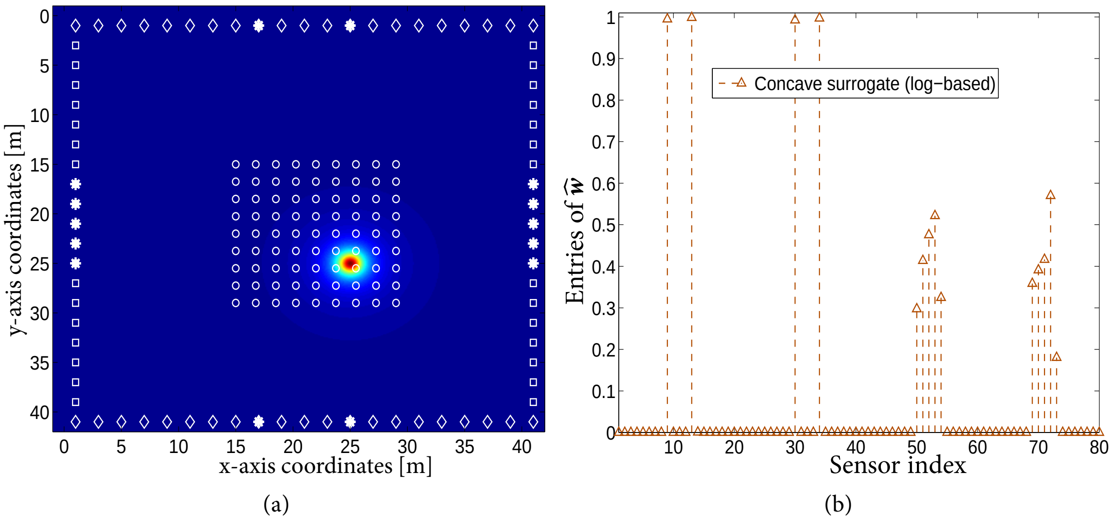


Figure 3.4: Sensor placement for field estimation. Here, the uncorrelated and correlated sensors are denoted by squares (□) and diamonds (◇), respectively. The source domain is indicated by circles (○), while the selected sensors are indicated by (∗). (a) Illustration of a field generated by a unit amplitude point source at location $\boldsymbol{\theta} = [25, 25]^T$ m according to (3.30). Out of $M = 80$ available sensors 14 sensors are selected. (b) Sensor selection is solved using log-based heuristics. The thresholds are computed using $R_e = 20$ m, and $P_e = 0.9$. We use $e = 1$, $\beta = 1$ and $\sigma_e^2 = 2 \times 10^{-5}$.

3.7.2 Numerical example

We consider a similar setup as in Section 3.6, but with another popular measurement model as detailed next. In applications related to field estimation, (active/passive) radar, and sonar, it is important to estimate the location of a point source that emits or reflects energy. Suppose the sensors measure the energy generated by a point source at location $\boldsymbol{\theta} \in \mathbb{R}^N$. The measurements are given as

$$y_m = h_m(\boldsymbol{\theta}) + n_m, \; m = 1, 2, \dots, M, \tag{3.30}$$

where e is the known field intensity emitted or reflected by the point source, the propagation function for some gain $\beta \geq 0$ is modeled as an isotropic exponential attenuation $h_m(\boldsymbol{\theta}) = \sqrt{e}\beta/(\beta + d_m^2)$, and n_m is the noise. Recall from Section 3.6 that $d_m = \|\boldsymbol{\theta} - \boldsymbol{a}_m\|_2$. In this case, we have

$$\boldsymbol{h}(\boldsymbol{\theta}) = [h_m(\boldsymbol{\theta}), h_2(\boldsymbol{\theta}), \dots, h_M(\boldsymbol{\theta})]^T$$

and $\boldsymbol{n} = [n_1, n_2, \dots, n_M]^T$ with $\boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$.

We consider the scenario shown in Fig. 3.4 with $M = 80$ sensors. We assume that the noise correlation matrix is of the form

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{\text{horz}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\text{vert}} \end{bmatrix} \in \mathbb{R}^{M \times M},$$

where $\boldsymbol{\Sigma}_{\text{horz}}$ is the noise correlation matrix corresponding to the horizontally located candidate sensors (denoted by $(\diamond)$ in Fig. 3.4) and $\boldsymbol{\Sigma}_{\text{vert}}$ is the noise correlation matrix corresponding to the vertically located candidate sensors (indicated by $(\square)$ in Fig. 3.4). We further assume that $\boldsymbol{\Sigma}_{\text{horz}} = \sigma_{\text{e}}^2 \left[(1 - \rho)\boldsymbol{I} + \rho\mathbf{1}\mathbf{1}^T\right]$ with correlation co-efficient ρ and nominal noise variance σ_{e}^2, and $\boldsymbol{\Sigma}_{\text{vert}} = \sigma_{\text{e}}^2\boldsymbol{I}$. That is, the vertically located candidate sensors are uncorrelated while the horizontally located candidate sensors are equally correlated.

We use the following simulation parameters: $e = 1$, $\beta = 1$ and $\sigma_{\text{e}}^2 = 2 \times 10^{-5}$, $R_e = 20$ m, $\rho = 0.5$, and $P_e = 0.9$. Fig. 3.4(a) shows the sensor placement obtained by solving Algorithm 3.2, but with (3.18) replaced with (3.29) and with $i_{\max} = 10$ and $\delta = 10^{-8}$. Fig. 3.4(b) shows the selected sensor index. We underline the following observations. Firstly, the sensors from the same region are selected as in Fig. 3.1(a). This is due to the structural similarity of the FIMs corresponding to the models (3.23) and (3.30). Secondly, to achieve the desired performance requirement we see that fewer number of correlated sensors are selected as compared to the number of selected uncorrelated sensors. Finally, as observed in the simulations, for this particular numerical example, the ℓ_1-norm based solution does not result in a sparse selection.

We next end this chapter with some concluding remarks.

3.8 Discussion

In this chapter we discussed discrete sparse sensing for estimation problems. In particular, we focussed on observations that follow a nonlinear model. We used a number of scalar functions related to the Cramér-Rao bound to design the sparse sensing function. The Cramér-Rao bound was used as a weaker surrogate for the error covariance matrix, which for nonlinear models does not admit a closed-form expression. The original nonconvex optimization problem is relaxed using convex relaxation techniques, which can then be efficiently solved in polynomial time. To handle large-scale problems, we have also presented a projected subgradient algorithm. The proposed framework is applied to sensor placement design for localization.

3.A Performance thresholds

Trace and minimum eigenvalue constraints

We can relate the accuracy requirement and CRB using Chebyshev's inequality [Cover and Thomas, 2012]

$$\Pr(\|\boldsymbol{\varepsilon}\|_2 \geq R_e) \leq \operatorname{tr}\{\boldsymbol{C}\}/R_e^2$$

which can be equivalently expressed as $\Pr(\|\boldsymbol{\varepsilon}\|_2 \leq R_e) \leq 1 - \operatorname{tr}\{\boldsymbol{C}\}/R_e^2$. Combining this inequality together with $\Pr(\|\boldsymbol{\varepsilon}\|_2 \leq R_e) \geq P_e$ in (3.7) results in the following sufficient condition

$$\operatorname{tr}\{\boldsymbol{C}\} \leq \lambda_{\text{tr}} = (1 - P_e)R_e^2. \tag{3.31}$$

Each eigenvalue of $\boldsymbol{C}^{-1}$ is greater than $\lambda_{\min}(\boldsymbol{F})$, and as a result, $\operatorname{tr}\{\boldsymbol{C}\} \leq N\lambda_{\min}^{-1}(\boldsymbol{F})$. Hence, a stronger sufficient condition (with a smaller feasible set) is $N\lambda_{\min}^{-1}(\boldsymbol{F}) \leq (1 - P_e)R_e^2$, or equivalently [Wang et al., 2009]

$$\lambda_{\min}(\boldsymbol{F}) \geq \lambda_{\text{eig}} = \frac{N}{R_e^2}\left(\frac{1}{1 - P_e}\right).$$

Determinant constraint

The determinant constraint is related to the volume or the mean radius of the confidence ellipsoid that contains $\boldsymbol{\varepsilon} = \boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}$ with probability P_e. Such a confidence ellipsoid can be expressed as

$$\mathcal{E} = \{\boldsymbol{\varepsilon} \,|\, \boldsymbol{\varepsilon}^T \boldsymbol{F}^{-1} \boldsymbol{\varepsilon} \leq \xi\},$$

where ξ is a constant that depends on P_e. Assuming $\boldsymbol{F}$ has ordered eigenvalues $\lambda_{\max} \geq \lambda_2 \cdots \geq \lambda_{\min}$, the length of the nth semi-axis of the ellipsoid $\mathcal{E}$ will be $\sqrt{\xi/\lambda_n}$. The geometric mean radius of the confidence ellipsoid $\mathcal{E}$ is given by

$$\bar{R}_e = \sqrt{\xi}/(\det\{\boldsymbol{F}\})^{1/2N},$$

which gives a quantitative measure of how informative the observations are. For the estimates to be within the confidence ellipsoid $\mathcal{E}$, we use the constraint

$$\ln\det\{\boldsymbol{F}\} \geq 2N \ln \frac{\sqrt{\xi}}{\bar{R}_e} = \lambda_{\text{det}},$$

where $\bar{R}_e$ and $\sqrt{\xi}$ specify the required accuracy, and are assumed to be known. A typical choice for ξ is $\xi = F^{-1}_{\mathcal{X}^2_N}(P_e)$. Here, $F^{-1}_{\mathcal{X}^2_N}$ is the cumulative distribution function of a chi-squared random variable with N degrees of freedom. This performance measure is related to the *D-optimality*.

3.B Projected Newton's method

In order to analyze the complexity of the interior point methods, we briefly describe the projected Newton's method. The Newton's method for an SDP problem in the inequality form is adapted to suit our problem [Boyd and Vandenberghe, 2004, p. 619].

The optimization problem in (3.12) can be approximated using the log-determinant barrier function which is given as

$$\underset{\boldsymbol{w}\in[0,1]^N}{\arg\min} \quad \psi(\boldsymbol{w}) = t\mathbf{1}_M^T\boldsymbol{w} - \ln\det\{\sum_{m=1}^{M} w_m\boldsymbol{F}_m - \lambda_{\text{eig}}\boldsymbol{I}_{DN}\},$$

where $t > 0$ is a parameter to tune the approximation. The projected Newton's update equation is given by

$$\boldsymbol{w}^{k+1} = \mathcal{P}_{\mathcal{W}}\left(\boldsymbol{w}^k - \alpha^k\boldsymbol{H}_{\psi}^{-1}(\boldsymbol{w}^k)\boldsymbol{g}_{\psi}(\boldsymbol{w}^k)\right), \tag{3.32}$$

where the (i,j)th entry of the Hessian matrix is given by

$$[\boldsymbol{H}_{\psi}(\boldsymbol{w}^k)]_{i,j} = \left.\frac{\partial^2\psi(\boldsymbol{w})}{\partial w_i\partial w_j}\right|_{\boldsymbol{w}=\boldsymbol{w}^k} = \text{tr}\{\boldsymbol{S}^{-1}\boldsymbol{F}_i\boldsymbol{S}^{-1}\boldsymbol{F}_j\},$$

and the ith entry of the gradient vector is given by

$$[\boldsymbol{g}_{\psi}(\boldsymbol{w}^k)]_i = \left.\frac{\partial\psi(\boldsymbol{w})}{\partial w_i}\right|_{\boldsymbol{w}=\boldsymbol{w}^k} = t + \text{tr}\{\boldsymbol{S}^{-1}\boldsymbol{F}_i\}.$$

Here, we have introduced the matrix $\boldsymbol{S} = \sum_{m=1}^{M} w_m\boldsymbol{F}_m - \lambda_{\text{eig}}\boldsymbol{I}_{DN}$, and recall the projection operator $\mathcal{P}_{\mathcal{W}}(\cdot)$ defined in (3.15). The step-length α^k is chosen by line-search.

3.C Power iterations

We briefly describe the power iterations [Golub and Van Loan, 1996] to compute the minimum eigenvalue of a matrix $\boldsymbol{F} \in \mathbb{S}^N$. Assuming $\boldsymbol{F}$ has ordered eigenvalues $\lambda_{\max} \geq \lambda_2 \cdots \geq \lambda_{\min}$, the power iterations

$$\boldsymbol{v}^{k+1} = \frac{\boldsymbol{F}\boldsymbol{v}^k}{\|\boldsymbol{F}\boldsymbol{v}^k\|_2}, \text{ and } \lambda^{k+1} = \frac{(\boldsymbol{v}^{k+1})^T\boldsymbol{F}\boldsymbol{v}^{k+1}}{\|\boldsymbol{v}^{k+1}\|_2},$$

converge to the maximum eigenvector $\boldsymbol{v}_{\max}$ and maximum eigenvalue $\lambda_{\max}$, respectively, as $k \to \infty$. Here, we use $\boldsymbol{v}^0 = [1, \mathbf{0}_{N-1}^T]^T$. By forming a matrix $\bar{\boldsymbol{F}} = \lambda_{\max}\boldsymbol{I}_N - \boldsymbol{F}$ which has the dominant eigenvalue $\lambda_{\max} - \lambda_{\min}$, we can apply the above power iterations on $\bar{\boldsymbol{F}}$ to compute $\lambda_{\max} - \lambda_{\min}$ and $\boldsymbol{v}_{\min}$, and thus the minimum eigenvalue of $\boldsymbol{F}$ and it's corresponding eigenvector.

Part of this chapter was published as: S.P. Chepuri and G. Leus, "Compression schemes for time-varying sparse signals", in *Proc. of Asilomar Conf. Signals, systems, and Computers (Asilomar 2014)*, Pacific Grove, California, USA, Nov. 2014, and S.P. Chepuri and G. Leus, "Sparsity-Promoting Adaptive Sensor Selection for nonlinear Filtering", in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, Florence, Italy, May 2014.

Chapter 4

Sparse Sensing for Filtering

Contents

4.1 Introduction

In this chapter we will extend the discrete sparse sensing framework for estimation introduced in Chapter 3 to nonlinear filtering problems. The proposed framework is useful for a variety of applications related to target/bearing tracking, dynamic field estimation, and nonlinear filtering in general, where the sensors (or measurements) are scheduled or activated parsimoniously to increase their battery lifetime as well as to reduce the communications and inference costs. The time-varying state parameters are described through a state-space model, which is assumed to be known. Given the state-space model, we will design sparse sensing mechanisms and develop the associated risk functions that quantify the filtering accuracy.

Over the past decade, sensor scheduling/polling problems for state estimation of linear dynamical systems (i.e., linear filtering problems) have been extensively studied with different flavors, such as different optimization criteria from experiment design, myopic (one-step ahead) and nonmyopic (a longer time horizon) scheduling, heuristic (submodular, convex, among others) algorithms, and budget constraints; see [Hernandez et al., 2004, Zuo et al., 2007, Krause, 2008, Joshi and Boyd, 2009, Fu et al., 2012, Jiang et al., 2013, Liu et al., 2014, Zhan et al., 2010]. For example, [Joshi and Boyd, 2009] proposed convex relaxation techniques, whereas [Krause, 2008] proposed greedy algorithms using submodular cost functions for sensor scheduling. In [Liu et al., 2014], the design of optimal periodic sensor scheduling (over an infinite time horizon) with restrictions on the total number of sensor activations was studied for additive linear Gaussian models. The optimal experiment design problems are well-studied for observations that follow a linear model with uncorrelated noise components for which the MSE covariance matrix has a known closed form. However, it is in general difficult to compute the error covariance matrix in closed form for nonlinear dynamical systems. Therefore, the above methods cannot be used directly.

We organize this chapter into three parts discussing sparse sensing mechanisms for: independent observations, dependent observations, and structured state sequences. To begin with, we focus on nonlinear state-space models (non additive and/or non Gaussian) with independent observations. In [Masazade et al., 2012], sensor selection for target tracking based on extended Kalman filtering has been proposed, in which the selection is performed by designing a sparse gain matrix. Moreover, [Masazade et al., 2012] focuses on an additive Gaussian nonlinear model, that is linearized around the (noisy) past state estimate. This chapter, on the other hand, deals with general nonlinear models, without an explicit linearization. For a general nonlinear filtering problem, we use a number of scalar measures (from experiment design) of the posterior Cramér-Rao bound matrix. We saw in Chapter 3 that for nonlinear estimation problems the risk functions based on the Cramér-Rao bound depend on the true parameter. Similarly, for nonlinear filtering problems the risk functions based on the posterior Cramér-Rao bound depend on the current as well as the previous true state parameters. The sensing patterns are time-varying, and we do not need actual measurements to design them (i.e., they are data independent and model-driven). However, running a nonlinear filter in parallel and incorporating the entire history of (actual) measurements up to that point significantly simplifies the design problem. We further model the evolution of the sensing pattern in time to control its smoothness. This is beneficial for mobile sensing (e.g., sensing with sensors mounted on bicycles or automobiles), for instance. Next, we extend the framework to include nonlinear additive Gaussian models with correlated measurements.

We also study discrete sparse sensing for time-varying structured signals. In particular, we will illustrate this framework with time-varying sparse signals with possibly time-varying sparsity patterns and/or order. This has received a lot of attention in the recent past through *compressive sensing* (CS) [Baraniuk, 2007]. Time-varying sparse signal reconstruction has been studied in the past leading to various forms of sparsity-aware filters [Vaswani, 2008, Angelosante et al., 2009, Carmi et al., 2010,Malioutov et al., 2010], and are applied to problems like visual surveillance [Warnell et al., 2012] and target localization [Farahmand et al., 2014]. We study the design of sensing matrices for such problems; however, the focus will not be on the signal recovery itself.

Sensing matrix design for sparse recovery has been studied in various forms. For example, in [Eldar and Kutyniok, 2012, Ch. 6], [Haupt et al., 2009] the variance of the distribution from which the (random) sensing matrices are generated is designed such that the average information gain is maximized. The Bayesian CS framework [Ji et al., 2008] allows to quantify the sparse reconstruction error through the so-called error bars, which again allows to adaptively design the sensing matrices. Both [Haupt et al., 2009] and [Ji et al., 2008] use experiment design techniques with performance measures like differential entropy to adaptively learn the sensing matrix starting from a random matrix. In [El Badawy et al., 2014], a greedy algorithm based on a submodular performance measure has been proposed for sensing operator design for a signal lying in the union of subspaces. However, the sensing design schemes discussed above are mostly limited to time-invariant signals and/or systems without any state-space representation. Hence, they are not adaptive in the true sense, and are not meant for tracking the signal variation over space and/or time. On the other hand, the proposed sensing matrix is designed at each time step based on the entire history of measurements and known dynamics described through a state-space model. Towards the end, we will also discuss a few extensions of the proposed framework to include general structured signals, such as group sparse and smooth signals.

4.2 Sensing time-varying observations

We consider a nonlinear measurement model for observing an unknown dynamic parameter $\boldsymbol{\theta}_k \in \mathbb{R}^{N\times 1}$ at time k:

$$x_{k,m} = h_{k,m}(\boldsymbol{\theta}_k, n_{k,m}), m = 1, 2, \dots, M, \tag{4.1}$$

where $h_{k,m}(\cdot,\cdot)$ is a nonlinear functional of the unknown vector $\boldsymbol{\theta}_k$ and the noise component $n_{k,m}$. The spatial (and/or temporal) measurements at temporal block k, $\{x_{k,m}\}_{m=1}^{M}$, are stacked in the measurement vector $\boldsymbol{x}_k = [x_{k,1}, x_{k,2}, \dots, x_{k,M}]^T \in \mathbb{R}^{M\times 1}$.

The unknown parameter is assumed to obey the following dynamical model:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{A}_k\boldsymbol{\theta}_k + \boldsymbol{u}_k, \tag{4.2}$$

where $\boldsymbol{A}_k \in \mathbb{R}^{N\times N}$ is the state transition matrix and $\boldsymbol{u}_k \in \mathbb{R}^{N\times 1}$ is the process noise that accounts for any unmodeled dynamics. Here, we model $\boldsymbol{u}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{u}})$, where $\boldsymbol{\Sigma}_{\boldsymbol{u}} \in \mathbb{R}^{N\times N}$ represents the state noise covariance matrix.

At each time instance k, we acquire the data $\boldsymbol{x}_k$ via the discrete sparse sensing mechanism that was introduced in Chapter 2. Specifically, we acquire data as

$$\boldsymbol{y}_k = \operatorname{diag}_{\mathrm{r}}(\boldsymbol{w}_k)\boldsymbol{x}_k = \boldsymbol{\Phi}_k(\boldsymbol{w}_k)\boldsymbol{x}_k$$

where $\boldsymbol{\Phi}_k(\boldsymbol{w}_k) = \mathrm{diag}_{\mathrm{r}}(\boldsymbol{w}_k) \in \{0,1\}^{K\times M}$ is the sparse sensing matrix whose entries are determined by the time-varying selection vector

$$\boldsymbol{w}_k = [w_{k,1}, w_{k,2}, \ldots, w_{k,M}]^T \in \{0,1\}^M.$$

Here, the time-varying variable $w_{k,m} = (0)1$ indicates whether $x_{k,m}$ is (not) selected.

We are interested in cases where $K \ll M$ and K is not known. The reduced dimension data vector $\boldsymbol{y}_k \in \mathbb{R}^K$ is used instead of $\boldsymbol{x}_k \in \mathbb{R}^M$ together with (4.2) to solve the nonlinear filtering problem. Therefore, our aim is to design, for each k, a sparsest $\boldsymbol{w}_k$ (and, hence a sequence of matrices $\{\boldsymbol{\Phi}_k(\boldsymbol{w}_k), k \in \mathbb{N}\}$) based on the entire history of measurements up to that point $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{k-1}\}$, such that a desired accuracy on the (*a posteriori*) estimate of $\boldsymbol{\theta}_k$ is guaranteed. For such nonlinear filtering problems, the risk function $f(\boldsymbol{w}_k)$ that quantifies the filtering accuracy is discussed next.

4.3 $f(\boldsymbol{w})$ for filtering

For general nonlinear filtering problems, it is difficult to compute the posterior error covariance matrix in closed form that is suitable for optimization. Therefore, along the lines of Section 3.3 of Chapter 3, we will discuss a weaker surrogate that can be optimized instead of the posterior error covariance.

We now recall Assumption 3.1 from Chapter 3 as:

Assumption 4.1 (Regularity condition). *The log-likelihood of the measurements* $\ln p_k(\boldsymbol{y};\boldsymbol{\theta}_k)$ *satisfies the regularity condition, that is,* $\mathbb{E}\left\{\frac{\partial \ln p_k(\boldsymbol{y};\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}}\right\} = \boldsymbol{0}$, *for all k, where* $p_k(\boldsymbol{y};\boldsymbol{\theta}_k)$ *is the probability density function (pdf) of* $\boldsymbol{y}_k$ *at time instance k parameterized by the unknown vector* $\boldsymbol{\theta}_k$.

Under the above assumption, the *a posteriori* estimate of $\boldsymbol{\theta}_k$ denoted by $\widehat{\boldsymbol{\theta}}_{k|k}$ satisfies the well-known posterior Cramér-Rao bound (PCRB) inequality given by

$$\mathbb{E}\left\{(\widehat{\boldsymbol{\theta}}_{k|k} - \boldsymbol{\theta}_k)(\widehat{\boldsymbol{\theta}}_{k|k} - \boldsymbol{\theta}_k)^T\right\} \geq \boldsymbol{C}_k(\boldsymbol{\theta}_k) = \boldsymbol{F}_k^{-1}(\boldsymbol{\theta}_k), \tag{4.3}$$

where $\boldsymbol{C}_k(\boldsymbol{\theta}_k)$ is the PCRB matrix and $\boldsymbol{F}_k(\boldsymbol{\theta}_k)$ is the posterior Fisher information matrix (FIM), which is given by the following recursion [Tichavsky et al., 1998]:

$$\boldsymbol{F}_k(\boldsymbol{w}_k, \boldsymbol{\theta}_k, \{\boldsymbol{\theta}_{\kappa-1}\}_{\kappa=1}^k) = \overbrace{(\boldsymbol{\Sigma}_{\boldsymbol{u}} + \boldsymbol{A}_k \boldsymbol{F}_{k-1}^{-1}(\{\boldsymbol{\theta}_{\kappa-1}\}_{\kappa=1}^k)\boldsymbol{A}_k^T)^{-1}}^{\boldsymbol{F}_{\mathrm{prior},k-1}(\{\boldsymbol{\theta}_{\kappa-1}\}_{\kappa=1}^k)} + \boldsymbol{F}_{\mathrm{obs},k}(\boldsymbol{w}_k, \boldsymbol{\theta}_k) \in \mathbb{R}^{N\times N}, \tag{4.4}$$

with the FIM related to the observations given by

$$\boldsymbol{F}_{\mathrm{obs},k}(\boldsymbol{w}_k, \boldsymbol{\theta}_k) = \mathbb{E}\left\{\left(\frac{\partial \ln p_k(\boldsymbol{y};\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}_k}\right)\left(\frac{\partial \ln p_k(\boldsymbol{y};\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}_k}\right)^T\right\} \in \mathbb{R}^{N\times N}. \tag{4.5}$$

The first term of (4.4) is the prior information related to the history up to time instance $k-1$.

We now introduce the independence assumption.

Assumption 4.2 (Independent observations). *The observations $\{x_{k,m}\}_{m=1}^{M}$ at time k are a sequence of independent random variables.*

Under Assumption 4.2, the information measure from each observation is *additive* in the variables $\{w_{k,m}\}_{m=1}^{M}$, which is intuitive as each independent measurement contributes some information (we have seen this property before in Chapter 3). Using this property, we can further simply (4.4) to arrive at

$$\boldsymbol{F}_k\left(\boldsymbol{w}_k, \boldsymbol{\theta}_k, \{\boldsymbol{\theta}_{\kappa-1}\}_{\kappa=1}^{k}\right) = \boldsymbol{F}_{\text{prior},k-1}(\{\boldsymbol{\theta}_{\kappa-1}\}_{\kappa=1}^{k}) + \sum_{m=1}^{M} w_{k,m}\boldsymbol{F}_{k,m}(\boldsymbol{\theta}_k), \tag{4.6}$$

where

$$\boldsymbol{F}_{k,m}(\boldsymbol{\theta}_k) = \mathbb{E}\left\{\left(\frac{\partial \ln p_{k,m}(x;\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}_k}\right)\left(\frac{\partial \ln p_{k,m}(x;\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}_k}\right)^T\right\} \in \mathbb{R}^{N\times N}$$

with $p_{k,m}(x;\boldsymbol{\theta}_k)$ being the pdf of $x_{k,m}$ at time instance k parameterized by the unknown vector $\boldsymbol{\theta}_k$. The posterior FIM (4.6) depends on the true state $\boldsymbol{\theta}_k$ at time k as well as all the previous states.

Remark 4.1 (Additive Gaussian linear state-space models).
For a linear measurement model: $x_{k,m} = \boldsymbol{h}_{k,m}^T\boldsymbol{\theta}_k + n_{k,m}, m = 1, 2, \ldots, M$, (i.e., $h_{k,m}(\boldsymbol{\theta}_k, n_{k,m}) := \boldsymbol{h}_{k,m}^T\boldsymbol{\theta}_k + n_{k,m}$) where $\boldsymbol{h}_{k,m} \in \mathbb{R}^N$ is the regressor. Assuming that the noise components $\{n_{k,m}\}_{m=1}^{M}$ are i.i.d. Gaussian with variance σ^2, the FIM for linear filtering is given by

$$\boldsymbol{F}_k(\boldsymbol{w}_k) = (\boldsymbol{\Sigma}_{\boldsymbol{u}} + \boldsymbol{A}_k\boldsymbol{F}_{k-1}^{-1}\boldsymbol{A}_k^T)^{-1} + \frac{1}{\sigma^2}\sum_{m=1}^{M} w_{k,m}\boldsymbol{h}_{k,m}\boldsymbol{h}_{k,m}^T. \tag{4.7}$$

The PCRB for linear state-space models in additive Gaussian noise is equal to the posterior error covariance, and it is independent of the unknown state vectors.

As a performance measure, we constrain the posterior estimation error $\boldsymbol{\varepsilon}_k = \widehat{\boldsymbol{\theta}}_{k|k} - \boldsymbol{\theta}_k$ to be within an origin centered circle of radius R_e with a probability higher than P_e, i.e.,

$$\Pr(\|\boldsymbol{\varepsilon}_k\|_2 \leq R_e) \geq P_e, \tag{4.8}$$

where R_e and P_e are specified to achieve a desired accuracy. This accuracy constraint is satisfied by the two popular experiment design criteria (cf. Section 3.3 of Chapter 3):

1. Trace constraint, which is related to the A-optimality measure. A sufficient condition for (4.8) is (see Appendix 3.A)

$$f(\boldsymbol{w}_k) := \operatorname{tr}\left\{\left(\boldsymbol{F}_k(\boldsymbol{w}_k, \boldsymbol{\theta}_k, \{\boldsymbol{\theta}_{\kappa-1}\}_{\kappa=1}^{k})\right)^{-1}\right\} \leq \lambda_{\text{tr}} = (1-P_e)R_e^2.$$

2. Minimum eigenvalue constraint, which is related to the E-optimality measure. A sufficient condition for (4.8) related to the eigenvalue constraint is (see Appendix 3.A)

$$f(\boldsymbol{w}_k) := \lambda_{\min}\left\{\boldsymbol{F}_k\left(\boldsymbol{w}_k, \boldsymbol{\theta}_k, \{\boldsymbol{\theta}_{\kappa-1}\}_{\kappa=1}^{k}\right)\right\} \geq \lambda_{\text{eig}} = \frac{N}{R_e^2}\left(\frac{1}{1-P_e}\right).$$

The above inequality can be alternatively expressed as the following linear matrix inequality (LMI):

$$\boldsymbol{F}_{\text{prior},k-1}(\{\boldsymbol{\theta}_{\kappa-1}\}_{\kappa=1}^{k}) + \sum_{m=1}^{M} w_{k,m}\boldsymbol{F}_{k,m}(\boldsymbol{\theta}_k) \succeq \lambda_{\text{eig}}\boldsymbol{I}_N. \tag{4.9}$$

We underline here that the determinant constraint similar to Section 3.5.5 of Chapter 3 can also be used as a reasonable performance measure. Unless the process noise is zero (e.g., a deterministic trajectory), the true states are not known. However, in practice $\boldsymbol{\theta}_k$ takes values within a certain domain denoted by the set $\mathcal{U}_k$. Therefore, we constrain (4.9) for all $\boldsymbol{\theta}_\kappa \in \mathcal{U}_\kappa, \kappa = 0, 1, 2, \ldots, k$.

As discussed in Chapter 3, the trace constraint, which can also be represented by LMIs, has a larger feasible set as compared to the minimum eigenvalue constraint. However, the resulting optimization problem is computationally less attractive compared to the minimum eigenvalue constraint (as we show later on in Section 3.5.5). For this reason, we focus on the minimum eigenvalue (LMI) constraint from now on.

4.4 Problem statement

The *adaptive discrete sparse sensing* (a.k.a. adaptive sensor selection) problem can be interpreted as the problem of choosing the best subset of sensors out of the M available sensors to acquire measurements for time step k such that a certain accuracy on the estimate $\widehat{\boldsymbol{\theta}}_{k|k}$ is guaranteed. Thus, the adaptive sensor selection problem can be formally stated as follows.

Problem 4.1 (Adaptive discrete sparse sensing)**.** *Given the state-space model* (4.1) *and* (4.2)*, at each time step k, based on the entire history of measurements up to that point find a sparsest* $\boldsymbol{w}_k \in \{0,1\}^M$ *, which satisfies the accuracy constraint* $\boldsymbol{F}_{\text{prior},k-1}(\{\boldsymbol{\theta}_{\kappa-1}\}_{\kappa=1}^{k}) + \sum_{m=1}^{M} w_{k,m}\boldsymbol{F}_{k,m}(\boldsymbol{\theta}_k) \succeq \lambda_{\text{eig}}\boldsymbol{I}_N$, $\forall \boldsymbol{\theta}_\kappa \in \mathcal{U}_\kappa, \kappa = 0, 1, 2, \ldots, k$.

At each time step, the above design problem is a specialization of the discrete sparse sensing problem (P0) introduced in Chapter 2, that is,

$$\begin{aligned}
&\underset{\boldsymbol{w}_k}{\arg\min} \quad \|\boldsymbol{w}_k\|_0 && \text{(4.10a)}\\
&\text{s.t. } \boldsymbol{F}_{\text{prior},k-1}(\{\boldsymbol{\theta}_{\kappa-1}\}_{\kappa=1}^{k}) + \sum_{m=1}^{M} w_{k,m}\boldsymbol{F}_{k,m}(\boldsymbol{\theta}_k) \succeq \lambda_{\text{eig}}\boldsymbol{I}_N, &&\\
&\qquad\qquad\qquad\qquad \forall \boldsymbol{\theta}_\kappa \in \mathcal{U}_\kappa, \kappa = 0, 1, 2, \ldots, k, && \text{(4.10b)}\\
&\quad \boldsymbol{w}_k \in \{0,1\}^M. && \text{(4.10c)}
\end{aligned}$$

This is a nonconvex Boolean optimization problem. Clearly, the number of LMI constraints (4.10b) depends on $|\mathcal{U}_k|$ and this increases with k, that is, the number of LMI constraints is $\sum_{\kappa=0}^{k} |\mathcal{U}_\kappa|$. For example, if $|\mathcal{U}_\kappa| = D, \forall \kappa$, then at time step k, the optimization problem (4.10) will have D^k size-N

LMI constraints. To reduce the computational complexity, the prior Fisher $\boldsymbol{F}_{\text{prior},k-1}(\{\boldsymbol{\theta}_{\kappa-1}\}_{\kappa=1}^{k})$ can be simply evaluated at the past estimates

$$\tilde{\boldsymbol{\theta}}_{k-1} := \widehat{\boldsymbol{\theta}}_{k-1|k-1},$$

obtained by solving any nonlinear filter (e.g., extended Kalman filter, unscented Kalman filter, or particle filters). That is, we approximate $\boldsymbol{F}_k(\boldsymbol{w}_k, \boldsymbol{\theta}_k, \{\boldsymbol{\theta}_{\kappa-1}\}_{\kappa=1}^{k})$ as

$$\boldsymbol{F}_k(\boldsymbol{w}_k, \boldsymbol{\theta}_k) \approx \overbrace{(\boldsymbol{\Sigma}_{\boldsymbol{u}} + \boldsymbol{A}_k \boldsymbol{F}_{k-1}^{-1}(\tilde{\boldsymbol{\theta}}_{k-1})\boldsymbol{A}_k^T)^{-1}}^{\boldsymbol{F}_{\text{prior},k-1}(\tilde{\boldsymbol{\theta}}_{k-1})} + \boldsymbol{F}_{\text{obs},k}(\boldsymbol{w}_k, \boldsymbol{\theta}_k) \ \in \mathbb{R}^{N \times N},$$

Therefore, running an extended Kalman filter in parallel (and incorporating the history of actual measurements) significantly reduces the number of constraints. Henceforth, we will not write the prior FIM $\boldsymbol{F}_{\text{prior},k-1}(\tilde{\boldsymbol{\theta}}_{k-1})$ as an explicit function of $\tilde{\boldsymbol{\theta}}_{k-1}$, i.e., we will simply write it as $\boldsymbol{F}_{\text{prior},k-1}$. As a consequence, (4.10) simplifies to

$$\begin{aligned} \underset{\boldsymbol{w}_k}{\arg\min} \quad & \|\boldsymbol{w}_k\|_0 && \text{(4.11a)} \\ \text{s.t. } \boldsymbol{F}_{\text{prior},k-1} + & \sum_{m=1}^{M} w_{k,m} \boldsymbol{F}_{k,m}(\boldsymbol{\theta}_k) \succeq \lambda_{\text{eig}} \boldsymbol{I}_N, \forall \boldsymbol{\theta}_k \in \mathcal{U}_k, && \text{(4.11b)} \\ \boldsymbol{w}_k & \in \{0,1\}^M. && \text{(4.11c)} \end{aligned}$$

The posterior error covariance matrix and the prediction computed in the extended Kalman filter algorithm (see Appendix 4.A) can be used as a guideline to determine $\mathcal{U}_k$ (we will discuss this with an example in Section 4.6).

4.5 Solvers

In this section, we provide an algorithm to solve the proposed optimization problem. In addition, we also model the evolution of $\boldsymbol{w}_k$ in time, which is useful to control the transient nature of the sensing patterns.

4.5.1 Convex approximation based on ℓ_1-norm

The optimization problem in (4.10) is nonconvex due to the ℓ_0(-quasi) norm cost function and the Boolean constraint. We use the traditional best convex surrogate for the ℓ_0(-quasi) norm based on the ℓ_1-norm heuristic, and the Boolean constraint is relaxed to the convex box constraint $[0,1]^M$. Due to this box constraint, the ℓ_1-norm will simply be the affine function $\mathbf{1}_M^T \boldsymbol{w}_k$. The relaxed optimization problem is of the form (R0) introduced in Chapter 2, and is given as

$$\widehat{\boldsymbol{w}}_k = \arg\min_{\boldsymbol{w}_k} \quad \mathbf{1}_M^T \boldsymbol{w}_k \tag{4.12a}$$

$$\text{s.t. } \boldsymbol{F}_{\text{prior},k-1} + \sum_{m=1}^{M} w_{k,m} \boldsymbol{F}_{k,m}(\boldsymbol{\theta}_k) \succeq \lambda_{\text{eig}} \boldsymbol{I}_N, \ \forall \boldsymbol{\theta}_k \in \mathcal{U}_k, \tag{4.12b}$$

$$0 \leq w_{k,m} \leq 1, m = 1, 2, \ldots, M, \tag{4.12c}$$

where the LMI constraint (4.12b) is convex on $\boldsymbol{w}_k \in [0,1]^M$.

The relaxed optimization problem is a standard semidefinite programming problem that can be solved efficiently in polynomial time using off-the-shelf solvers like SeDuMi [Sturm, 1999] or the projected subgradient algorithm developed in Section 3.5.2 of Chapter 3. Further, a concave surrogate based on the sum-of-logarithms can be used to approximate the ℓ_0(-quasi) norm to obtain sparsity enhancing iterations as discussed in Section 3.5.3 of Chapter 3. An approximate Boolean solution can then be recovered by randomization rounding as explained in Section 3.5.4 of Chapter 3.

4.5.2 Smooth sensing

We now model the evolution of $\boldsymbol{w}_k$ in time. A smooth evolution of the selection vector is important for mobile sensing to control the transient nature of a mobile sensor—a spatial sampling device. In other words, smoothness in the selection vector ensures an easy hand-off between the selected sensors.

The evolution of the selection vector is modeled as a linear recursion

$$\boldsymbol{w}_k = \boldsymbol{B}_k \boldsymbol{w}_{k-1} + \boldsymbol{v}_k, \tag{4.13}$$

where $\boldsymbol{B}_k \in \mathbb{R}^{M \times M}$ could be a banded matrix and $\boldsymbol{v}_k \in \mathbb{R}^M$ is the process noise vector. The smoothness depends on the construction of the matrix $\boldsymbol{B}_k$. In order to incorporate the smoothing effect on the sensing pattern between subsequent time instances, we use the sparse estimate $\widehat{\boldsymbol{w}}_{k-1}$ instead of $\boldsymbol{w}_{k-1}$. The optimization problem taking into account the smoothness is given as

$$\widehat{\boldsymbol{w}}_{k,\text{sm}} = \arg\min_{\boldsymbol{w}_k} \mathbf{1}_M^T \boldsymbol{w}_k + \mu \left\| \boldsymbol{w}_k - \boldsymbol{B}_k \widehat{\boldsymbol{w}}_{k-1,\text{sm}} \right\|_2^2 \tag{4.14a}$$

$$\text{s.t. } \boldsymbol{F}_{\text{prior},k-1} + \sum_{m=1}^{M} w_{k,m} \boldsymbol{F}_{k,m}(\boldsymbol{\theta}_k) \succeq \lambda \boldsymbol{I}_N, \forall \boldsymbol{\theta}_k \in \mathcal{U}_k, \tag{4.14b}$$

$$0 \leq w_{k,m} \leq 1, m = 1, 2, \ldots, M, \tag{4.14c}$$

where μ is the smoothness controlling parameter.

4.6 Numerical example: sensor scheduling

We now apply the developed theory of adaptive sparse sensing to *sensor scheduling* for target tracking based on distance measurements. The sensor scheduling problem can be interpreted as the problem of polling (or activating) the best subset of sensors from a large pool of available sensors, such that a desired tracking accuracy is achieved. Sensor scheduling is typically used to increase the lifetime of battery powered sensor nodes, but also to reduce the communications and inferring costs.

At each time step k, the selected sensors are used to estimate the state vector $\boldsymbol{\theta}_k = [\boldsymbol{p}_k^T, \dot{\boldsymbol{p}}_k^T]^T \in \mathbb{R}^{4\times1}$, where $\boldsymbol{p}_k \in \mathbb{R}^{2\times1}$ is the target position vector and $\dot{\boldsymbol{p}}_k \in \mathbb{R}^{2\times1}$ is the velocity vector. We assume that M sensors are, respectively, located at known two-dimensional positions $\{\boldsymbol{a}_m\}_{m=1}^M$. They are each capable of measuring the distance to the target. That is, we assume the measurement model

$$\begin{aligned} x_{k,m} &= \|\boldsymbol{p}_k - \boldsymbol{a}_m\|_2 + n_{k,m} \\ &= d_{k,m} + n_{k,m}, \ m = 1, 2, \dots, M, \end{aligned}$$

where $n_{k,m} \sim \mathcal{N}(0, \sigma_{k,m}^2)$ with $\sigma_{k,m}^2 = \sigma^2/d_{k,m}^{-2}$ and σ^2 is the nominal noise variance. We use $\sigma^2 = 2 \times 10^{-5}$.

We consider an area of 60 × 60 square meter with $M = 49$ equally spaced sensors as shown in Fig. 4.1(a). We use the following parameters for simulations [Moon and Stirling, 2000]:

$$\boldsymbol{A}_k = \begin{bmatrix} 1 & 0 & \tau_s & 0 \\ 0 & 1 & 0 & \tau_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ \boldsymbol{\Sigma}_{\boldsymbol{u}} = 10^{-2} \begin{bmatrix} \frac{\tau_s^3}{3} & 0 & \frac{\tau_s^2}{2} & 0 \\ 0 & \frac{\tau_s^3}{3} & 0 & \frac{\tau_s^2}{2} \\ \frac{\tau_s^2}{2} & 0 & \tau_s & 0 \\ 0 & \frac{\tau_s^2}{2} & 0 & \tau_s \end{bmatrix},$$

with sampling time $\tau_s = 2.5$ s. The dynamic model is initialized with $\boldsymbol{p}_0 \sim \mathcal{N}(\mathbf{1}_2, 2.778\boldsymbol{I}_2)$ and $\dot{\boldsymbol{p}}_0 \sim \mathcal{N}(2\mathbf{1}_2, 0.01\boldsymbol{I}_2)$ to emulate a target heading towards the north-east direction. We also run an extended Kalman filter in parallel and it is initialized with $\widehat{\boldsymbol{\theta}}_{0|0} = \mathbf{0}$, and $\boldsymbol{P}_{0|0} = 1000\boldsymbol{I}_2$. For the sake of completeness, we have summarized the extended Kalman filter algorithm in Appendix 4.A. The stochastic matrix $\boldsymbol{B}_k$ is designed such that the transition to the one-hop sensor grid points and staying in the current state takes equal probabilities. In other words, in Fig. 4.1(a), the corner most grid point has 3 one-hop neighbors, hence, it can move to any of these one-hop neighbors each with a probability of 1/4 or stay in the current state with a probability of 1/4. The parameters determining the accuracy are set to $R_e = 25$ cm and $P_e = 0.90$ to compute λ_{eig}.

We do not make velocity measurements, hence, we constrain only the FIM related to the distance measurements. Assuming that the FIM is composed of the following submatrices

$$\boldsymbol{F}_k(\boldsymbol{w}_k, \boldsymbol{\theta}_k) = \begin{bmatrix} \boldsymbol{F}_{k,\boldsymbol{pp}} & \boldsymbol{F}_{k,\boldsymbol{p}\dot{\boldsymbol{p}}} \\ \boldsymbol{F}_{k,\dot{\boldsymbol{p}}\boldsymbol{p}} & \boldsymbol{F}_{k,\dot{\boldsymbol{p}}\dot{\boldsymbol{p}}} \end{bmatrix},$$

then using the Schur complement, the *a posteriori* position estimate $\widehat{\boldsymbol{p}}_{k|k}$ satisfies the following PCRB inequality $\mathbb{E}\{(\widehat{\boldsymbol{p}}_{k|k} - \boldsymbol{p}_k)(\widehat{\boldsymbol{p}}_{k|k} - \boldsymbol{p}_k)^T\} \geq \tilde{\boldsymbol{F}}_k^{-1}(\boldsymbol{w}_k, \boldsymbol{\theta}_k)$, where

$$\tilde{\boldsymbol{F}}_k(\boldsymbol{w}_k, \boldsymbol{\theta}_k) = \boldsymbol{F}_{k,\boldsymbol{pp}} - \boldsymbol{F}_{k,\boldsymbol{p}\dot{\boldsymbol{p}}}\boldsymbol{F}_{k,\dot{\boldsymbol{p}}\dot{\boldsymbol{p}}}^{-1}\boldsymbol{F}_{k,\dot{\boldsymbol{p}}\boldsymbol{p}}$$

assuming $\boldsymbol{F}^{-1}_{k,\dot{\boldsymbol{p}}\dot{\boldsymbol{p}}}$ exists. Similarly, assume that the prior FIM is also composed of the following submatrices

$$\boldsymbol{F}_{\text{prior},k}(\boldsymbol{\theta}_k) = \begin{bmatrix} \boldsymbol{F}_{\text{prior},k,\boldsymbol{p}\boldsymbol{p}} & \boldsymbol{F}_{\text{prior},k,\boldsymbol{p}\dot{\boldsymbol{p}}} \\ \boldsymbol{F}_{\text{prior},k,\dot{\boldsymbol{p}}\boldsymbol{p}} & \boldsymbol{F}_{\text{prior},k,\dot{\boldsymbol{p}}\dot{\boldsymbol{p}}} \end{bmatrix}.$$

Using the Schur complement and some straightforward matrix properties, we can show that

$$\tilde{\boldsymbol{F}}_k(\boldsymbol{w}_k, \boldsymbol{\theta}_k) = \tilde{\boldsymbol{F}}_{\text{prior},k-1} + \sum_{m=1}^{M} w_{k,m}\boldsymbol{F}_{k,m}(\boldsymbol{p}_k) \in \mathbb{R}^{2\times 2},$$

where we define

$$\tilde{\boldsymbol{F}}_{\text{prior},k} = \boldsymbol{F}_{\text{prior},k,\boldsymbol{p}\boldsymbol{p}} - \boldsymbol{F}_{\text{prior},k,\boldsymbol{p}\dot{\boldsymbol{p}}}\boldsymbol{F}^{-1}_{\text{prior},k,\dot{\boldsymbol{p}}\dot{\boldsymbol{p}}}\boldsymbol{F}_{\text{prior},k,\dot{\boldsymbol{p}}\boldsymbol{p}}.$$

Therefore, for the case when only a part of the state parameters are measured, the LMI constraints in (4.10b) should be modified to

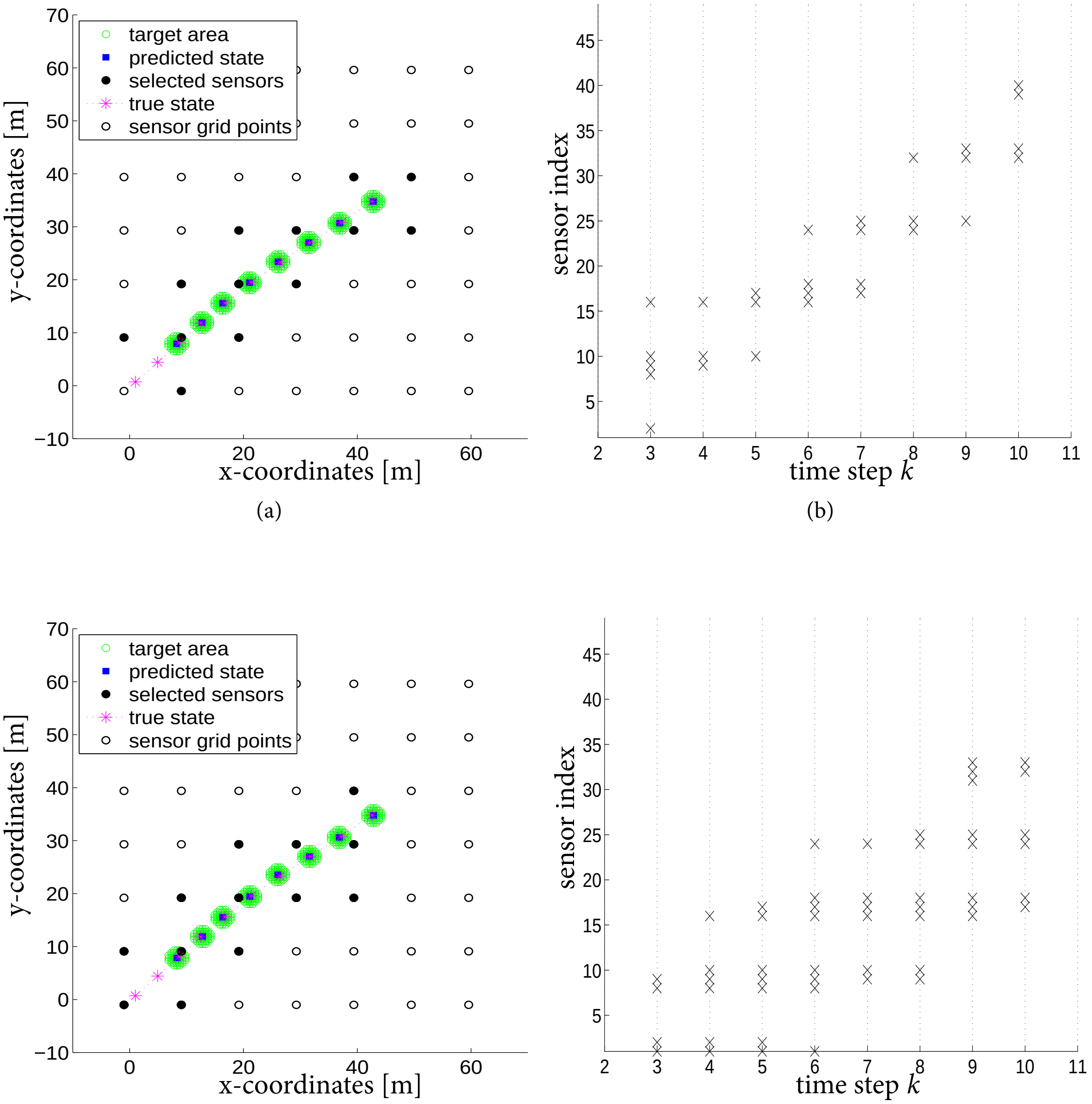
$$\tilde{\boldsymbol{F}}_{\text{prior},k} + \sum_{m=1}^{M} w_{k,m}\boldsymbol{F}_{k,m}(\boldsymbol{p}_k) \succeq \lambda_{\text{eig}}\boldsymbol{I}_2, \ \forall\boldsymbol{\theta}_k \in \mathcal{U}_k.$$

For a practical implementation of the algorithm, we construct $\mathcal{U}_k$ based on the predicted estimate $\widehat{\boldsymbol{\theta}}_{k|k-1}$ and the covariance matrix $\boldsymbol{P}_{k|k-1}$ (see Appendix 4.A). More specifically, $\mathcal{U}_k$ is designed to have points within a circle of radius $5\sqrt{\text{tr}\{\boldsymbol{P}_{k|k-1}\}}$ centered around $\widehat{\boldsymbol{\theta}}_{k|k-1}$. Since $\widehat{\boldsymbol{\theta}}_{k|k-1} \sim \mathcal{N}(\boldsymbol{\theta}_k, \boldsymbol{P}_{k|k-1})$, the *true state* lies within a circle of radius $5\sqrt{\text{tr}\{\boldsymbol{P}_{k|k-1}\}}$ with an overwhelming probability.

Here, we discretize $\mathcal{U}_k$ with 25 points as shown in Fig. 4.1(a) and Fig. 4.1(c) (indicated as the target area in green color). For the sake of easy visibility, we plot the results in the time interval $(3\tau_s, 10\tau_s)$, as the target area is very large for initial estimates. Even though the predicted estimates are not necessarily on top of the true state, the true location will be within the target area with an overwhelming probability. Due to the assumed path-loss model, the sensors close to the target area are selected. The sensor activation time pattern without ($\mu = 0$) and with ($\mu > 0$) smooth sensing is shown in Fig. 4.1(b) and Fig. 4.1(d), respectively. An approximate Boolean solution is obtained by simply rounding all the nonzero entries of $\widehat{\boldsymbol{w}}$ to one. The number of selected sensors with $\mu = 0.5$ is larger as compared to the case with $\mu = 0$. However, the sensors stay active for a longer duration ensuring a smooth hand-off between the selected sensors.

4.7 Dependent observations

So far we have assumed that the observations are mutually independent random variables. If we relax Assumption 4.1, then we can accommodate a much larger class of observation models, for example, to include ambient noises or stochastic state variables. However, by relaxing Assumption 4.1, the FIM information matrix related to the observations $\boldsymbol{F}_{\text{obs},k}(\boldsymbol{w}_k, \boldsymbol{\theta}_k)$, will not be linear in $\boldsymbol{w}_k$, which makes the optimization over $\boldsymbol{w}_k$ much harder (i.e., nonconvex and nonlinear, in general). In this section, we will extend the results developed for sparse sensing for estimation with dependent Gaussian nonlinear measurements in Section 3.7 of Chapter 3 to filtering tasks.



(a) (b)

(c) (d)

Figure 4.1: Sensor scheduling for target tracking based on range measurements for the time interval $(3\tau_s, 10\tau_s)$. (a) and (c) trajectory of the true state for a certain realization, predicted estimate from the EKF, and the target area $\mathcal{U}_k$. (b) and (d) sensor activation time pattern for $\mu = 0$ (without smoothing) and $\mu = 0.5$ (with smoothing), respectively.

Suppose the measurements (4.1) at time instance k are of the form

$$\boldsymbol{x}_k = \boldsymbol{h}_k(\boldsymbol{\theta}_k) + \boldsymbol{n}_k,$$

where $\boldsymbol{h}_k : \mathbb{R}^N \mapsto \mathbb{R}^M$ is a nonlinear functional of the unknown vector $\boldsymbol{\theta}_k$ and $\boldsymbol{n}_k$ is zero-mean Gaussian noise with variance $\boldsymbol{\Sigma}_k \in \mathbb{R}^{M\times M}$. For the above model, the FIM related to the observations (cf.(4.5)) will be

$$\boldsymbol{F}_{\mathrm{obs},k}(\boldsymbol{w}_k, \boldsymbol{\theta}_k) = (\boldsymbol{\Phi}_k(\boldsymbol{w}_k)\boldsymbol{J}_k(\boldsymbol{\theta}_k))^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{w}_k)\,(\boldsymbol{\Phi}_k(\boldsymbol{w}_k)\boldsymbol{J}_k(\boldsymbol{\theta}_k)), \tag{4.15}$$

where $\boldsymbol{J}_k(\boldsymbol{\theta}_k) = \partial \boldsymbol{h}_k(\boldsymbol{\theta}_k)/\partial \boldsymbol{\theta}_k^T \in \mathbb{R}^{M\times N}$ and

$$\boldsymbol{\Sigma}_k(\boldsymbol{w}_k) = \boldsymbol{\Phi}_k(\boldsymbol{w}_k)\boldsymbol{\Sigma}_k\boldsymbol{\Phi}_k^T(\boldsymbol{w}_k) \in \mathbb{R}^{K\times K}$$

is a submatrix of $\boldsymbol{\Sigma}_k$ that includes only the entries corresponding to the selected measurements. The FIM (4.15) is no more additive or linear in $\boldsymbol{w}_k$.

Let us express the noise covariance matrix $\boldsymbol{\Sigma}_k$ as

$$\boldsymbol{\Sigma}_k = a_k\boldsymbol{I}_M + \boldsymbol{S}_k, \tag{4.16}$$

where any scalar $a_k \in \mathbb{R}$ is chosen such that $\boldsymbol{S}_k \in \mathbb{R}^{M\times M}$ is invertible and well-conditioned. Using Property 3.1 from Chapter 3, we can simplify $\boldsymbol{F}_{\mathrm{obs},k}(\boldsymbol{w}_k, \boldsymbol{\theta}_k)$ to

$$\begin{aligned}\boldsymbol{F}_{\mathrm{obs},k}(\boldsymbol{w}_k, \boldsymbol{\theta}_k) &= \boldsymbol{J}_k^T(\boldsymbol{\theta}_k)\boldsymbol{S}_k^{-1}\boldsymbol{J}_k(\boldsymbol{\theta}_k)\\ &\quad - \boldsymbol{J}_k^T(\boldsymbol{\theta}_k)\boldsymbol{S}_k^{-1}\left[\boldsymbol{S}_k^{-1} + a_k^{-1}\mathrm{diag}(\boldsymbol{w}_k)\right]^{-1}\boldsymbol{S}_k^{-1}\boldsymbol{J}_k^T(\boldsymbol{\theta}_k),\end{aligned} \tag{4.17}$$

where in contrast to (4.15), the design parameter $\boldsymbol{w}_k$ appears only once in (4.17), which makes the problem much easier to solve.

Finally, using the Schur complement, the constraint $\lambda_{\min}\{\boldsymbol{F}_k(\boldsymbol{w}_k, \boldsymbol{\theta}_k)\} \geq \lambda_{\mathrm{eig}}$ can now be equivalently expressed under $a_k > 0$ as a size-$(M + N)$ LMI:

$$\begin{bmatrix} \boldsymbol{S}_k^{-1} + a_k^{-1}\mathrm{diag}(\boldsymbol{w}_k) & \boldsymbol{S}_k^{-1}\boldsymbol{J}_k(\boldsymbol{\theta}_k) \\ \boldsymbol{J}_k^T(\boldsymbol{\theta}_k)\boldsymbol{S}_k^{-1} & \boldsymbol{F}_{\mathrm{prior},k-1} + \boldsymbol{J}_k^T(\boldsymbol{\theta})\boldsymbol{S}_k^{-1}\boldsymbol{J}_k(\boldsymbol{\theta}_k) - \lambda_{\mathrm{eig}}\boldsymbol{I}_N \end{bmatrix} \succeq \boldsymbol{0}_{M+N}. \tag{4.18}$$

The constraint (4.18) also depends on the unknown parameter vector $\boldsymbol{\theta}_k$, which otherwise would be independent of $\boldsymbol{\theta}_k$ for additive linear Gaussian measurement models. For the dependent noise case, we also solve (4.10), but by replacing the size-N LMI in (4.10b) with the size-$(M + N)$ LMI (4.18).

4.8 Structured signals

In this section, we will discuss sparse sensing mechanisms for filtering problems involving structured signals (more generally, filtering with equality constraints on the state variables). In particular, we will illustrate sparse sensing for state sequences that are sparse in nature.

Suppose the time-varying vector of interest $\boldsymbol{\theta}_k \in \mathbb{R}^M$ at time k follows a linear model corrupted by additive noise:

$$\boldsymbol{x}_k = \boldsymbol{H}_k\boldsymbol{\theta}_k + \boldsymbol{n}_k, \tag{4.19}$$

where $\boldsymbol{\theta}_k \in \mathbb{R}^M$ (in contrast to the rest of this chapter, without loss of generality, here we assume $M = N$) has just a few nonzero coefficients, i.e., $\|\boldsymbol{\theta}_k\|_0 \ll M$ and $\boldsymbol{H}_k$ denotes some known linear basis of size $M \times M$. The additive noise vector $\boldsymbol{n}_k \in \mathbb{R}^M$ is assumed to be zero-mean Gaussian with covariance matrix $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}_M$.

Under the assumption that the parameter vector $\boldsymbol{\theta}_k$ is sparse, CS theory asserts an exact recovery of $\boldsymbol{\theta}_k$ from observations which are typically much smaller than M, i.e, signals acquired via a linear compression matrix. We now demonstrate that discrete sparse sensing mechanisms can be used for designing time-varying compression matrices as well as determining the optimal compression rate to reach a desired information gain or mean squared error. More specifically, we design $\boldsymbol{\Phi}_k(\boldsymbol{w}_k) \in \{0,1\}^{K \times M}$ to acquire the data as before, i.e.,

$$\boldsymbol{y}_k = \boldsymbol{\Phi}_k(\boldsymbol{w}_k)\boldsymbol{x}_k. \tag{4.20}$$

For $K \ll M$, the sampling matrix will be a compression matrix, and the measurement vector will be much shorter than $\boldsymbol{x}_k$. Note that the sampling matrix $\boldsymbol{\Phi}_k$ and sparsity pattern (including the sparsity order) of the vector $\boldsymbol{\theta}_k$ can both be time-varying.

One way to model the evolution of time-varying sparse sequences (more generally, states with structural constraints) is through the pseudo-measurement formulation [Julier and LaViola, 2007, Carmi et al., 2010, Farahmand et al., 2014], where it is assumed that $\boldsymbol{\theta}_k$ evolves according to the following model

$$\text{dynamics:} \quad \boldsymbol{\theta}_{k+1} = \boldsymbol{A}_k \boldsymbol{\theta}_k + \boldsymbol{u}_k; \tag{4.21a}$$

$$\text{pseudo-measurement:} \quad 0 = g(\boldsymbol{\theta}_k) + e_k, \tag{4.21b}$$

where $\boldsymbol{A}_k$ is an $M \times M$ state-transition matrix, $\boldsymbol{u}_k \in \mathbb{R}^M$ is the process noise, $g(\boldsymbol{\theta}_k)$ is a structure-constraining function, and e_k is zero-mean unit-variance noise. For example, to model sparse states, $g(\cdot)$ can include any one of the well-known approximations of the ℓ_0(-quasi) norm such as the ℓ_1-norm $(\gamma_k \sum_{m=1}^{M} |\theta_{k,m}|)^{1/2}$, the inverse Gaussian function $\left(\gamma_k \sum_{m=1}^{M}(1 - \exp(-\theta_{k,m}^2/2\sigma_g^2))\right)^{1/2}$ with shape parameter σ_g^2, or the sum-of-logarithms function $\left(\gamma_k \sum_{m=1}^{M} \log(|\theta_{k,m}| + \delta)\right)^{1/2}$ with $\delta > 0$. Here, γ_k is the tuning parameter and $\theta_{k,m}$ denotes the mth entry of $\boldsymbol{\theta}_k$. With the square root in each of these functions, the extra least-squares error term for the pseudo-measurement will be equivalent to the sparsity-inducing regularizer. Note that this formulation also accommodates a much richer class of structured signals as discussed later on in Section 4.8.3.

In the considered adaptive sparse sensing problem,we are basically replacing the random measurement operation traditionally used in the CS framework with a deterministic and structured sensing operation, which is more favorable for practical implementation. In what follows, we will develop risk functions that accommodate such structural constraints.

4.8.1 Risk function

The linear system (4.20) and (4.21a) can be solved using the celebrated Kalman filter algorithm [Kay, 1993]. We now recall Remark 4.1, where the posterior FIM (equal to the posterior error covariance matrix) for the state-space equations (4.20) and (4.21a) without the pseudo-measurement was given by (4.7), that is,

$$\boldsymbol{F}(\boldsymbol{w}_k) = (\boldsymbol{\Sigma}_{\boldsymbol{u}} + \boldsymbol{A}_k \boldsymbol{F}_{k-1}^{-1} \boldsymbol{A}_k^T)^{-1} + \frac{1}{\sigma^2} \sum_{m=1}^{M} w_{k,m} \boldsymbol{h}_{k,m} \boldsymbol{h}_{k,m}^T,$$

where $\boldsymbol{h}_{k,m} \in \mathbb{R}^{M \times 1}$, $m = 1, 2, \ldots, M$, are the rows of $\boldsymbol{H}_k$.

Due to the compression ($K \ll M$), the conventional Kalman filter in less meaningful (especially for sparse recovery), unless the inherent sparsity of the state sequence is taken into account. We do this through an (independent) extra pseudo-measurement (4.21b). For the state-space equations (4.20) and (4.21), the additional measurement modifies the posterior FIM to

$$\boldsymbol{F}_k(\boldsymbol{\theta}_k, \boldsymbol{w}_k) = (\boldsymbol{\Sigma}_{\boldsymbol{u}} + \boldsymbol{A}_k \boldsymbol{F}_{k-1}^{-1} \boldsymbol{A}_k^T)^{-1} + \partial g(\boldsymbol{\theta}_k) \partial g(\boldsymbol{\theta}_k)^T + \frac{1}{\sigma^2} \sum_{m=1}^{M} w_{k,m} \boldsymbol{h}_{k,m} \boldsymbol{h}_{k,m}^T, \tag{4.22}$$

where $\partial g(\boldsymbol{\theta}) \in \mathbb{R}^M$ is the (sub)gradient of $g(\boldsymbol{\theta})$ towards $\boldsymbol{\theta}$ evaluated at $\boldsymbol{\theta}_k$. In fact, the above expression is the posterior error covariance of the extended Kalman filter, when the true state $\boldsymbol{\theta}_k$ in (4.22) is replaced with the prediction $\widehat{\boldsymbol{\theta}}_{k|k-1} = \boldsymbol{A}_k \widehat{\boldsymbol{\theta}}_{k-1|k-1}$, where the past estimate can be computed using any of the favorite sparse recovery algorithms or using an iterative extended Kalman filter [Carmi et al., 2010, Farahmand et al., 2014, Vaswani, 2008].

As before, we could use either one of the constraints discussed in Section 4.3, i.e., use $f(\boldsymbol{w}_k) := \operatorname{tr}\{(\boldsymbol{F}_k(\boldsymbol{\theta}_k, \boldsymbol{w}_k))^{-1}\} \leq \lambda_{\text{tr}}$ or $f(\boldsymbol{w}_k) := \lambda_{\min}\{\boldsymbol{F}_k(\boldsymbol{\theta}_k, \boldsymbol{w}_k)\} \geq \lambda_{\text{eig}}$ with $\boldsymbol{F}_k(\boldsymbol{\theta}_k, \boldsymbol{w}_k)$ in (4.22) in place of (4.12c) to solve (4.12).

4.8.2 Example: CS-based target tracking

In this section, we illustrate the developed theory with the following target tracking example. Let $\boldsymbol{p}_k = [p_{k,x}, p_{k,y}]^T \in \mathbb{R}^2$ denote the position of the target at time instance k and $\boldsymbol{a}_m \in \mathbb{R}^2$ denote the position of the mth sensor. Let us assume that there are M such locations where we can place these sensors. The sensors are capable of measuring the signal strength as

$$h_{k,m}(\boldsymbol{p}_k) = \frac{\beta s}{\beta + \|\boldsymbol{p}_k - \boldsymbol{a}_m\|_2^2}, m = 1, 2, \ldots, M, \tag{4.23}$$

with a constant $\beta > 0$. Here, s denotes the signal strength. We linearize (4.23) around M grid points $\{\boldsymbol{g}_m\}_{m=1}^M$, where the target could be potentially located. As a result, we arrive at the linear grid-based model given by

$$x_{k,m} = \boldsymbol{h}_{k,m}^T \boldsymbol{\theta}_k + n_{k,m}, m = 1, 2, \ldots, M,$$

where $\boldsymbol{h}_{k,m} = [h_{k,m}(\boldsymbol{g}_1), h_{k,m}(\boldsymbol{g}_2), \ldots, h_{k,m}(\boldsymbol{g}_M)]^T \in \mathbb{R}^M$ is time-invariant, but $\boldsymbol{\theta}_k$ is time-varying. All the entries of the vector $\boldsymbol{\theta}_k \in \mathbb{R}^M$ are equal to zero except for the mth entry, $\theta_{k,m}$, which is equal to the target signal strength s at time k if and only if the target is located at the mth grid point, i.e., for $\boldsymbol{p}_k = \boldsymbol{g}_m$. Note that the number of grid points can be much larger than the number of (selected) sensors.

We are interested in tracking a target moving along the grid points $\boldsymbol{g}_m = [m, m]^T$ for $m = 0, 1, \ldots, 29$ as shown in Fig. 4.2(a). This can be modeled as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{A}\boldsymbol{\theta}_k + \boldsymbol{u}_k,$$

where the entries of the initial vector $\boldsymbol{\theta}_0$ are all zero except for the first entry $\theta_{1,1} = s$. Here, the state-transition matrix is a shift matrix, i.e.,

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{M \times M}.$$

In scenarios like the one considered here, compression via random linear projections would still need all the M sensors with no reduction in the sensing and communications cost. On the contrary, sparse sensing enables a completely decentralized sensing, and it needs only $K \ll M$ sensors.

In this example, the sparsity pattern is time-varying, but the sparsity order is fixed. We stress here that the proposed framework is not limited to signals with a fixed sparsity order. We use the following parameters in the simulations: The number of grid points/candidate sensors $M = 30$, $K = 5$, $\beta = 100$, $s = 10$, and $\sigma = 10^{-3}$. The sensors are deployed uniformly at random within a 30×30 m^2 surveillance area as shown in Fig. 4.2(a).

We use $f_k(\boldsymbol{w}_k) := \operatorname{tr}\{(\boldsymbol{F}_k(\boldsymbol{\theta}_k, \boldsymbol{w}_k))^{-1}\}$ and λ_{tr} is chosen such that $N = 5$ sensors are selected at each time step. We evaluate $\boldsymbol{F}_k(\boldsymbol{\theta}_k, \boldsymbol{w}_k)$ at the prediction $\widehat{\boldsymbol{\theta}}_{k|k-1} = \boldsymbol{A}\widehat{\boldsymbol{\theta}}_{k-1|k-1}$, where the past estimate is obtained by solving the following ℓ_1-norm regularized least squares:

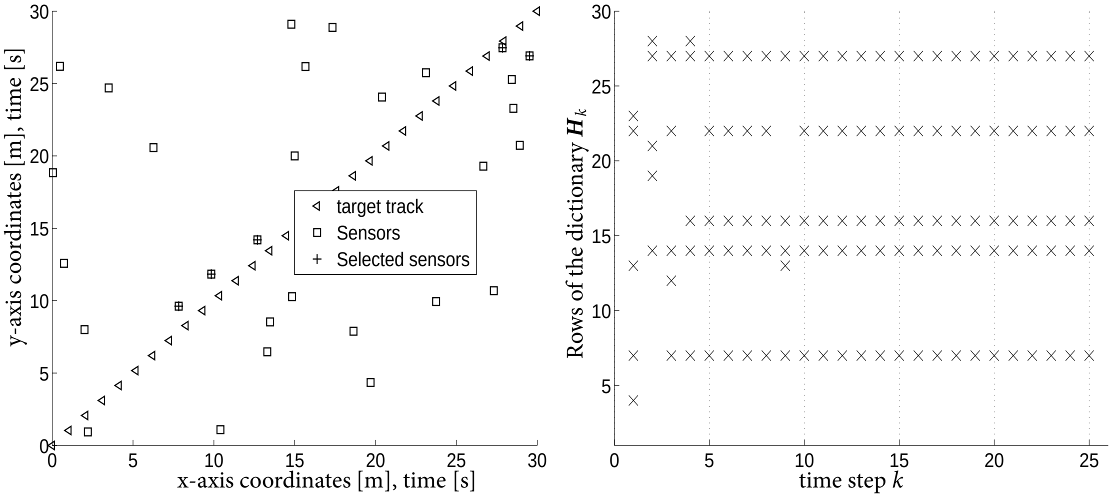
$$\begin{aligned} \widehat{\boldsymbol{\theta}}_{k|k} = \arg\min_{\boldsymbol{\theta}_k} \quad & \|\widehat{\boldsymbol{\theta}}_{k|k-1} - \boldsymbol{\theta}_k\|^2_{\boldsymbol{P}^{-1}_{k|k-1}} \\ & + \|\boldsymbol{y}_k - \boldsymbol{\Phi}_k(\boldsymbol{w}_k)\boldsymbol{H}\boldsymbol{\theta}_k\|^2_{\boldsymbol{\Sigma}^{-1}} + g^2(\boldsymbol{\theta}_k), \end{aligned} \tag{4.24}$$

with $g(\boldsymbol{\theta}_k) = \sqrt{2\gamma_k \|\boldsymbol{\theta}_k\|_1}$ (here, $\|\boldsymbol{\theta}_k\|_1 = \mathbf{1}^T\boldsymbol{\theta}_k$ as the entries of $\boldsymbol{\theta}_k$ are all nonnegative), $\boldsymbol{P}_{k|k-1} = \boldsymbol{A}\boldsymbol{P}_{k-1|k-1}\boldsymbol{A}^T + \boldsymbol{\Sigma}_{\boldsymbol{u}}$, and

$$\begin{aligned} \boldsymbol{P}_{k|k} = \Bigg(& \boldsymbol{P}^{-1}_{k|k-1} + \frac{\gamma_k}{2\|\boldsymbol{\theta}_k\|_1}\mathbf{1}\mathbf{1}^T \\ & + \frac{1}{\sigma^2}\sum_{m=1}^{M} \boldsymbol{h}_{k,m}\boldsymbol{h}_{k,m}^T \Bigg)^{-1}. \end{aligned}$$

Here, γ_k is a tuning parameter. We initialize $\widehat{\boldsymbol{\theta}}_{0|0} = \mathbf{1}_M$, $\boldsymbol{P}_{0|0} = \boldsymbol{I}_M$, $\boldsymbol{\Sigma}_{\boldsymbol{u}} = 0.01\boldsymbol{I}_M$, and we compute γ_k using the method described in [Farahmand et al., 2014].

The relaxed optimization problem is solved using SeDuMi [Sturm, 1999]. Fig. 4.2(b) illustrates the solution path for $k = 1, 2, \ldots, 25$ s. The Boolean solution is recovered using deterministic rounding. The sensors selected for time step $k = 25$ s are also shown in Fig. 4.2(a). In this example, the same subset of sensors are selected for $k > 10$ because the matrices $\boldsymbol{H}$ and $\boldsymbol{A}$, and the sparsity order are not changing with time. In other words, the selected sensors optimize the average error, averaged over the entire track.



(a) (b)

Figure 4.2: Tracking a target using a grid-based model with $M = 30$ and $K = 5$. (a) A target is moving along the straight line $p_{x,k} = p_{y,k} = k$, i.e., it moves with a constant velocity of $\sqrt{2}$ m/s. The selected sensors shown correspond to $k = 25$ s. (b) The solution path illustrating the selected rows of the dictionary $\boldsymbol{H}_k$ for $k = 1, 2, \dots, 25$ s.

4.8.3 Extensions to other structured signals

In this section, we highlight some important generalizations of the proposed framework for structured signals, which are often studied together with the CS framework. The sparsity prior can be extended to a much broader class of structured signals, including structured sparse signals (or block-sparse signals) [Friedman et al., 2010], smooth (i.e., sparsity of the coefficients and also sparsity of their differences) [Tibshirani et al., 2005], to list a few. Depending on the structure of the state, the $g(\boldsymbol{\theta}_k)$ has to be modified accordingly. More specifically, for structured sparse signals we use a regularizer that accounts for block sparsity, i.e.,

$$g(\boldsymbol{\theta}_k) := \left(\gamma_k \sum_{i=1}^{G} \|\boldsymbol{\theta}_{k,i}\|_2\right)^{1/2},$$

where the state vector $\boldsymbol{\theta}_k$ is grouped into G subvectors each of length N/G as

$$\boldsymbol{\theta}_k = [\boldsymbol{\theta}_{k,1}^T, \boldsymbol{\theta}_{k,2}^T, \dots, \boldsymbol{\theta}_{k,G}^T]^T.$$

Similarly, for signal smoothness, we use the regularizer

$$g(\boldsymbol{\theta}_k) := \left(\gamma_{k,1}\|\boldsymbol{\theta}_k\|_1 + \gamma_{k,2} \sum_{m=1}^{M-1} |\theta_{k,m} - \theta_{k,m-1}|\right)^{1/2},$$

where $\gamma_{k,1}$ and $\gamma_{k,2}$ are tuning parameters.

4.9 Discussion

In this chapter we have discussed discrete sparse sensing for state estimation in nonlinear dynamical systems. In particular, we have focused on filtering problems of three types: independent observations, additive Gaussian dependent observations, and structured state sequences. For designing sparse sensing patterns, we have used scalar measures related to the posterior Cramér-Rao bound, which depend on the previous as well as the current true states. The sensing patterns are designed one step ahead in time, hence can be computed offline.

4.A Extended Kalman filter

For the sake of completeness, we provide the steps involved in the Extended Kalman filter, which is used for state estimation with nonlinear additive Gaussian models; see [Kay, 1993] for more details.

Given the state-space equations

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{h}_k(\boldsymbol{\theta}_k) + \boldsymbol{n}_k, \\ \boldsymbol{\theta}_{k+1} &= \boldsymbol{A}_k \boldsymbol{\theta}_k + \boldsymbol{u}_k, \end{aligned}$$

where $\boldsymbol{h}_k(\cdot) : \mathbb{R}^N \mapsto \mathbb{R}^M$ is a nonlinear function of the unknown state parameter vector $\boldsymbol{\theta}_k \in \mathbb{R}^N$, $\boldsymbol{A}_k \in \mathbb{R}^{N \times N}$ is the state-transition matrix, and $\boldsymbol{n}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ and $\boldsymbol{u}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{u}})$ are the measurement and process noise vectors of length $M \times 1$, respectively.

Suppose that the estimate $\widehat{\boldsymbol{\theta}}_{k-1|k-1}$ and its covariance matrix $\mathbf{P}_{k-1|k-1}$ are available from the previous time step. At current time step k, the prediction and its covariance are given as

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_{k|k-1} &= \boldsymbol{A}_k \widehat{\boldsymbol{\theta}}_{k-1|k-1}, \\ \boldsymbol{P}_{k|k-1} &= \boldsymbol{A}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{A}_k^T + \boldsymbol{\Sigma}_{\boldsymbol{u}}. \end{aligned}$$

The update equations include the posterior estimate and posterior covariance matrices, which are computed using:

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_{k|k} &= \widehat{\boldsymbol{\theta}}_{k|k-1} + \boldsymbol{P}_{k|k-1} \boldsymbol{H}_{k|k-1}^T (\boldsymbol{H}_{k|k-1} \boldsymbol{P}_{k|k-1} \boldsymbol{H}_{k|k-1}^T + \boldsymbol{\Sigma})^{-1} (\boldsymbol{x}_k - \boldsymbol{h}_k(\widehat{\boldsymbol{\theta}}_{k|k-1})), \\ \boldsymbol{P}_{k|k} &= \boldsymbol{P}_{k|k-1} - \boldsymbol{P}_{k|k-1} \boldsymbol{H}_{k|k-1}^T (\boldsymbol{H}_{k|k-1} \boldsymbol{P}_{k|k-1} \boldsymbol{H}_{k|k-1}^T + \boldsymbol{\Sigma})^{-1} \boldsymbol{H}_{k|k-1} \boldsymbol{P}_{k|k-1}, \end{aligned}$$

where $\boldsymbol{h}_k(\cdot)$ is linearized around the prediction to obtain

$$\boldsymbol{H}_{k|k-1} = \left. \frac{\partial \boldsymbol{h}_k(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \right|_{\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}}_{k|k-1}} \in \mathbb{R}^{M \times N}.$$

Part of this chapter is under review for publication as: S.P. Chepuri and G. Leus. Sparse Sensing for Distributed Detection. *In Subm. to IEEE Trans. on Signal Processing.*

Chapter 5

Sparse Sensing for Detection

Contents

5.1 Introduction

In this chapter, we will study the discrete sparse sensing framework for another important statistical inference problem, that is, detection. Statistical detection is pertinent to applications in sensor networks, radar and sonar systems, wireless cognitive radio networks, biometrics, social networks, imaging platforms, to list a few. We assume that the field is sampled by (spatially or temporally) distributed sensors, and these samples are delivered to a central unit. The central unit then makes a single global decision as to the true hypothesis using binary hypothesis testing. More specifically, the observations at each sensor are related to the state of nature $\mathcal{H}$, where the random variable $\mathcal{H}$ is drawn from a binary alphabet set $\{\mathcal{H}_0, \mathcal{H}_1\}$. In the Bayesian setting, we assume that the prior probabilities $\pi_0 = \Pr(\mathcal{H}_0)$ and $\pi_1 = \Pr(\mathcal{H}_1)$ are known, whereas in the Neyman-Pearson setting, the prior probabilities are not known.

5.1.1 Related earlier works

The minimum error probability criterion is a standard performance measure for design problems related to statistical detection such as signal design [Grettenberg, 1963, Kadota and Shepp, 1967, Kailath, 1967], censoring [Rago et al., 1996], sampling design [Yu and Varshney, 1997], and so on. However, in most cases, optimizing the error probabilities is very difficult. This may be because these error probabilities do not admit a known closed form or their expression is too complicated for numerical optimization. Therefore, weaker performance criteria that are easier to evaluate and optimize are often used. A number of measures related to the distance between the conditional probabilities are widely used in the design of experiments as proxies for the error probability [Kailath, 1967,Rago et al., 1996,Yu and Varshney, 1997,Bajovic et al., 2011,Chamberland and Veeravalli, 2007]. Some of the prominent distance measures that are often used are the Kullback-Leibler distance, J-divergence, Chernoff information, and Bhattacharyya distance.

A related topic in the context of energy-efficient distributed detection is *data censoring*, wherein the uninformative sensor observations are not transmitted to the central unit [Rago et al., 1996, Appadwedula et al., 2008, Blum and Sadler, 2008]. However, in censoring, data still has to be acquired in order to choose informative sensors, thus, it incurs a sensing cost. That is, censoring schemes are data dependent as opposed to the proposed data-independent sparse sensing schemes that can be designed offline. In other words, the actual measurements are not needed and only model information is used.

5.1.2 Main results

We focus on both the Bayesian as well as the Neyman-Pearson setting for binary hypothesis testing. The sparse sensing operation is designed based on a number of distance measures that belong to the general class of Ali-Silvey distances [Ali and Silvey, 1966].

The main question addressed in the chapter is similar to that of [Cambanis and Masry, 1983, Bahr and Bucklew, 1990, Yu and Varshney, 1997, Bajovic et al., 2011, Quan et al., 2009, Sung et al., 2005], but with the following differences. Firstly, the proposed framework is *general*, that is, it is not limited to Gaussian observations, especially for conditionally independent observations. Secondly, we propose a sparsity-promoting cost function to design structured samplers to achieve the lowest sensing cost as compared to the previously adopted periodic, regular, or random samplers. The main contributions of this chapter that broaden the existing literature are listed below.

- For conditionally independent observations, the best subset of sensors is the one with the *smallest local average root-likelihood ratio* and *largest local average log-likelihood ratio* in the

Bayesian and Neyman-Pearson setting, respectively. This leads to an explicit solution for the sensing design problem that is optimal in terms of the error exponents. As a special case, for Gaussian observations with common covariances and uncommon means under both hypotheses, the selected sensors are also optimal in terms of the error probabilities (initial results for the Gaussian case were reported in [Chepuri and Leus, 2015a]). The computational complexity of the proposed solvers is independent of the number of candidate sensors, and is as low as $O(K)$, where K is the number of selected sensors (or sampling locations).

- For conditionally dependent observations, we focus on the Gaussian setting. When the mean vectors are uncommon and covariance structure is common under both hypotheses, the sensing design problem can be relaxed to a convex optimization problem. Although this leads to a suboptimal solution, we propose a randomized rounding technique that leads to a near-optimal solution. Furthermore, in this case, for nonidentical sensor observations, we show that the number of sensors required to achieve a prescribed detection performance decreases significantly as the correlation among them increases (i.e., when the sensors become more coherent), which is in complete contrast to the case of identical sensor observations. When the covariances are uncommon and means are common under both hypotheses, the sensing design problem remains nonconvex, except for the J-divergence optimization.

5.2 Sensing conditionally distributed observations

Consider a network with M candidate sensors. These candidate sensors might represent temporal, spatial, or even spatio-temporal samples. The observations are related to the following model

$$\mathcal{H}_0: \quad x_m \sim p_m(x|\mathcal{H}_0),\ m = 1, 2, \dots, M, \tag{5.1a}$$

$$\mathcal{H}_1: \quad x_m \sim p_m(x|\mathcal{H}_1),\ m = 1, 2, \dots, M, \tag{5.1b}$$

where the probability density function (pdf) of the observation at the mth sensor, x_m, conditioned on the state of nature $\mathcal{H}$ is denoted by $p_m(x|\,\mathcal{H}_i)$ for $i = 0, 1$. Further, the observations are collected in $\boldsymbol{x} = [x_1, x_2, \dots, x_M]^T \in \mathbb{R}^M$. The pdf of $\boldsymbol{x}$ under $\mathcal{H}_0$ and $\mathcal{H}_1$ is denoted by $p(\boldsymbol{x}|\,\mathcal{H}_0)$ and $p(\boldsymbol{x}|\,\mathcal{H}_1)$, respectively.

We acquire the data $\boldsymbol{x}$ via a linear sensing operation, where the sensing task is modeled through a vector whose entries belong to a binary alphabet, i.e., through

$$\boldsymbol{w} = [w_1, w_2, \dots, w_M]^T \in \{0, 1\}^M,$$

where the variable $w_m = (0)1$ indicates whether the mth sensor is (not) selected. More specifically, we define the sensing matrix $\boldsymbol{\Phi}(\boldsymbol{w}) = \text{diag}_\text{r}(\boldsymbol{w}) \in \{0, 1\}^{K \times M}$, to acquire the data as

$$\boldsymbol{y} = \text{diag}_\text{r}(\boldsymbol{w})\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{w})\boldsymbol{x},$$

where K is not assumed to be known. Note that we are interested in cases where $K \ll M$. The reduced dimension data vector $\boldsymbol{y} \in \mathbb{R}^K$ is used instead of $\boldsymbol{x} \in \mathbb{R}^M$ to solve the detection problem. In this chapter, we seek a sparsest $\boldsymbol{w}$, i.e., a vector with many zeros and just a few nonzero entries, such that a prescribed global detection performance is achieved.

5.3 Problem statement

Let $\widehat{\mathcal{H}}$ denote an estimate of the state of nature $\mathcal{H}$, based on a certain decision rule. In the Neyman-Pearson setting, the optimal detector is the well-known Neyman-Pearson detector that minimizes the probability of miss detection (type II error),

$$P_m = \Pr(\widehat{\mathcal{H}} \neq \mathcal{H}_1 | \mathcal{H}_1)$$

for a fixed probability of false alarm (type I error),

$$P_f = \Pr(\widehat{\mathcal{H}} = \mathcal{H}_1 | \mathcal{H}_0).$$

In the Bayesian setting, given the prior probabilities $\pi_i = \Pr(\mathcal{H}_i)$ for $i = 0, 1$, the optimal detector minimizes the Bayesian error probability,

$$P_e = \Pr(\widehat{\mathcal{H}} \neq \mathcal{H}) = \pi_0 P_f + \pi_1 P_m,$$

or more generally, the detector minimizes the Bayes' risk. Having introduced the data model, we now formally state the design problem of interest.

Problem 5.1 (Sparse sampler design)**.** *Given the data model* (5.1)*, design a sparsest Boolean vector* $\mathbf{w}$ *that results in a prescribed*

(i) *Bayesian probability of error, P_e, in the Bayesian setting, or*

(ii) *probability of miss detection, P_m, for a fixed probability of false alarm, P_f, in the Neyman-Pearson setting.*

Mathematically, the sparse sensing problem for distributed detection can be formulated as

$$\begin{aligned} \text{P-B} : &\arg\min_{\boldsymbol{w}\in\{0,1\}^M} \|\boldsymbol{w}\|_0 \\ &\quad \text{s.to} \quad P_e(\boldsymbol{w}) \leq e; \end{aligned} \tag{5.2a}$$

$$\begin{aligned} \text{P-N} : &\arg\min_{\boldsymbol{w}\in\{0,1\}^M} \|\boldsymbol{w}\|_0 \\ &\quad \text{s.to} \quad P_f(\boldsymbol{w}) \leq \alpha, \text{ and } P_m(\boldsymbol{w}) \leq \beta, \end{aligned} \tag{5.2b}$$

where e, α and β are, respectively, the desired Bayesian probability of error, maximum false-alarm rate and miss-detection rate. Here, $P_e(\boldsymbol{w})$, $P_f(\boldsymbol{w})$, and $P_m(\boldsymbol{w})$ denote the error probabilities due to the selected sensor subset indicated by the nonzero entries of $\boldsymbol{w}$. When prior probabilities are available, we solve P-B (P denotes problem and B denotes Bayesian), otherwise in the Neyman-Pearson setting we solve P-N (N denotes Neyman-Pearson).

In order to ease the design, we next discuss some performance measures that can substitute the error probabilities in the above optimization problems.

5.4 $f(\boldsymbol{w})$ for detection

The error probabilities P_e, P_m or P_f might not admit a known closed-form expression or their expressions might not be favorable for numerical optimization. In this section, we will discuss several weaker and simpler substitutes, which can be optimized instead of the error probabilities. These substitutes are based on the notion of distance (closeness or divergence) between the two distributions of the observations under test. They lead to tractable, if not always optimal (in terms of the error probabilities), design procedures for sampler design. Nevertheless, optimizing the distance measures improves the performance of any practical system.

Let the likelihood ratio of the two hypotheses under test be defined as

$$l(\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathcal{H}_1)}{p(\boldsymbol{y}|\mathcal{H}_0)}.$$

In what follows, we consider a number of distance measures that belong to the general class of Ali-Silvey distances [Ali and Silvey, 1966], which are of the form

$$\psi\left(\mathbb{E}_{|\mathcal{H}_i}\{\phi\left[l(\boldsymbol{y})\right]\}\right),$$

where $\psi(\cdot)$ is an increasing real-valued function, $\phi[\cdot]$ is a continuous convex function on $(0, \infty)$, and the notation $\mathbb{E}_{|\mathcal{H}_i}\{\phi\left[l(\boldsymbol{y})\right]\}$ indicates that $\phi\left[l(\boldsymbol{y})\right]$ is averaged under the pdf $p(\boldsymbol{y}|\mathcal{H}_i)$ for either $i = 0$ or $i = 1$.

5.4.1 The Bayesian setting

The Bayes detector minimizes P_e, and makes a decision based on comparing the optimal statistic to a threshold:

$$\log l(\boldsymbol{y}) = \log \frac{p(\boldsymbol{y}|\mathcal{H}_1)}{p(\boldsymbol{y}|\mathcal{H}_0)} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\lesseqgtr}} \log \frac{\pi_0}{\pi_1}.$$

In the Bayesian setting, our goal is to choose the best subset of sensors that results in a prescribed Bayesian probability of error P_e. The best achievable exponent in the Bayesian probability of error is parameterized by the Chernoff information (sometimes also referred to as the Chernoff distance) [Cover and Thomas, 2012, Chernoff's theorem], and it is given by

$$\begin{aligned} \mathcal{C}(\mathcal{H}_1\|\mathcal{H}_0) &= -\log \min_{0\le n\le 1} \int [p(\boldsymbol{y}|\mathcal{H}_1)]^n [p(\boldsymbol{y}|\mathcal{H}_0)]^{1-n} d\boldsymbol{y} \\ &= -\log \min_{0\le n\le 1} \mathbb{E}_{|\mathcal{H}_0}\{[l(\boldsymbol{y})]^n\}. \end{aligned} \tag{5.3}$$

Due to the involved minimization over n, the Chernoff information in (5.3) is difficult to optimize over $\boldsymbol{w}$. Therefore, we use a special case of the Chernoff information called the *Bhattacharyya distance* as the optimization criterion, where the Bhattacharyya distance is obtained by fixing $n = 0.5$ in (5.3). The Bhattacharyya distance is given by

$$\mathcal{B}(\mathcal{H}_1\|\mathcal{H}_0) = -\log \rho, \tag{5.4}$$

where the *Bhattacharyya coefficient* [Kailath, 1967] or the *Hellinger integral* [Kadota and Shepp, 1967], ρ, is given by

$$\begin{aligned} \rho &= \int \sqrt{p(\boldsymbol{y}|\mathcal{H}_1)p(\boldsymbol{y}|\mathcal{H}_0)}d\boldsymbol{y} = \int p(\boldsymbol{y}|\mathcal{H}_0)\sqrt{\frac{p(\boldsymbol{y}|\mathcal{H}_1)}{p(\boldsymbol{y}|\mathcal{H}_0)}}d\boldsymbol{y} \\ &= \mathbb{E}_{|\mathcal{H}_0}\left\{\sqrt{l(\boldsymbol{y})}\right\}. \end{aligned} \tag{5.5}$$

It is easy to verify from (5.5) that the Bhattacharya distance is symmetric, which means $\mathcal{B}(\mathcal{H}_1\|\mathcal{H}_0) = \mathcal{B}(\mathcal{H}_0\|\mathcal{H}_1)$. More importantly, upper and lower bounds for the Bayesian probability of error can be obtained using the Bhattacharyya coefficient. The bounds are given as follows [Kadota and Shepp, 1967, Appendix A], [Kailath, 1967]:

$$\frac{1}{2}\min(\pi_0, \pi_1)\rho^2 \leq P_e \leq \sqrt{\pi_0\pi_1}\rho. \tag{5.6}$$

Therefore, in place of the Bayesian error probability, we minimize the Hellinger integral, or equivalently, maximize the Bhattacharyya distance.

Furthermore, when $\int [p(\boldsymbol{y}|\mathcal{H}_1)]^n[p(\boldsymbol{y}|\mathcal{H}_0)]^{1-n}d\boldsymbol{y}$ is symmetric in n and the observations are independent and identically distributed, the Bhattacharyya distance is exponentially the best [Kailath, 1967], i.e.,

$$P_e \overset{a.s.}{=} \exp\left(-\mathcal{B}(\mathcal{H}_1\|\mathcal{H}_0)\right) \quad \text{for} \quad P_e \to 0.$$

We now introduce the following assumption:

Assumption 5.1 (Conditional independence)**.** *The sensor observations are statistically independent, conditioned on the hypothesis $\mathcal{H}$.*

Under Assumption 5.1, the likelihood ratio simplifies to

$$l(\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mathcal{H}_1)}{p(\boldsymbol{y}|\mathcal{H}_0)} = \prod_{m=1}^{M}[l_m(x)]^{w_m}$$

where $l_m(x) = p_m(x|\mathcal{H}_1)/p_m(x|\mathcal{H}_0)$ is the local likelihood ratio related to the mth sensor, and $p_m(x|\mathcal{H}_i)$ for $i = 0, 1$ are the conditional pdfs of x for the mth sensor. Here, the conditional pdf of the selected sensors is of reduced dimension, i.e., it does not include the measurements that are set to zero.

Besides being a reasonable measure, the Bhattacharya distance is much simpler to optimize under Assumption 5.1 because of the following result:

Proposition 5.1 (Linearity of the Bhattacharyya distance)**.** *The considered sparse sensing mechanism preserves the additivity of the Bhattarcharyya distance under Assumption 5.1, i.e., we can express*

$$f_{\mathrm{B}}(\boldsymbol{w}) := \mathcal{B}(\mathcal{H}_1\|\mathcal{H}_0) = \sum_{m=1}^{M} w_m\mathcal{B}_m(\mathcal{H}_1\|\mathcal{H}_0), \tag{5.7}$$

where

$$\mathcal{B}_m(\mathcal{H}_1\|\mathcal{H}_0) = -\log \mathbb{E}_{|\mathcal{H}_0}\{\sqrt{l_m(x)}\}. \tag{5.8}$$

Proof. See Appendix 5.A □

Thus, Proposition 5.1 enables us to optimize $f_{\mathrm{B}}(\boldsymbol{w})$ over $\boldsymbol{w}$ (subscript B denotes Bayesian). We underline here that $f_{\mathrm{B}}(\boldsymbol{w})$ assumes only the knowledge of the data model and does not need actual measurements, hence the sensing operation can be designed offline. We also remark that the Chernoff information (5.3) is not additive for conditionally independent observations, unlike the Bhattacharyya distance.

Before discussing the optimization criterion for the Neyman-Pearson setting, we end this subsection with the following remark that generalizes the sampling design in the Bayesian setting.

Remark 5.1 (Bayes risk). *Let C_{ij} be the cost if we decide $\mathcal{H}_i$ when $\mathcal{H}_j$ is true. A generalization of the minimum P_e detector, is to minimize the Bayes risk*

$$\mathcal{R} = \sum_{i=0}^{1}\sum_{j=0}^{1} \Pr(\mathcal{H}_i|\mathcal{H}_j)\Pr(\mathcal{H}_j),$$

where we arrive at a special case of $\mathcal{R} = P_e$ for $C_{00} = C_{11} = 0, C_{10} = C_{01} = 1$. This results in the sensing design problem

$$\operatorname*{arg\,min}_{\boldsymbol{w}\in\{0,1\}^M} \|\boldsymbol{w}\|_0 \quad \text{s.to} \quad \mathcal{R}(\boldsymbol{w}) \le e_r,$$

where $\mathcal{R}(\boldsymbol{w})$ denotes the Bayes risk due to the selected sensor subset indicated by the nonzero entries of $\boldsymbol{w}$, and e_r is the desired Bayes risk.

The bounds in (5.6) *can be generalized to [Kobayashi and Thomas, 1967]*

$$\mathcal{R}_0 + \mathcal{R}_2\rho^2 \le \mathcal{R} \le \mathcal{R}_0 + \sqrt{\mathcal{R}_1}\rho,$$

where $\mathcal{R}_0 = \pi_0 C_{00} + \pi_1 C_{11}$, $\mathcal{R}_1 = \pi_0\pi_1(C_{11} - C_{01})(C_{00} - C_{10})$, and $\mathcal{R}_2 = \mathcal{R}_1/(\pi_0(C_{00} - C_{10}) + \pi_1(C_{11} - C_{01}))$. Therefore, maximizing the Bhattacharyya distance (or minimizing the Hellinger integral) is a reasonable optimality criterion also for a more general minimum Bayes risk detector.

5.4.2 The Neyman-Pearson setting

When the prior probabilities are not known, we solve the Neyman-Pearson problem, where one of the error probabilities (P_f, for example) is fixed while the second error probability, P_m is minimized. More specifically, the decision is based upon the log-likelihood ratio test

$$\log l(\boldsymbol{y}) = \log \frac{p(\boldsymbol{y}|\mathcal{H}_1)}{p(\boldsymbol{y}|\mathcal{H}_0)} \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\lessgtr}} \gamma, \tag{5.9}$$

where γ is the threshold obtained by setting $P_f = \alpha$. In what follows, we discuss two distance measures that we can optimize in the Neyman-Pearson setting.

Kullback-Leibler distance

For a Neyman-Pearson problem, the best achievable error exponent in the probability of error (P_m, for example) is given by the relative entropy or *Kullback-Leibler distance* $\mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0)$ [Cover and Thomas, 2012, Stein's lemma]. That is, for a fixed value of P_f,

$$\log P_m \overset{as.}{=} -\mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0) \quad \text{for} \quad P_m \to 0.$$

The Kullback-Leibler distance is the average log-likelihood ratio, and is given by [Kullback, 2012]

$$\begin{aligned}\mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0) &= \mathbb{E}_{|\mathcal{H}_1}\{\log l(\boldsymbol{y})\}\\ &= \int \log l(\boldsymbol{y}) p(\boldsymbol{y}|\mathcal{H}_1) d\boldsymbol{y}.\end{aligned} \tag{5.10}$$

A lower bound on P_m for a fixed P_f, say α $(0 \leq \alpha \leq 1)$ can be obtained using [Kullback, 2012, pp. 74-75 and tables in pp. 378-379]

$$\begin{aligned}\mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0) \geq \alpha \log\left(\frac{\alpha}{1-P_m}\right)\\ + (1-\alpha)\log\left(\frac{1-\alpha}{P_m}\right) = g(P_m).\end{aligned} \tag{5.11}$$

Since $g(P_m)$ is a strictly monotonic function, we can write

$$P_m \geq g^{-1}\big(\mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0)\big). \tag{5.12}$$

For example, a very small (close to zero) α simplifies (5.12) to $P_m \geq \exp\big(-\mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0)\big)$. The following theorem gives an upper bound on P_m.

Theorem 5.1 (Upper bound on P_m)**.** *If the variance of the log-likehood ratio is v^2, then*

$$P_m \leq \frac{1}{1 + \frac{(\mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0) - \log \gamma)^2}{v^2}}, \tag{5.13}$$

where the threshold γ corresponds to a desired $P_f = \alpha$.

Proof. See Appendix 5.B. □

Therefore, the bounds in (5.12) and (5.13) make the maximization of $\mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0)$ a reasonable optimality criterion. We stress here that the above bounds (5.12) and (5.13) are valid even when Assumption 5.1 is not true.

The following property of the Kullback-Leibler distance further allows its easy numerical optimization.

Proposition 5.2 (Linearity of the Kullback-Leibler distance)**.** *The considered sparse sensing mechanism preserves the additivity of the Kullback-Leibler distance under Assumption 5.1, i.e., we can express*

$$f_{\mathrm{N},1}(\boldsymbol{w}) := \mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0) = \sum_{m=1}^{M} w_m \mathcal{K}_m(\mathcal{H}_1\|\mathcal{H}_0), \tag{5.14}$$

where

$$\begin{aligned}\mathcal{K}_m(\mathcal{H}_1\|\mathcal{H}_0) &= \int \log l_m(x) p_m(x|\mathcal{H}_1) dx \\ &= \mathbb{E}_{|\mathcal{H}_1}\{\log l_m(x)\}\end{aligned} \tag{5.15}$$

with $l_m(x) = p_m(x|\mathcal{H}_1)/p_m(x|\mathcal{H}_0)$ being the local likelihood ratio that was defined earlier.

Proof. See Appendix 5.C □

Therefore, Proposition 5.2 allows us to maximize $f_{\mathrm{N},1}(\boldsymbol{w})$ over $\boldsymbol{w}$ (subscript N denotes Neyman-Pearson).

Remark 5.2. *For the problem that minimizes the probability of false alarm P_f for a fixed probability of miss detection P_m, the Kullback-Leibler distance*

$$\begin{aligned}\mathcal{K}(\mathcal{H}_0\|\mathcal{H}_1) &= -\mathbb{E}_{|\mathcal{H}_0}\{\log l(\boldsymbol{y})\} \\ &= -\int \log l(\boldsymbol{y}) p(\boldsymbol{y}|\mathcal{H}_0) d\boldsymbol{y}\end{aligned} \tag{5.16}$$

has to be optimized. Note that the Kullback-Leibler distance is not symmetric, i.e., $\mathcal{K}(\mathcal{H}_0\|\mathcal{H}_1) \neq \mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0)$. Furthermore, Proposition 5.2 holds with the 0 *and* 1 *subscripts interchanged in equations* (5.14) *and* (5.15)*, which leads to the objective*

$$f_{\mathrm{N},2}(\boldsymbol{w}) := \sum_{m=1}^{M} w_m \mathcal{K}_m(\mathcal{H}_0\|\mathcal{H}_1). \tag{5.17}$$

J-divergence

The symmetric form of the Kullback-Leibler distance, *J-divergence*, is another frequently used criterion in the design of experiments. The J-divergence is defined as

$$\mathcal{D}(\mathcal{H}_1\|\mathcal{H}_0) = \mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0) + \mathcal{K}(\mathcal{H}_0\|\mathcal{H}_1). \tag{5.18}$$

A lower bound on $(P_f + P_m)/2$ can be obtained using [Kullback, 2012]

$$\begin{aligned}\mathcal{D}(\mathcal{H}_1\|\mathcal{H}_0) \geq 2\Bigg[&\frac{P_f + P_m}{2}\log\left(\frac{(P_f + P_m)/2}{1 - (P_f + P_m)/2}\right) \\ &+ \left(1 - \frac{P_f + P_m}{2}\right)\log\left(\frac{1 - (P_f + P_m)/2}{(P_f + P_m)/2}\right)\Bigg],\end{aligned}$$

and the results from Theorem 5.1 can be generalized to arrive at an upper bound. The J-divergence is also a reasonable measure in the Bayesian setting with $\pi_0 = 0.5$ as the Bayesian error probability $P_e = (P_f + P_m)/2$ can be both upper and lower bounded by $\mathcal{D}(\mathcal{H}_1 \| \mathcal{H}_0)$. However, for other prior probabilities an upper bound on P_e can be obtained in terms of $\mathcal{D}(\mathcal{H}_1 \| \mathcal{H}_0)$ only for Gaussian observations [Kadota and Shepp, 1967].

The J-divergence is also additive for conditionally independent observations, i.e.,

$$f_{\mathrm{N},3}(\boldsymbol{w}) := \mathcal{D}(\mathcal{H}_1 \| \mathcal{H}_0) = \sum_{m=1}^{M} w_m \mathcal{D}_m(\mathcal{H}_1 \| \mathcal{H}_0),$$

where

$$\mathcal{D}_m(\mathcal{H}_1 \| \mathcal{H}_0) = \mathcal{K}_m(\mathcal{H}_1 \| \mathcal{H}_0) + \mathcal{K}_m(\mathcal{H}_0 \| \mathcal{H}_1). \tag{5.19}$$

The additive property of the J-divergence is straightforward to verify, and it follows directly from Proposition 5.2.

Note that all the distance measures introduced in this section admit a closed-form expression irrespective of the observation distributions. The solvers for designing the sensing operation based on the developed performance measures are presented next.

5.5 Solvers

The performance measures derived in Section 5.4 greatly simplify the sensing design problems P-B and P-N, which are otherwise difficult to solve. The simplified problem is stated as follows.

Problem 5.2 (Simplified sparse sensing design). *Under Assumption 5.1, given M candidate sensors characterized by the conditional pdfs* $\{p_m(x|\mathcal{H}_i)\}_{m=1}^{M}$ *for* $i = 0, 1$, *design a sparsest vector* $\mathbf{w}$ *such that a desired*

(i) Bhattacharyya distance in the Bayesian setting, or

(ii) Kullback-Leibler distance (or J-divergence) in the Neyman-Pearson setting,

is achieved.

These sampling design problems are, respectively, expressed as the following cardinality minimization problems

$$\text{S-B} : \underset{\boldsymbol{w} \in \{0,1\}^M}{\arg\min} \quad \|\boldsymbol{w}\|_0 \quad \text{s.to} \quad f_{\mathrm{B}}(\boldsymbol{w}) \geq \lambda_{\mathrm{B}}; \tag{5.20a}$$

$$\text{S-N} : \underset{\boldsymbol{w} \in \{0,1\}^M}{\arg\min} \quad \|\boldsymbol{w}\|_0 \quad \text{s.to} \quad f_{\mathrm{N}}(\boldsymbol{w}) \geq \lambda_{\mathrm{N}}, \tag{5.20b}$$

where λ_B and λ_N specify the required Bhattacharyya distance and Kullback-Leibler distance (or J-divergence), respectively. The optimization problems S-B and S-N (S denotes simplified problem) are Boolean linear programming problems. In place of $f_\text{N}(\boldsymbol{w})$ in (5.20b), either one of the three performance measures $f_{\text{N},1}(\boldsymbol{w})$, $f_{\text{N},2}(\boldsymbol{w})$, or $f_{\text{N},3}(\boldsymbol{w})$ can be used; however, there is no general answer to the question of how does one performance metric compare with the other. These problems are of the form (P0) introduced in Chapter 2.

For the sake of brevity, we collect the distances $\{\mathcal{B}_m(\mathcal{H}_1\|\mathcal{H}_0)\}$, $\{\mathcal{K}_m(\mathcal{H}_1\|\mathcal{H}_0)\}$, $\{\mathcal{K}_m(\mathcal{H}_0\|\mathcal{H}_1)\}$, or $\{\mathcal{D}_m(\mathcal{H}_1\|\mathcal{H}_0)\}$ in a common vector denoted by $\boldsymbol{d} \in \mathbb{R}^M$. The optimization problems in (5.20) can then be expressed in a general form as

$$\underset{\boldsymbol{w}\in\{0,1\}^M}{\arg\min}\ \|\boldsymbol{w}\|_0 \quad \text{s.to} \quad \boldsymbol{d}^T\boldsymbol{w} \geq \lambda, \tag{5.21}$$

where the threshold corresponds to $\lambda := \lambda_\text{B}$ or $\lambda := \lambda_\text{N}$ for the Bayesian or Neyman-Pearson setting, respectively, with $0 \leq \lambda \leq \mathbf{1}^T\boldsymbol{d}$. Boolean linear programming problems are in general hard to solve [Matoušek and Gärtner, 2007]. However, S-B and S-N are some of the few special cases of a Boolean linear program that have an explicit solution. We give the solution to the considered offline sampling design problem in the following theorem.

Theorem 5.2 (Sparse sampler for distributed detection)**.** *Assuming the entries of* $\boldsymbol{d}$ *are (pre-)sorted in descending order and the entries of* $\mathbf{w}$ *are sorted accordingly. The optimal solution* $\mathbf{w}$ *to* (5.21) *has entries equal to* 1 *at the first* $\widehat{K}$ *entries corresponding to the largest entries in* $\boldsymbol{d}$*, where*

$$\widehat{K} = \min\{i \in \{1,2,\dots,M\}|d_1 + d_2 + \cdots d_i \geq \lambda\}. \tag{5.22}$$

Proof. The proof is straightforward, thus, not detailed. □

In essence, the integer program (5.21) has an explicit solution and it is optimal for (5.21). The solution can be interpreted as follows: recalling $\widehat{K}$ from (5.22), the best subset of sensors out of the M candidate sensors are those $\widehat{K}$ sensors having the smallest local average root-likelihood ratio and largest local average log-likelihood ratio in the Bayesian and Neyman-Pearson setting, respectively.

The appeal of the proposed solution lies in its simplicity. Computationally, the proposed solver is very attractive, for example, with a complexity of $\mathcal{O}(M\log M)$, which is essentially the complexity of the involved sorting algorithm [Papadimitriou, 2003]. A parallel implementation on different processors (i.e., still in an offline centralized setting) of the ordering algorithm further reduces the complexity to $\mathcal{O}(\widehat{K})$ using a back-off mechanism as detailed next: The distance measure d_m is made available to the central unit after a time c/d_m, where c is a known positive constant, and the central unit computes the sum of the received values. If the accumulated sum exceeds the desired threshold λ, the central unit declares a transmission stop[1]. Thus, only the $\widehat{K}$ largest distance values are gathered at the central unit.

[1] If more than one distance is made available at the same time, we randomly pick as many as we need.

In many applications, we might know the number of sensors to select (e.g., we might have already purchased the hardware and we want to use all of them). That is, for a fixed sampler size K, the sensing design problem can be expressed as

$$\text{E-B} : \underset{\boldsymbol{w}\in\{0,1\}^M}{\arg\min}\, P_e(\boldsymbol{w}) \quad \text{s.to} \quad \|\boldsymbol{w}\|_0 = K; \tag{5.23a}$$

$$\text{E-N} : \begin{array}{l} \underset{\boldsymbol{w}\in\{0,1\}^M}{\arg\min}\, P_m(\boldsymbol{w}) \\ \quad \text{s.to} \quad P_f(\boldsymbol{w}) \leq \alpha, \\ \quad \|\boldsymbol{w}\|_0 = K, \end{array} \quad \text{or} \quad \begin{array}{l} \underset{\boldsymbol{w}\in\{0,1\}^M}{\arg\min}\, P_f(\boldsymbol{w}) \\ \quad \text{s.to} \quad P_m(\boldsymbol{w}) \leq \beta, \\ \quad \|\boldsymbol{w}\|_0 = K, \end{array} \tag{5.23b}$$

where E-B (E-N) represents the equivalent Bayesian (equivalent Neyman-Pearson) problem, and α and β are, respectively, the maximum false-alarm rate and miss-detection rate to be satisfied. By appropriately choosing the thresholds e, α and β in (5.2), we can obtain the optimal objective value of (5.2) equal to K, for which P-B (P-N) and E-B (E-N) are equivalent.

We can also simplify E-B and E-N using the Bhattacharyya and Kullback-Leibler distance (or J-divergence) as proxies for the error probabilities, respectively, to arrive at a general form given by

$$\underset{\boldsymbol{w}\in\{0,1\}^M}{\arg\max}\, \boldsymbol{d}^T\boldsymbol{w} \quad \text{s.to} \quad \|\boldsymbol{w}\|_0 = K, \tag{5.24}$$

where it is straightforward to verify that the optimal objective value is given by the sum of the K largest entries of $\boldsymbol{d}$.

We underline that the proposed solver is valid as long as Assumption 5.1 holds, and the observations need not necessarily be Gaussian distributed.

5.6 Illustrative examples

In this section, we illustrate the developed theory of offline sampling design for binary hypothesis testing with a few examples. The sensing operation is designed such that a desired detection performance determined by the Bhattacharyya distance, Kullback-Leibler distance, or J-divergence is achieved. We begin with some examples of Gaussian observations and later on extend it to exponential observation distributions.

5.6.1 Gaussian observations

Uncommon means and common covariances

Detecting signals in Gaussian noise is a well-studied problem in detection theory. In particular, it finds applications in spectrum sensing, target detection, and communications, to list a few. For binary signals in Gaussian noise, that is, observations with uncommon means and common covariance structure under both hypotheses, the conditional distributions are given by

$$\begin{aligned} \mathcal{H}_0 :& \quad \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\theta}_0, \boldsymbol{\Sigma}) \\ \mathcal{H}_1 :& \quad \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\theta}_1, \boldsymbol{\Sigma}), \end{aligned} \tag{5.25}$$

where $\mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma})$ denotes a Gaussian distribution with mean vector $\boldsymbol{\theta}$ and covariance matrix $\boldsymbol{\Sigma}$, the mean vectors $\boldsymbol{\theta}_i = [\theta_{i,1}, \theta_{i,2}, \dots, \theta_{i,M}]^T \in \mathbb{R}^M$ for $i = 0, 1$ as well as the covariance matrix $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_M^2) \in \mathbb{R}^{M \times M}$ are assumed to be perfectly known. The error probabilities admit the following expressions [Moon and Stirling, 2000, pg. 475]

$$\begin{aligned} P_f(\boldsymbol{w}) &= \mathcal{Q}\left(\frac{\gamma + s(\boldsymbol{w})/2}{\sqrt{s(\boldsymbol{w})}}\right); \\ P_m(\boldsymbol{w}) &= 1 - \mathcal{Q}\left(\frac{\gamma - s(\boldsymbol{w})/2}{\sqrt{s(\boldsymbol{w})}}\right), \end{aligned} \tag{5.26}$$

where γ is the threshold defined in (5.9),

$$s(\boldsymbol{w}) = \frac{1}{\sigma^2}(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0)^T \operatorname{diag}(\boldsymbol{w}) \boldsymbol{\Sigma}^{-1} (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0) \tag{5.27}$$

is the *signal-to-noise ratio* (sometimes referred to as the deflection coefficient), and $\mathcal{Q}$ is the complementary Gaussian cumulative distribution function

$$\mathcal{Q}(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-y^2/2\right) dy.$$

Note that the signal-to-noise ratio (5.27) is also linear in $\boldsymbol{w}$. The Bayesian error probability is given by [Moon and Stirling, 2000, pg. 494]

$$\begin{aligned} P_e(\boldsymbol{w}) = \pi_0 \mathcal{Q}&\left(\frac{\gamma' + s(\boldsymbol{w})/2}{\sqrt{s(\boldsymbol{w})}}\right) \\ &+ \pi_1 \left[1 - \mathcal{Q}\left(\frac{\gamma' - s(\boldsymbol{w})/2}{\sqrt{s(\boldsymbol{w})}}\right)\right], \end{aligned} \tag{5.28}$$

where $\gamma' = \log(\pi_0/\pi_1)$ is the threshold in the Bayesian setting.

For the detection problem (5.25), the local Bhattacharrya distance, Kullback-Leibler distance, and J-Divergence can be computed respectively as

$$\begin{aligned} \mathcal{B}_m(\mathcal{H}_1 \| \mathcal{H}_0) &= \frac{1}{8\sigma_m^2}(\theta_{0,m} - \theta_{1,m})^2, \\ \mathcal{K}_m(\mathcal{H}_1 \| \mathcal{H}_0) &= \mathcal{K}_m(\mathcal{H}_0 \| \mathcal{H}_1) = \frac{1}{2\sigma_m^2}(\theta_{0,m} - \theta_{1,m})^2, \\ \mathcal{D}_m(\mathcal{H}_1 \| \mathcal{H}_0) &= \frac{1}{\sigma_m^2}(\theta_{0,m} - \theta_{1,m})^2. \end{aligned}$$

We next remark the following interesting observation. All the three distance measures are equal to the signal-to-noise ratio up to a constant. That is, $\mathcal{B}(\mathcal{H}_1\|\mathcal{H}_0) = s(\boldsymbol{w})/8$, $\mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0) = s(\boldsymbol{w})/2$, and $\mathcal{D}(\mathcal{H}_1\|\mathcal{H}_0) = s(\boldsymbol{w})$. However, these relations are not universal (e.g., they do not hold for non-Gaussian observations). This facts allow us to state the following fundamental result in sampling design for conditionally independent Gaussian observations with common covariance:

Theorem 5.3. *For Gaussian observations with uncommon means and common covariance structure under both hypotheses, maximizing the signal-to-noise ratio over all the possible sampler choices is optimal for P-B and P-N.*

Proof. The proof is straightforward. It can be derived based on results from [Cambanis and Masry, 1983] and the monotonicity of the $\mathcal{Q}$ function. Thus, it is omitted. □

As an example, consider the sinusoidal detection problem with $M = 15$ candidate sensors. The means are $\theta_{0,m} = 0$ and $\theta_{1,m} = \cos 2\pi f m$ with $f := 0.33$ for $m = 1, 2, \ldots, M$. Furthermore, we use $\sigma^2 = 1$, $\pi_0 = 0.3$, $\pi_1 = 0.7$, and $\alpha = 0.01$. In this example, we use a smaller dimension for M to compare the results with the optimal solution of (5.23). Nevertheless, the proposed solvers based on ordering easily scale to higher dimensional problems. We solve (5.23) using exhaustive search over all the $\binom{M}{K}$ combinations for different values of K such that the error probabilities (5.26) and (5.28) are optimized. This is labelled as "Neyman-Pearson/Bayesian optimal" in Fig. 5.1. For this particular example, due to Theorem 5.3, the simplified sensing design problem can be solved optimally also in terms of error probabilities. This is evident from Fig. 5.1, where the solution based on ordering the distance measures (labelled as "Neyman-Pearson/Bayesian simplified, sorting") is on top of the optimum solution based on exhaustive search. The shaded regions in Fig. 5.1 indicate the error probabilities with the worst to best subset of K sensors (including any possible subset of K sensors) for different numbers of selected sensors. In particular, the error probabilities with *random sampling* (or any other sub-optimal sampling), for example, [Cambanis and Masry, 1983, Sung et al., 2005], would span the shaded region.

Remark 5.3 (Choosing λ)**.** *For a desired P_m, say β, and fixed P_f, say α, the threshold $\lambda := \lambda_{\rm N}$ (for a desired signal-to-noise ratio) can be computed using* (5.26). *Specifically,*

$$\lambda_{\rm N} = \left(\mathcal{Q}^{-1}(\alpha) - \mathcal{Q}^{-1}(1-\beta)\right)^2.$$

When λ does not admit a closed form (e.g., with other distributions), the solution path can be used as a guideline to choose λ that results in a desired error probability (often needs to be computed numerically); for example, see Fig. 5.2 to compute $\lambda := \lambda_{\rm B}$, where we solve (5.21) *with the same simulation parameters as before.*

Uncommon covariances and common means

Detecting a change in variance is also frequently encountered in practice, for example, while measuring a physical phenomenon with different sensors each characterized with different noise levels

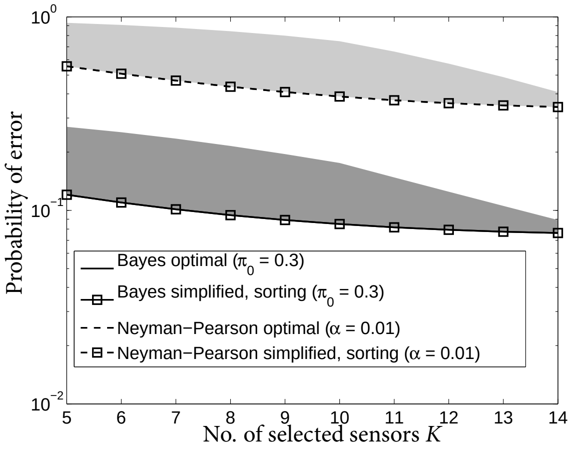


Figure 5.1: The (Bayesian/Neyman-Pearson) probability of error for (5.25) with different numbers of selected sensors K out of $M = 15$ sensors for independent observations. The shaded regions indicate the performance with the worst to best subset of K sensors.

both across the sensors and under both hypotheses. The conditional distributions in this case are given by

$$\begin{aligned} \mathcal{H}_0 : &\quad \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma}_0) \\ \mathcal{H}_1 : &\quad \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma}_1), \end{aligned} \tag{5.29}$$

where $\boldsymbol{\theta}$ is the known mean vector and the diagonal matrix $\boldsymbol{\Sigma}_i = \mathrm{diag}(\sigma_{i,1}^2, \sigma_{i,2}^2, \ldots, \sigma_{i,M}^2)$ for $i = 0, 1$ is known. The local log-likelihood ratio is

$$\log l_m(x) = \frac{1}{2}\log\frac{\sigma_{0,m}^2}{\sigma_{1,m}^2} + x^2\left(\frac{1}{2\sigma_{0,m}^2} - \frac{1}{2\sigma_{1,m}^2}\right).$$

Quantifying the performance of the detector, i.e., expressing P_m, P_f, and P_e in closed form is more difficult than before, as the pdf of $l(\boldsymbol{x})$ can be obtained only by numerical integration [Moon and Stirling, 2000]. However, the proposed performance measures admit known expressions as given next. The local Bhattacharyya distance between the conditional distributions in (5.29) is given by

$$\mathcal{B}_m(\mathcal{H}_1\|\mathcal{H}_0) = \frac{1}{2}\log\left(\frac{\sigma_{0,m}^2 + \sigma_{1,m}^2}{2\sigma_{0,m}\sigma_{1,m}}\right), \tag{5.30}$$

the local Kullback-Leibler distance is given by

$$\mathcal{K}_m(\mathcal{H}_1\|\mathcal{H}_0) = \frac{1}{2}\left(\frac{\sigma_{1,m}^2}{\sigma_{0,m}^2} - 1 - \log\frac{\sigma_{1,m}^2}{\sigma_{0,m}^2}\right), \tag{5.31}$$

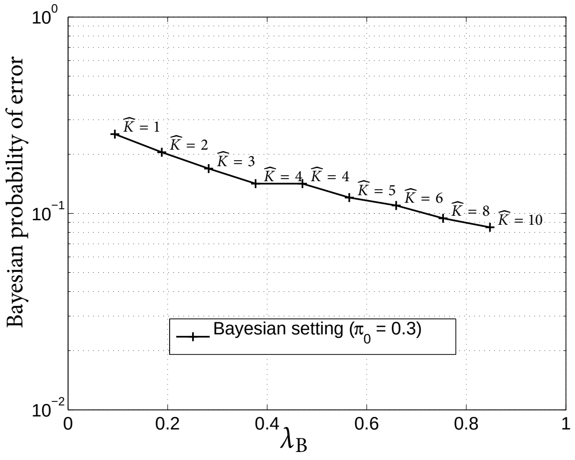


Figure 5.2: The solution path illustrates the Bayesian error probability for different values of the threshold λ_{B}. We use $M = 15$. The number of selected sensors $\widehat{K}$ for a specific value of the threshold is also shown.

and $\mathcal{K}_m(\mathcal{H}_0 \| \mathcal{H}_1)$ is obtained by interchanging the subscripts 0 and 1 in the above equation. Finally, the J-divergence is given by

$$\mathcal{D}_m(\mathcal{H}_1 \| \mathcal{H}_0) = \frac{1}{2}\left(\frac{\sigma_{1,m}^2}{\sigma_{0,m}^2} + \frac{\sigma_{0,m}^2}{\sigma_{1,m}^2} - 2 \right). \tag{5.32}$$

Assume that

$$\mathbf{\Sigma}_0 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.01 \end{bmatrix} \quad \text{and} \quad \mathbf{\Sigma}_1 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.25 \end{bmatrix}$$

and that we want to find the best sensor out of $M = 2$ candidate sensors ($K = 1$). A quick calculation shows that $d_2 > d_1$ for all distances (i.e., the local distance measure of the second sensor is larger than that of the first sensor). Thus, the solution to the S-B (and S-N) will be $\boldsymbol{w} = [0, 1]^T$. This is intuitive as the conditional variance of the second sensor has a larger gap as compared to that of the first sensor, hence the second sensor is more informative.

5.6.2 Exponential observations

Exponentially distributed observations occur while detecting a complex Gaussian signal at the output of a noncoherent receiver. The conditional distributions for exponentially distributed observations for $m = 1, 2, \ldots, M$, are given by

$$\begin{aligned} \mathcal{H}_0 &: \quad x_m \sim \mu_{0,m} \exp\left(-\mu_{0,m} x\right) \\ \mathcal{H}_1 &: \quad x_m \sim \mu_{1,m} \exp\left(-\mu_{1,m} x\right), \end{aligned} \tag{5.33}$$

where $x \in [0, \infty)$. The local log-likelihood ratio is

$$\log l_m(x) = \log \frac{\mu_{1,m}}{\mu_{0,m}} + x(\mu_{0,m} - \mu_{1,m}).$$

Using (5.8), the local Bhattacharyya distance can be computed as

$$\mathcal{B}_m(\mathcal{H}_1 \| \mathcal{H}_0) = -\log \frac{\sqrt{4\mu_{0,m}\mu_{1,m}}}{\mu_{0,m} + \mu_{1,m}}.$$

Similarly, the local Kullback-Leibler distance can be computed as

$$\mathcal{K}_m(\mathcal{H}_1 \| \mathcal{H}_0) = \log \frac{\mu_{0,m}}{\mu_{1,m}} + \frac{\mu_{1,m}}{\mu_{0,m}} - 1,$$

$\mathcal{K}_m(\mathcal{H}_0 \| \mathcal{H}_1)$ is obtained by interchanging the subscripts 0 and 1 in the above equation, and the local J-divergence is given as

$$\mathcal{D}_m(\mathcal{H}_1 \| \mathcal{H}_0) = 2\log \frac{\mu_{1,m}}{\mu_{0,m}} + \frac{\mu_{0,m}^2 - \mu_{1,m}^2}{\mu_{0,m}\mu_{1,m}}.$$

These measures can be directly used in the proposed solvers to design sparse samplers.

5.7 Dependent observations

Throughout most of this chapter so far, we have assumed that the observations are conditionally independent. This assumption is generally valid if the sensors are responsible for the noise in the observations (i.e., receiver noise). However, if the sensors are subject to external noise or if the signal itself is stochastic in nature, then Assumption 5.1 might not be reasonable anymore. Consequently, the additive property of the considered distance measures is also no more valid.

The simplified design problem for this general case (i.e., without any independence assumption), again consists of finding a sparsest $\boldsymbol{w}$ that results in a prescribed distance measure, where we express the Bhattacharyya distance, Kullback-Leibler distance, or J-divergence in terms of $\boldsymbol{w}$. The solution to the above generic problem is hard, nevertheless, we can solve it using standard nonlinear and often nonconvex optimization techniques for a given problem instance (see the example in Section 5.7.2). However, in some cases, a solution can be computed efficiently. As an example, the Gaussian observation case with uncommon means is detailed next.

5.7.1 Gaussian observations with uncommon means

Let us consider the case of binary signal detection in Gaussian noise, and assume the related conditional distributions are given by

$$\begin{aligned} \mathcal{H}_0 &: \quad \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\theta}_0, \boldsymbol{\Sigma}) \\ \mathcal{H}_1 &: \quad \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\theta}_1, \boldsymbol{\Sigma}), \end{aligned} \tag{5.34}$$

where the mean vectors $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ as well as the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{N\times N}$ are assumed to be perfectly known. Note that this model is a generalization of (5.25) with a nondiagonal covariance matrix. The results from Theorem 5.3 generalize to dependent observations. Thus, the error probabilities in (5.2) (or (5.23)) can without loss of optimality be replaced with the signal-to-noise ratio (which is also related to the considered distance measures up to a constant)

$$s(\boldsymbol{w}) := [\boldsymbol{\Phi}(\boldsymbol{w})\boldsymbol{m}]^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{w})\,[\boldsymbol{\Phi}(\boldsymbol{w})\boldsymbol{m}], \tag{5.35}$$

where we use $\boldsymbol{m} = \boldsymbol{\theta}_1 - \boldsymbol{\theta}_0$ and

$$\boldsymbol{\Sigma}(\boldsymbol{w}) = \boldsymbol{\Phi}(\boldsymbol{w})\boldsymbol{\Sigma}\boldsymbol{\Phi}^T(\boldsymbol{w}) \in \mathbb{R}^{K\times K}$$

is a submatrix of $\boldsymbol{\Sigma}$ that includes only the entries corresponding to the selected measurements. More specifically, we want to solve the problem

$$\underset{\boldsymbol{w}\in\{0,1\}^M}{\arg\min} \|\boldsymbol{w}\|_0 \quad \text{s.to} \quad s(\boldsymbol{w}) \geq \lambda, \tag{5.36}$$

where λ is the desired signal-to-noise ratio (or distance measure, or error probability). However, in this case, the simplified problem does not admit an explicit solution. The optimal sampling scheme maximizes $s(\boldsymbol{w})$ in (5.35) over all possible $\boldsymbol{w} \in \{0,1\}^M$ such that $\boldsymbol{w}$ is as sparse as possible. This incurs a combinatorial search over all the 2^M possible combinations. For example, with $M = 100$ candidate sensors, a performance evaluation of about 10^{30} possible choices is needed whose direct enumeration is clearly impossible.

The sampling design $\boldsymbol{w}$ for (5.34) has the following properties:

1. It depends on the first and second order moments of the observations.

2. The mth sampling point depends on the correlation between the sensors.

We next propose some simplifications to solve this problem sub-optimally in polynomial time, yet with a performance that is comparable to the optimal one. Firstly, we write the covariance matrix $\boldsymbol{\Sigma}$ as

$$\boldsymbol{\Sigma} = a\boldsymbol{I} + \boldsymbol{S}, \tag{5.37}$$

where a nonzero $a \in \mathbb{R}$ is chosen such that $\boldsymbol{S} \in \mathbb{R}^{M\times M}$ is invertible and well-conditioned. Using (5.37) in (5.35), we obtain

$$s(\boldsymbol{w}) = \boldsymbol{m}^T\boldsymbol{\Phi}^T(\boldsymbol{w})\left[a\boldsymbol{I} + \boldsymbol{\Phi}(\boldsymbol{w})\boldsymbol{S}\boldsymbol{\Phi}^T(\boldsymbol{w})\right]^{-1}\boldsymbol{\Phi}(\boldsymbol{w})\boldsymbol{m}. \tag{5.38}$$

Using Property 3.1 from Chapter 3, we can simplify $s(\boldsymbol{w})$ to

$$\begin{aligned} s(\boldsymbol{w}) &= \boldsymbol{m}^T\boldsymbol{S}^{-1}\boldsymbol{m} \\ &\quad - \boldsymbol{m}^T\boldsymbol{S}^{-1}\left[\boldsymbol{S}^{-1} + a^{-1}\text{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}^{-1}\boldsymbol{m}. \end{aligned} \tag{5.39}$$

Note that in contrast to (5.38), the design parameter $\boldsymbol{w}$ only shows up at one place in (5.39), which makes the problem much easier. Using the Schur complement, the performance constraint in (5.36), i.e.,

$$\boldsymbol{m}^T\boldsymbol{S}^{-1}\left[\boldsymbol{S}^{-1} + a^{-1}\mathrm{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}^{-1}\boldsymbol{m} \leq \lambda'$$

with $\lambda' := \lambda - \boldsymbol{m}^T\boldsymbol{S}^{-1}\boldsymbol{m}$ can be equivalently expressed as a linear matrix inequality in $\boldsymbol{w}$ under $a > 0$, i.e.,

$$\begin{bmatrix} \boldsymbol{S}^{-1} + a^{-1}\mathrm{diag}(\boldsymbol{w}) & \boldsymbol{S}^{-1}\boldsymbol{m} \\ \boldsymbol{m}^T\boldsymbol{S}^{-1} & \lambda' \end{bmatrix} \geq \boldsymbol{0}, \tag{5.40}$$

and therefore, it is convex in $\boldsymbol{w}$. The parameter a should be chosen such that $\boldsymbol{S}$ is invertible and well-conditioned. Furthermore, because of (5.40) the matrix $\boldsymbol{S}^{-1}+a^{-1}\mathrm{diag}(\boldsymbol{w})$ should be positive definite. This can be achieved, for example, by choosing a such that it satisfies the condition $0 < a < \lambda_{\min}\{\boldsymbol{\Sigma}\}$, since $w_m \geq 0$ for $m = 1, 2, \dots, M$. Although the constraint (5.40) is convex on $\boldsymbol{w}$, the optimization problem (5.36) is still not a convex problem due to the ℓ_0-(quasi) norm cost function and the Boolean constraint.

Convex relaxation

The Boolean constraint set is relaxed to its convex hull, i.e., $0 \leq w_m \leq 1$, $m = 1, 2, \dots, M$, and we also relax the $\|\boldsymbol{w}\|_0$ constraint in (5.36) to its best convex approximate $\boldsymbol{1}^T\boldsymbol{w}$. Thus, the relaxed convex problem, more specifically, a semi-definite programming problem, is given as (cf. (R0) from Chapter 2)

$$\begin{aligned} &\arg\min_{\boldsymbol{w}} \ \boldsymbol{1}^T\boldsymbol{w} \\ &\text{s.to} \quad \begin{bmatrix} \boldsymbol{S}^{-1} + a^{-1}\mathrm{diag}(\boldsymbol{w}) & \boldsymbol{S}^{-1}\boldsymbol{m} \\ \boldsymbol{m}^T\boldsymbol{S}^{-1} & \lambda' \end{bmatrix} \geq \boldsymbol{0}, \\ &\qquad\quad 0 \leq w_m \leq 1, m = 1, 2, \dots, M. \end{aligned} \tag{5.41}$$

For a fixed K, the equivalent problem of the form (5.24) can be relaxed to

$$\begin{aligned} &\arg\max_{\boldsymbol{w}} \ s(\boldsymbol{w}) \\ &\text{s.to} \quad \boldsymbol{1}^T\boldsymbol{w} = K, \\ &\qquad\quad 0 \leq w_m \leq 1, m = 1, 2, \dots, M, \end{aligned}$$

which simplifies to

$$\begin{aligned} &\arg\min_{\boldsymbol{w}} \quad \boldsymbol{m}^T\boldsymbol{S}^{-1}\left[\boldsymbol{S}^{-1} + a^{-1}\mathrm{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}^{-1}\boldsymbol{m} \\ &\text{s.to} \quad \boldsymbol{1}^T\boldsymbol{w} = K, \\ &\qquad\quad 0 \leq w_m \leq 1, m = 1, 2, \dots, M. \end{aligned} \tag{5.42}$$

Algorithm 5.1 Randomized Rounding

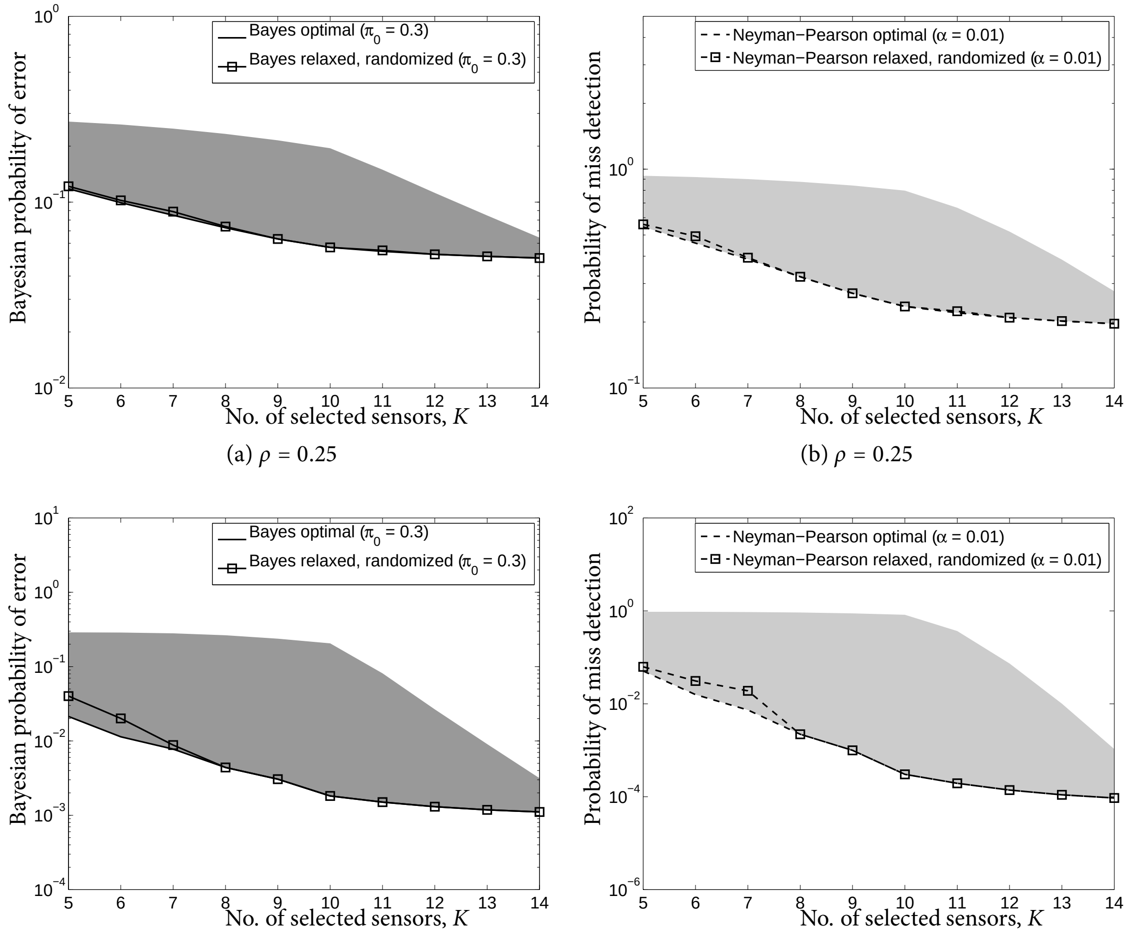
```
1. Given the solution $\boldsymbol{w}^\star$ of (5.41) or (5.43) and a number of randomizations $L$.
2. for $l = 1$ to $L$
3.         generate $w_{m,l} = 1$ with a probability $w_m^\star$
           (or $w_{m,l} = 0$ with a probability $1 - w_m^\star$)
           for $m = 1, 2, \ldots, M$, where $w_m^\star = [\boldsymbol{w}^\star]_m$.
4. end
5. define $\boldsymbol{w}_l = [w_{1,l}, \ldots, w_{M,l}]^T$ and the index set of the candidate estimates
   satisfying the constraints as
       $\Omega \triangleq \begin{cases} \{l \mid s(\boldsymbol{w}_l) \geq \lambda, l = 1, 2, \ldots, L\}, & \text{for (5.41)} \\ \{l \mid \|\boldsymbol{w}_l\|_0 = K, l = 1, 2, \ldots, L\}, & \text{for (5.43).} \end{cases}$
6. If the set $\Omega$ is empty, go back to step 2.
7. output approximate solution $\boldsymbol{w}^\star_{\text{round}} = \boldsymbol{w}_{l^\star}$, where
       $l^\star = \begin{cases} \arg\min_{l \in \Omega} \|\boldsymbol{w}_l\|_0, & \text{for (5.41)} \\ \arg\max_{l \in \Omega} s(\boldsymbol{w}_l), & \text{for (5.43).} \end{cases}$
```

Here, only the second term of (5.39), which depends on $\boldsymbol{w}$ is optimized (minimization is due to its negative sign). Writing (5.42) in the epigraph form [Boyd and Vandenberghe, 2004], we obtain

$$\begin{aligned} &\arg\min_{\boldsymbol{w}, t} \; t \\ &\text{s.to} \quad \mathbf{1}^T \boldsymbol{w} = K, \\ &\qquad \begin{bmatrix} \boldsymbol{S}^{-1} + a^{-1}\text{diag}(\boldsymbol{w}) & \boldsymbol{S}^{-1}\boldsymbol{m} \\ \boldsymbol{m}^T \boldsymbol{S}^{-1} & t \end{bmatrix} \succeq \mathbf{0}, \\ &\qquad 0 \leq w_m \leq 1, m = 1, 2, \ldots, M, \end{aligned} \tag{5.43}$$

with auxiliary variable $t \in \mathbb{R}$.

Subsequently, the selected sensors (i.e., an approximate Boolean solution) can be computed using randomization techniques based on the solution from (5.41) or (5.43) as described in Algorithm 5.1. The relaxed convex problem can be solved using off-the-shelf software, for example, SeDuMi [Sturm, 1999].



(a) ρ = 0.25

(b) ρ = 0.25

(c) ρ = 0.75

(d) ρ = 0.75

Figure 5.3: Error probabilities for (5.34) with different numbers of selected sensors K out of M = 15 sensors. The shaded regions indicate the performance with the worst to best subset of K sensors.

Numerical example

To illustrate sparse sensing with dependent observations, we recall the simulation parameters from Section 5.6.1, but instead of independent noise, we use an autoregressive correlation matrix $\mathbf{\Sigma}$, which is a Toeplitz matrix of the form

$$\mathbf{\Sigma} = \sigma^2 \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{M-1} \\ \rho & 1 & \rho & & \\ \rho^2 & \rho & 1 & & \vdots \\ \vdots & & & \ddots & \\ \rho^{M-1} & & & \cdots & 1 \end{bmatrix}, \tag{5.44}$$

with a known correlation coefficient $\rho \in [0,1]$ and variance $\sigma^2 = 1$. Such a $\mathbf{\Sigma}$ is useful for modeling correlations between distributed sensors; for example, it can represent a spatially decaying correlation function. The convex relaxed problem (5.43) is solved using SeDuMi [Sturm, 1999].

The probability of error, i.e., P_m in the Neyman-Pearson setting and P_e in the Bayesian setting for different numbers of selected sensors is shown in Fig. 5.3. We underline the following observations. The solution with randomized rounding ($L = 50$) is shown in Fig. 5.3 for $\rho = \{0.25, 0.75\}$ with $a = 0.11$ in (5.37). For low values of the correlation coefficient, ρ, the convex relaxation with deterministic rounding is very close to optimal. For larger values of ρ, the solution of the relaxed problem with randomized rounding is still very close to optimal for large values of K, but less optimal for small values of K. As observed in the simulations, for $L = 50 \ll 2^{15}$, the sensing design with randomization is near-optimal in terms of the error probability.

Correlation versus number of selected sensors

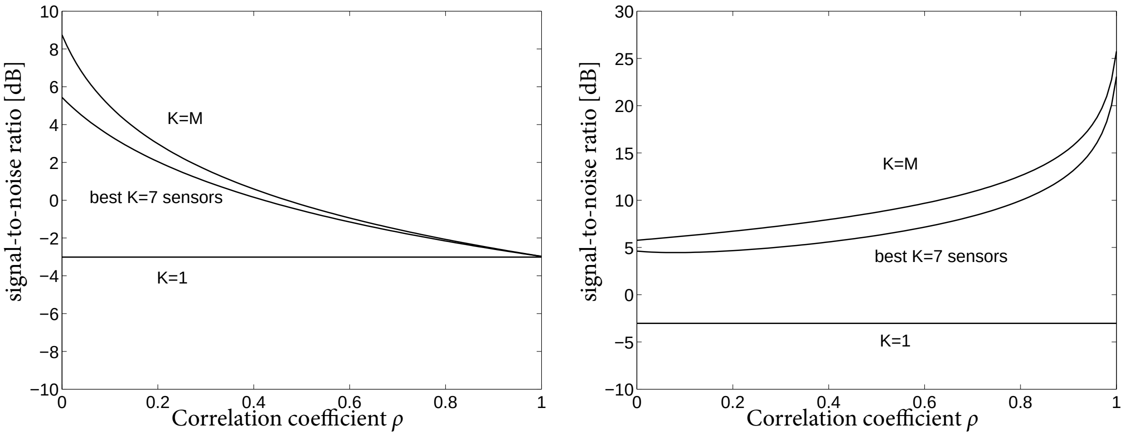
In this subsection, we focus on the number of sensors required to achieve a certain detection performance when the sensors become more *coherent*, i.e., as the correlation coefficient ρ approaches 1. To illustrate this, let us consider the numerical example introduced in Section 5.6.1 with $f \in \{0, 0.33\}$, but with an equi-correlated covariance matrix of the form

$$\mathbf{\Sigma} := \begin{bmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \cdots & \rho \\ \vdots & \vdots & \ddots & \rho \\ \rho & \rho & \cdots & 1 \end{bmatrix} = (1-\rho)\boldsymbol{I}_M + \rho \mathbf{1}_M \mathbf{1}_M^T, \tag{5.45}$$

with a known correlation coefficient $\rho \in [0,1]$. Note that for such a covariance matrix, any $a \neq 1-\rho$ leads to an invertible $\boldsymbol{S}$ in (5.37) that can be used in the solver (5.41).

We first consider the case when $f = 0$, where all the M sensors have the same mean value, i.e., $\boldsymbol{m}$ is the all-one vector up to a constant scaling. We refer to them as *identical sensors*. In this case, any subset of sensors is also the best subset of sensors, hence, random sensing is optimal. As the correlation coefficient ρ approaches 1, the amount of information (Kullback-Liebler distance/Bhattacharyya distance/J-divergence/signal-to-noise ratio) contributed by any random subset of $K > 1$ sensors is the same as that of the contribution from $K = 1$ sensor; see Fig 5.4(a). Thus, even with all the sensors selected the detection performance is limited to that of the performance with one sensor. This is a well-known result from distributed detection that extends to sampling design problems [Chamberland and Veeravalli, 2007].

A more interesting case, in particular for sensing design problems, is when the sensors are not identical ($f = 0.33$), i.e., $\boldsymbol{m}$ has all different entries. When the sensors are not identical, as the correlation coefficient ρ approaches 1, the amount of information contained in the best subset of $K > 1$ sensors increases significantly; see Fig. 5.4(b). More specifically, with equi-correlated yet different observations, to achieve a certain detection performance, the number of sensors required decreases significantly as the correlation coefficient ρ increases. The maximum achievable signal-to-noise ratio is proportional to the inverse of the minimum eigenvalue of $\mathbf{\Sigma}(\boldsymbol{w})$, which is $\lambda_{\min}^{-1}\{\mathbf{\Sigma}(\boldsymbol{w})\} = 1/(1-\rho)$,



(a)

(b)

Figure 5.4: The signal-to-noise ratio for different values of the correlation coefficient ρ. (a) Identical sensors ($f = 0$). (b) Nonidentical sensors ($f = 0.33$).

for any sampler size $K \neq 0$. The optimal sparse sampler would choose the entries of $\boldsymbol{m}$ that are most aligned to the eigenvector corresponding to the minimum eigenvalue of $\boldsymbol{\Sigma}(\boldsymbol{w})$ (hence, as $\rho \to 1$ the signal-to-noise ratio is large). Similarly, if the entries of $\boldsymbol{m}$ are parallel to eigenvector corresponding to the maximum eigenvalue of $\boldsymbol{\Sigma}(\boldsymbol{w})$, that is, the all-one vector, then the signal-to-noise ratio is minimized; this is the case in Fig 5.4(a).

5.7.2 Gaussian Observation with Uncommon Covariances

We now provide some extensions and offer guidelines for determining sparse sensing mechanisms for testing between two covariance matrices. That is, when the covariance structures are different under both hypotheses. Suppose the conditional distribution are given by

$$\begin{aligned} \mathcal{H}_0 : &\quad \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma}_0) \\ \mathcal{H}_1 : &\quad \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma}_1), \end{aligned} \tag{5.46}$$

where the mean vector $\boldsymbol{\theta} \in \mathbb{R}^N$ as well as the $N \times N$ covariance matrices $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$ are assumed to be perfectly known. This model is a generalization of (5.29) with nondiagonal covariance matrices.

As with (5.29), the distance measures are not equal to each other. Using (5.5), the Bhattacharyya distance for the observations of the form $\boldsymbol{y} = \boldsymbol{\Phi}(\boldsymbol{w})\boldsymbol{x}$ can be computed as

$$\begin{aligned} \mathcal{B}(\mathcal{H}_1 \| \mathcal{H}_0) = &\ \frac{1}{2} \log \det\{\boldsymbol{\Sigma}_{01}(\boldsymbol{w})\} \\ &- \frac{1}{4} \left(\log \det\{\boldsymbol{\Sigma}_0(\boldsymbol{w})\} + \log \det\{\boldsymbol{\Sigma}_1(\boldsymbol{w})\} \right), \end{aligned} \tag{5.47}$$

where $\boldsymbol{\Sigma}_i(\boldsymbol{w}) = \boldsymbol{\Phi}(\boldsymbol{w})\boldsymbol{\Sigma}_i\boldsymbol{\Phi}^T(\boldsymbol{w})$ for $i = 01, 0, 1$, with $2\boldsymbol{\Sigma}_{01} = \boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1$. Similarly, using (5.10), we can show that the Kullback-Leibler distance is given by

$$\begin{aligned}\mathcal{K}(\mathcal{H}_1 \| \mathcal{H}_0) = \frac{1}{2}\big(&\mathrm{tr}\{\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{w})\boldsymbol{\Sigma}_1(\boldsymbol{w})\} - \|\boldsymbol{w}\|_0 \\ &- \log\det\{\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{w})\boldsymbol{\Sigma}_1(\boldsymbol{w})\}\big). \end{aligned} \tag{5.48}$$

Here, $\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{w})\boldsymbol{\Sigma}_1(\boldsymbol{w})$ is the *signal-to-noise ratio* matrix.

We can express the Bhattacharyya and Kullback-Leibler distance as a difference of concave functions by relaxing $\boldsymbol{w} \in \{0,1\}^M$ to $[0,1]^M$. That is, we can express (5.47) and (5.48) as

$$f(\boldsymbol{w}) = f_0(\boldsymbol{w}) - f_1(\boldsymbol{w}),$$

where $f_0(\boldsymbol{w})$ and $f_1(\boldsymbol{w})$ are concave functions of its arguments; see Appendix 5.D for the explicit expressions of $f_0(\boldsymbol{w})$ and $f_1(\boldsymbol{w})$. As a consequence, the relaxed problem (for fixed K)

$$\underset{\boldsymbol{w}\in[0,1]^M}{\arg\min} \quad f_1(\boldsymbol{w}) - f_0(\boldsymbol{w}) \quad \text{s.to} \quad \mathbf{1}^T\boldsymbol{w} = K,$$

is not a convex problem as the cost is not a convex function of its argument and has to be solved using nonconvex optimization techniques.

One such heuristic to solve the difference of convex problems is the convex-concave procedure [Yuille and Rangarajan, 2003], where the concave term (here, $f_1(\boldsymbol{w})$) is replaced with its affine approximation (more generally, any reasonable convex approximation of $f_1(\boldsymbol{w})$) while the convex portion , i.e., $-f_0(\boldsymbol{w})$ is retained. The resulting convex problem is iteratively solved to obtain a local optimum.

The J-divergence can be computed using (5.18) as

$$\begin{aligned}\mathcal{D}(\mathcal{H}_1 \| \mathcal{H}_0) = &\frac{1}{2}\mathrm{tr}\{\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{w})\boldsymbol{\Sigma}_1(\boldsymbol{w})\} \\ &+ \frac{1}{2}\mathrm{tr}\{\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{w})\boldsymbol{\Sigma}_0(\boldsymbol{w})\} - \|\boldsymbol{w}\|_0. \end{aligned} \tag{5.49}$$

We next show that maximizing the J-divergence over $\boldsymbol{w}$ can be cast as a convex problem.

Let the covariance matrices $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_0^{T/2}$ and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_1^{T/2}$, respectively, admit the decomposition

$$\boldsymbol{\Sigma}_0 = a_0\boldsymbol{I} + \boldsymbol{S}_0,$$

and

$$\boldsymbol{\Sigma}_1 = a_1\boldsymbol{I} + \boldsymbol{S}_1,$$

with scalars a_0 and a_1 chosen such that $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$ are invertible. Using Property 3.1 from Chapter 3, we can show that the J-divergence (5.49) is equivalent to

$$\begin{aligned}\mathcal{D}(\mathcal{H}_1\|\mathcal{H}_0) = &\frac{1}{2}\operatorname{tr}\{\boldsymbol{S}_0^{-1}\boldsymbol{\Sigma}_1\\ &-\boldsymbol{S}_0^{-1}\left[\boldsymbol{S}_0^{-1}+a_0^{-1}\operatorname{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}_0^{-1}\boldsymbol{\Sigma}_1\}\\ &+\frac{1}{2}\operatorname{tr}\{\boldsymbol{S}_1^{-1}\boldsymbol{\Sigma}_0\\ &-\boldsymbol{S}_1^{-1}\left[\boldsymbol{S}_1^{-1}+a_1^{-1}\operatorname{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}_1^{-1}\boldsymbol{\Sigma}_0\}-\|\mathbf{w}\|_0.\end{aligned}$$

Thus, maximizing the J-divergence over $\boldsymbol{w}$ for a fixed K is the same as minimizing

$$\begin{aligned}&\frac{1}{2}\operatorname{tr}\{\boldsymbol{S}_0^{-1}\left[\boldsymbol{S}_0^{-1}+a_0^{-1}\operatorname{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}_0^{-1}\boldsymbol{\Sigma}_1\}\\ &+\frac{1}{2}\operatorname{tr}\{\boldsymbol{S}_1^{-1}\left[\boldsymbol{S}_1^{-1}+a_1^{-1}\operatorname{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}_1^{-1}\boldsymbol{\Sigma}_0\}\end{aligned}$$

over $\boldsymbol{w}$. To cast this as a convex problem, we introduce two variables

$$\begin{aligned}\boldsymbol{Z}_0 &= \boldsymbol{\Sigma}_1^{T/2}\boldsymbol{S}_0^{-1}\left[\boldsymbol{S}_0^{-1}+a_0^{-1}\operatorname{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}_0^{-1}\boldsymbol{\Sigma}_1^{1/2};\\ \boldsymbol{Z}_1 &= \boldsymbol{\Sigma}_0^{T/2}\boldsymbol{S}_1^{-1}\left[\boldsymbol{S}_1^{-1}+a_1^{-1}\operatorname{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}_1^{-1}\boldsymbol{\Sigma}_0^{1/2},\end{aligned}$$

and obtain

$$\begin{aligned}\underset{\boldsymbol{w},\boldsymbol{Z}_0,\boldsymbol{Z}_1}{\arg\min}\;&\frac{1}{2}\operatorname{tr}\{\boldsymbol{Z}_0\}+\frac{1}{2}\operatorname{tr}\{\boldsymbol{Z}_1\}\\ \text{s.to}\quad&\mathbf{1}^T\boldsymbol{w}=K,\\ &\boldsymbol{\Sigma}_1^{T/2}\boldsymbol{S}_0^{-1}\left[\boldsymbol{S}_0^{-1}+a_0^{-1}\operatorname{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}_0^{-1}\boldsymbol{\Sigma}_1^{1/2}\leq\boldsymbol{Z}_0\\ &\boldsymbol{\Sigma}_0^{T/2}\boldsymbol{S}_1^{-1}\left[\boldsymbol{S}_1^{-1}+a_1^{-1}\operatorname{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}_1^{-1}\boldsymbol{\Sigma}_0^{1/2}\leq\boldsymbol{Z}_1\\ &0\leq w_m\leq 1, m=1,2,\ldots,M.\end{aligned}\tag{5.50}$$

The second and the third constraint can be, respectively, expressed as an LMI in $\boldsymbol{w}$, i.e.,

$$\begin{bmatrix}\boldsymbol{Z}_0 & \boldsymbol{S}_0^{-1}\boldsymbol{\Sigma}_1^{1/2}\\ \boldsymbol{\Sigma}_1^{T/2}\boldsymbol{S}_0^{-1} & \boldsymbol{S}_0^{-1}+a_0^{-1}\operatorname{diag}(\boldsymbol{w})\end{bmatrix}\geq\mathbf{0},$$

$$\begin{bmatrix}\boldsymbol{Z}_1 & \boldsymbol{S}_1^{-1}\boldsymbol{\Sigma}_0^{1/2}\\ \boldsymbol{\Sigma}_0^{T/2}\boldsymbol{S}_1^{-1} & \boldsymbol{S}_1^{-1}+a_1^{-1}\operatorname{diag}(\boldsymbol{w})\end{bmatrix}\geq\mathbf{0}.$$

An approximate Boolean solution has to be subsequently computed using randomized rounding.

The optimization problem of the form (5.20) with unknown K can be derived along similar lines by relaxing the $\|\boldsymbol{w}\|_0$ in the cost function. Before we end this section, we make the following remarks.

- For Gaussian observations, we recall that an upper bound on P_e and P_m can be obtained in terms of J-divergence. Hence, optimizing J-divergence is reasonable under the Bayesian and Neyman-Pearson setting.

- For general Gaussian dependent observations (with uncommon means and uncommon covariances under both hypotheses), the design problems are straightforward combinations of the problems derived in Sections 5.7.1 and 5.7.2.

5.8 Discussion

In this chapter, we have developed a framework for structured and sparse sampler design for distributed detection problems. In particular, we have addressed binary hypothesis testing in both the Bayesian and Neyman-Pearson setting. The proposed framework can be directly applied to sensor placement/selection, sample selection, and fully-decentralized data compression, where we seek the best subset of sensor/sampling locations or data samples that results in a desired detection probability. To simplify the design problem, we have used a number of distance measures that quantify the closeness or divergence between the conditional distributions of the observations. We give an explicit solution for the sampling design problem with conditionally independent observations and the results are summarized as follows. The best sensors are the ones with the smallest local average root-likelihood ratio and largest local average log-likelihood ratio in the Bayesian and Neyman-Pearson setting, respectively. The framework has also been generalized to conditionally dependent observations, with a thorough analysis for the Gaussian case. In that context, we have shown that, for uncommon means and common covariances under both hypotheses, the number of nonidentical Gaussian sensors required to achieve a desired detection performance reduces significantly as the sensors become more coherent.

5.A Proof of Proposition 5.1

In this section, we prove that the additivity of the Bhattacharyya distance is preserved with compression using $\boldsymbol{\Phi}(\boldsymbol{w})$. Using the conditional independence Assumption 5.1, the Bhattacharyya distance in (5.5) can be expressed as

$$\begin{aligned}
\mathcal{B}(\mathcal{H}_1\|\mathcal{H}_0) &= -\log \mathbb{E}_{|\mathcal{H}_0}\{\sqrt{l(\boldsymbol{y})}\} \\
&= -\log \mathbb{E}_{|\mathcal{H}_0}\left\{\prod_{m=1}^{M}[l_m(x)]^{w_m/2}\right\} \\
&= -\log \prod_{m=1}^{M}\mathbb{E}_{|\mathcal{H}_0}\left\{[l_m(x)]^{w_m/2}\right\},
\end{aligned}$$

where $l_m(x)$ is the local likelihood ratio at the mth sensor. Since $w_m \in \{0,1\}$, we can further simplify $\mathcal{B}(\mathcal{H}_1\|\mathcal{H}_0)$ to

$$\begin{aligned}\mathcal{B}(\mathcal{H}_1\|\mathcal{H}_0) &= -\log \prod_{m=1}^{M} \left(\mathbb{E}_{|\mathcal{H}_0}\left\{\sqrt{l_m(x)}\right\}\right)^{w_m} \\ &= \sum_{m=1}^{M} -w_m \log \mathbb{E}_{|\mathcal{H}_0}\left\{\sqrt{l_m(x)}\right\} \\ &= \sum_{m=1}^{M} w_m \mathcal{B}_m(\mathcal{H}_1\|\mathcal{H}_0).\end{aligned}$$

5.B Upper bound on P_m

To derive the upper bound on P_m stated in Theorem 5.1, we use Chebyshev's inequality [Hoeffding, 1963]

$$\Pr(X - \mathbb{E}\{X\} \geq t) \leq \frac{1}{1+\frac{t^2}{v^2}}, \tag{5.51}$$

where X is a random variable with variance v^2 and $t > 0$. Then, P_m simplifies to

$$\begin{aligned}P_m &= \Pr\left(\log l(\boldsymbol{y}) \leq \log\gamma|\mathcal{H}_1\right) \\ &= \Pr\big(\log l(\boldsymbol{y}) - \mathbb{E}_{|\mathcal{H}_1}\{\log l(\boldsymbol{y})\} \\ &\qquad \leq \log\gamma - \mathbb{E}_{|\mathcal{H}_1}\{\log l(\boldsymbol{y})\}|\mathcal{H}_1\big) \\ &= \Pr\left(\log l(\boldsymbol{y}) - \mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0) \geq \mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0) - \log\gamma|\mathcal{H}_1\right),\end{aligned}$$

where the last equation has the same form as the inequality (5.51) with $t = \log\gamma - \mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0)$. Note that the Kullback-Leibler distance is nonnegative, i.e., $\mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0) \geq 0$ with equality only when $\mathcal{H}_0 = \mathcal{H}_1$, and $\log\gamma \leq 0$.

If the variance of $\log l(\boldsymbol{y})$ is v^2, then, from (5.51), we have

$$P_m \leq \frac{1}{1+\frac{\left(\mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0)-\log\gamma\right)^2}{v^2}}$$

This completes the proof.

5.C Proof of Proposition 5.2

In this section, we prove that the additivity of the Kullback-Leibler distance for independent observations is preserved with compression using $\boldsymbol{\Phi}(\boldsymbol{w})$. Assuming Assumption 5.1 holds, then the

Kullback-Leibler distance in (5.10) can be expressed as

$$\begin{aligned}
\mathcal{K}(\mathcal{H}_1\|\mathcal{H}_0) &= \mathbb{E}_{|\mathcal{H}_1}\{\log l(\boldsymbol{y})\} \\
&= \mathbb{E}_{|\mathcal{H}_1}\left\{\log \prod_{m=1}^{M}[l_m(x)]^{w_m}\right\} \\
&= \mathbb{E}_{|\mathcal{H}_1}\left\{\sum_{m=1}^{M} w_m \log l_m(x)\right\} \\
&= \sum_{m=1}^{M} w_m \mathbb{E}_{|\mathcal{H}_1}\{\log l_m(x)\} \\
&= \sum_{m=1}^{M} w_m \mathcal{K}_m(\mathcal{H}_1\|\mathcal{H}_0),
\end{aligned}$$

where $l_m(x)$ is the local likelihood ratio at the mth sensor.

5.D Expressions for $f_0(\boldsymbol{w})$ and $f_1(\boldsymbol{w})$

Let the covariance matrices $\boldsymbol{\Sigma}_{01}$, $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$, respectively, admit a decomposition of the form $\boldsymbol{\Sigma}_{01} = a_{01}\boldsymbol{I} + \boldsymbol{S}_{01}$, $\boldsymbol{\Sigma}_0 = a_0\boldsymbol{I} + \boldsymbol{S}_0$, and $\boldsymbol{\Sigma}_1 = a_0\boldsymbol{I} + \boldsymbol{S}_0$. Here, the scalars a_{01}, a_0, and a_1 are, respectively, chosen such that the matrices $\boldsymbol{S}_{01}$, $\boldsymbol{S}_0$, and $\boldsymbol{S}_1$ are invertible.

Using the Sylvester's determinant identity

$$\det\{\boldsymbol{A} + \boldsymbol{B}\boldsymbol{C}\} = \det\{\boldsymbol{A}\}\det\{\boldsymbol{I} + \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}\}, \tag{5.52}$$

we can express, for example,

$$\begin{aligned}
\det\{\boldsymbol{\Phi}\boldsymbol{\Sigma}_0\boldsymbol{\Phi}^T\} &= \det\{a_0\boldsymbol{I} + \boldsymbol{\Phi}\boldsymbol{S}_0\boldsymbol{\Phi}^T\} \\
&= a_0^M \det\{\boldsymbol{I} + a_0^{-1}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{S}_0\} \\
&= a_0^M \det\{\boldsymbol{I} + a_0^{-1}\mathrm{diag}(\boldsymbol{w})\boldsymbol{S}_0\}.
\end{aligned}$$

Bhattacharyya distance

Ignoring the terms that are independent of the optimization variable $\boldsymbol{w}$, we can express the Bhattacharyya distance (5.47) as

$$f(\boldsymbol{w}) = f_0(\boldsymbol{w}) - f_1(\boldsymbol{w}),$$

where

$$f_0(\boldsymbol{w}) := \frac{1}{2}\log\det\{\boldsymbol{I} + a_{01}^{-1}\mathrm{diag}(\boldsymbol{w})\boldsymbol{S}_{01}\}$$

and

$$\begin{aligned} f_1(\boldsymbol{w}) := \frac{1}{4}\big(&\log\det\{\boldsymbol{I} + a_0^{-1}\mathrm{diag}(\boldsymbol{w})\boldsymbol{S}_0\} \\ &+ \log\det\{\boldsymbol{I} + a_1^{-1}\mathrm{diag}(\boldsymbol{w})\boldsymbol{S}_1\}\big), \end{aligned}$$

are concave functions on $\boldsymbol{w} \in [0,1]^M$.

Kullback-Leibler distance

Using Property 3.1 from Chapter 3, we can write the first term of (5.48), that is,

$$\mathrm{tr}\{\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{w})\boldsymbol{\Phi}\boldsymbol{\Sigma}_1\boldsymbol{\Phi}^T\} = \mathrm{tr}\{\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{w})\boldsymbol{\Phi}\boldsymbol{\Sigma}_1\}$$

as

$$\mathrm{tr}\{\boldsymbol{S}_0^{-1}\boldsymbol{\Sigma}_1 - \boldsymbol{S}_0^{-1}\left[\boldsymbol{S}_0^{-1} + a_0^{-1}\mathrm{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}_0^{-1}\boldsymbol{\Sigma}_1\}.$$

The above function can be expressed as a convex function in $\boldsymbol{w}$ (e.g., using the epigraph form). The second term of (5.48) can be relaxed to a convex function $\boldsymbol{1}^T\boldsymbol{w}$. The last term of (5.48), that is, $\log\det\{\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{w})\boldsymbol{\Sigma}_1(\boldsymbol{w})\}$ is equivalent to

$$\begin{aligned} &\log\det\{\boldsymbol{\Phi}\boldsymbol{\Sigma}_1\boldsymbol{\Phi}^T\} - \log\det\{\boldsymbol{\Phi}\boldsymbol{\Sigma}_0\boldsymbol{\Phi}^T\} = \\ &\log\det\{\boldsymbol{I} + a_1^{-1}\mathrm{diag}(\boldsymbol{w})\boldsymbol{S}_1\} - \log\det\{\boldsymbol{I} + a_0^{-1}\mathrm{diag}(\boldsymbol{w})\boldsymbol{S}_0\}. \end{aligned}$$

Thus, we can equivalently express (5.48) as $f(\boldsymbol{w}) = f_0(\boldsymbol{w}) - f_1(\boldsymbol{w})$ with

$$\begin{aligned} f_0(\boldsymbol{w}) := -\mathrm{tr}\{\boldsymbol{S}_0^{-1}\left[\boldsymbol{S}_0^{-1} + a_0^{-1}\mathrm{diag}(\boldsymbol{w})\right]^{-1}\boldsymbol{S}_0^{-1}\boldsymbol{\Sigma}_1\} - \boldsymbol{1}^T\boldsymbol{w} \\ + \log\det\{\boldsymbol{I} + a_1^{-1}\mathrm{diag}(\boldsymbol{w})\boldsymbol{S}_1\} \end{aligned}$$

and

$$f_1(\boldsymbol{w}) := \log\det\{\boldsymbol{I} + a_0^{-1}\mathrm{diag}(\boldsymbol{w})\boldsymbol{S}_0\},$$

which are concave in $\boldsymbol{w}$.

Part of this chapter was published as: S.P. Chepuri and G. Leus. Continuous Sensor Placement. *IEEE Signal Processing Letters*, 22(5): 544-548, May 2015.

Chapter 6

Continuous Sparse Sensing

Contents

6.1 Introduction

In the sparse sensing schemes considered so far in this thesis, the focus was on discrete sparse sensing. That is, we were selecting sparse sensing patterns from a discrete set of candidates, e.g., temporal samples, sensor positions, which were obtained by gridding the output space, in order to reach a desired inference performance. In this chapter, we will discuss continuous (or off-the-grid) sparse sensing, where we can take samples anywhere in the continuous output space (i.e., we can sample in between the grid points). Continuous sparse sensing will reduce the sensing costs for a given inference performance, or in other words, it will improve the inference performance for a fixed sampler size.

To realize continuous sparse sensing, we start with a discretized output space and model every sampling point in the continuous sampling space as a discrete sampling point plus a continuous perturbation. Then, we solve for the smallest set of possible discrete points as well as the best possible perturbations that result in a desired inference performance. Although we focus on the estimation task with linear models in this chapter, the proposed approach can be generalized to nonlinear models and also other inference tasks (e.g., filtering and detection) discussed in this thesis. Depending on the nature of the inference task, the risk function that quantifies the inference quality will depend on the discrete sampling points as well as their perturbations. In sum, the main contribution of this chapter is a framework of continuous sparse sensing, which allows for off-the-grid sensor placement.

Let $x(t)$ denote the observation signal with a continuous-domain argument, where without loss of generality $t \in [0, T]$ denotes the sampling domain. We will restrict ourselves to the one-dimensional spatial domain, but the ideas presented can be applied directly to higher dimensions and even to temporal or spatio-temporal domains. Assume that $x(t)$ represents the measured physical field over a continuous one-dimensional space t, and it satisfies the linear model

$$x(t) = \boldsymbol{h}^H(t)\boldsymbol{\theta} + n(t) \tag{6.1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^N$ collects the parameters to be estimated, $\boldsymbol{h}(t) \in \mathbb{C}^N$ is the known linear model representing the mapping between the parameters and the measurements, and $n(t)$ is the noise. Furthermore, we assume $\boldsymbol{h}(t) = 0$ for $t < 0$ and $t > T$. In other words, $\boldsymbol{h}(t)$ is completely described by its variation in the interval $t \in [0, T]$.

The fundamental question of interest is—where to sample $x(t)$ in order to reach a desired inference performance? We next state the problem more precisely.

Problem 6.1. *Given the model* (6.1) *and a desired estimation accuracy, find the optimal sampling locations in the range* $[0, T]$ *such that the number of samples is minimum and the desired estimation accuracy is achieved.*

6.2 Sensitivity to Gridding

We now recall the sensing model (2.3) that we introduced in Chapter 2, where we discretize (e.g., by regular sampling) the output space with M points denoted by $\{t_m\}_{m=1}^M$. The inference performance (estimation accuracy, in this case) is limited by the choice of the initial grid points $\{t_m\}_{m=1}^M$ as the resolution might be too low, especially when $\boldsymbol{h}(t)$ is fast varying compared to the chosen grid.

Let $x_m = x(t_m)$ be the discrete-domain observations, $\boldsymbol{h}_m = \boldsymbol{h}(t_m)$ denote the discretized model, and $n_m = n(t_m)$ represent the noise. We assume that the noise is white Gaussian with variance σ^2 (for spatial sampling, the sensor noise variance is independent of the sampling density). Using the above notations, we can write the discrete-domain version of (6.1) as

$$x_m = \boldsymbol{h}_m^H\boldsymbol{\theta} + n_m, m = 1, 2, \ldots, M. \tag{6.2}$$

The data $\boldsymbol{x} = [x(t_1), x(t_2), \ldots, x(t_M)]^T$ is acquired through the discrete sparse sensing operation as

$$\boldsymbol{y} = \boldsymbol{\Phi}(\boldsymbol{w})\boldsymbol{x} = \operatorname{diag}_{\mathrm{r}}(\boldsymbol{w})\boldsymbol{x},$$

where we design a sparsest vector

$$\boldsymbol{w} = [w_1, w_2, \dots, w_M] \in \{0,1\}^M$$

by solving (P0) introduced in Chapter 2 with $f(\boldsymbol{w}) = \sigma^2 \mathrm{tr}\{(\sum_{m=1}^{M} w_m \boldsymbol{h}_m \boldsymbol{h}_m^H)^{-1}\}$. That is, we solve

$$\begin{aligned} &\underset{\boldsymbol{w}\in\{0,1\}^M}{\arg\min} \ \|\boldsymbol{w}\|_0 \\ &\text{s.to} \quad \sigma^2 \mathrm{tr}\left\{\left(\sum_{m=1}^{M} w_m \boldsymbol{h}_m \boldsymbol{h}_m^H\right)^{-1}\right\} \leq \lambda. \end{aligned} \tag{6.3}$$

Recall that $f(\boldsymbol{w}) = \sigma^2 \mathrm{tr}\{(\sum_{m=1}^{M} w_m \boldsymbol{h}_m \boldsymbol{h}_m^H)^{-1}\}$ is the Cramér-Rao bound (equal to the mean squared error of the least squares estimation) for linear models in Gaussian noise; see Remark 3.1 from Chapter 3.

We now illustrate the sensitivity to gridding with the following numerical example. Consider the linear model in (6.1) with the following specifications. Let the parameter vector $\boldsymbol{\theta}$ be of length 2×1. Consider a sum of sinusoids model for $\boldsymbol{h}(t) = [h_1(t), h_2(t)]^T$ with $h_i(t) = \sum_{p=1}^{P_i} \alpha_{p,i} \sin(2\pi f_{p,i} t)$ for $i = 1, 2$. Let $\boldsymbol{f}_i = [f_{1,i}, \dots, f_{P_i,i}]^T$ and $\boldsymbol{\alpha}_i = [\alpha_{1,i}, \dots, \alpha_{P_i,i}]^T$ for $i = 1, 2$. We use the following parameters: $P_1 = 5$, $P_2 = 5$, $T = 10$, $\sigma^2 = 1$,

$$\begin{aligned} \boldsymbol{f}_1 &= [0.1, 0.33, 0.67, 0.78, 0.95]^T, \\ \boldsymbol{f}_2 &= [0.15, 0.7, 0.4, 0.58, 0.85]^T, \\ \boldsymbol{\alpha}_1 &= [0.5, 0.65, 0.3, -0.15, 0.45]^T, \\ \boldsymbol{\alpha}_2 &= [-0.25, -0.33, -0.6, 0.95, -0.25]^T. \end{aligned}$$

6.2.1 Coarse Gridding

Assume that the field can be measured at $M = 5$ potential locations. Let us consider the following case of candidate sampling locations $t_m := \{1, 3, 5, 7, 9\}$. The mean squared error resulting from the samples at these locations is 1.47. The optimal mean squared error using 5 samples, on the other hand, is around 0.16.

We remark the following two observations. Firstly, the mean squared error resulting from the samples at the above locations is much lower than the optimal mean squared error. Secondly, any subset of these 5 samples will naturally also result in a mean squared error larger than 1.47. Hence, due to the involved discretization, coarse gridding might not lead to the desired mean squared error even if all candidate samples/sensors are selected.

6.2.2 Fine Gridding

Alternatively, the initial grid can be very dense with the candidate sampling locations at infinitesimal distance apart. Then (6.3) would choose many sensors within one or more virtual sampling bins.

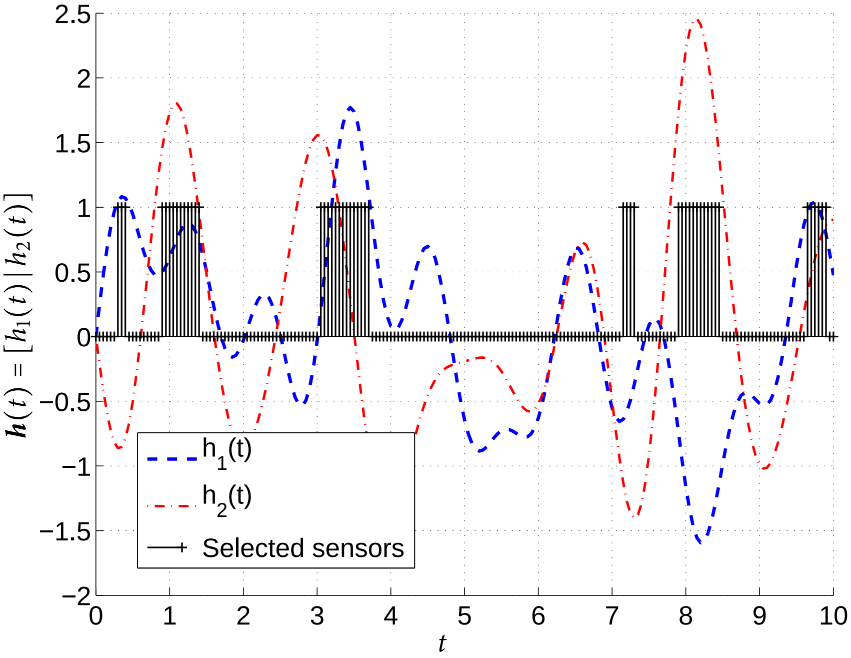


Figure 6.1: The field is measured at $M = 201$ discrete locations obtained by uniformly gridding the interval $[0, 10]$ m. The mean squared error bound is set to $\lambda = 0.026$, leading to $K = 50$ selected sensors.

For the setup discussed in the previous subsection, for a fine grid with $M = 201$ candidate sampling points the solution to the relaxed version of (6.3) is illustrated in Fig. 6.1. The sensors within the most informative bin are selected first before going to the next informative bin, and so on, till the desired mean squared error is achieved.

The fine gridding has two main drawbacks. Firstly, it might not be practically feasible to sample so close to each other. In addition, the reason why more samples are selected within a certain bin is to improve the signal-to-noise ratio. In case of spatial sampling, it might be desirable to restrict the number of spatial samples. That is, instead of placing additional (expensive) sensors within a certain bin, the signal-to-noise ratio can be compensated by other (cheaper) means, e.g., temporal averaging using a single sensor. Finally, note that the solvers based on convex optimization techniques incur a cubic complexity making fine gridding also computationally less viable.

6.3 Sensing model based on binning

The motivation behind a coarse discretization in discrete sparse sensing was computational tractability, but its performance is limited by the choice of the initial grid. On the contrary, fine gridding suffers from a high computational complexity and multiple closely spaced sensors for signal-to-noise ratio improvement. In this section, we present the sparse sensing model based on binning, which allows to sample anywhere in the continuous sampling domain. Specifically, we take one sample per bin. To realize this, we augment the discrete model by including additional variables that account for the continuous nature of the sampling domain.

We now recall the sparse sensing model that we introduced in Chapter 2, where we acquire $x(t)$ through a continuous-domain sparse sensing function $w(t) = \sum_{m=1}^{K} \delta(t-\tau_m)$ with unknown indices $\{\tau_m\}_{m=1}^{K}$ and unknown number of samples K, as

$$y(t) = w(t)x(t) = \sum_{m=1}^{K} x(\tau_m)\delta(t-\tau_m).$$

By discretizing (e.g., regular sampling) the output space with $M \gg K$ points denoted by $\{t_m\}_{m=1}^{M}$ we alternatively modeled $y(t)$ as (cf. (2.3))

$$y(t) = \sum_{m=1}^{M} w_m x(t_m)\delta(t-t_m)$$

where $w_m = (0)1$ indicates whether sample $x(t_m)$ is (not) selected. Here, the assumption was that the $\{\tau_m\}_{m=1}^{K}$ lie on the discrete grid.

When they do not lie on the discrete grid we follow a binning approach. If $x(t)$ is sufficiently smooth (i.e., its first-order derivative exists and is continuous), then local shifts of $x(t)$ can be approximated using its derivative based on a first-order Taylor expansion:

$$x(t_m + p_m) \approx x_m + p_m x'_m, \; m = 1, 2, \ldots, M, \tag{6.4}$$

where $\{t_m\}_{m=1}^{M}$ are the discrete sampling points, p_m represents the continuous perturbation around t_m with $|p_m| < 0.5\delta$ (δ denotes the bin size), and x'_m is the derivative of $x(t)$ towards t evaluated at t_m, i.e., $x'_m = \left.\frac{\partial x(t)}{\partial t}\right|_{t=t_m}$. Using this approximation, we arrive at an off-the-grid sensing model, i.e., we can model $y(t)$ as

$$y(t) = \sum_{m=1}^{M} w_m \left(x_m + p_m x'_m\right) \delta(t-t_m). \tag{6.5}$$

Such first-order interpolations have also been used in the context of sparse signal recovery in continuous compressive sensing to overcome problems due to gridding, but at the input grid [Ekanadham et al., 2011, Zhu et al., 2011]. As a remark, alternative interpolation techniques (e.g., polar interpolation [Ekanadham et al., 2011]) can be considered.

Stacking $\{w_m\}_{m=1}^{M}$ in vector $\boldsymbol{w}$, $\{p_m\}_{m=1}^{M}$ in vector $\boldsymbol{p}$, and $\{x'_m\}_{m=1}^{M}$ in vector $\boldsymbol{x}'$, we can write the discrete-domain counterpart of (6.5) as

$$\boldsymbol{y} = \boldsymbol{\Phi}(\boldsymbol{w}) \left[\boldsymbol{x} + \mathrm{diag}(\boldsymbol{x}')\boldsymbol{p}\right] = \mathrm{diag}_{\mathrm{r}}(\boldsymbol{w}) \left[\boldsymbol{x} + \mathrm{diag}(\boldsymbol{x}')\boldsymbol{p}\right], \tag{6.6}$$

where $\boldsymbol{y}$ stacks $\{y_m\}_{m=1}^{M}$.

For the linear model (6.1), we can then represent the off-the-grid samples as

$$y_m = w_m \left[(\boldsymbol{h}_m + p_m \boldsymbol{h}'_m)^H \boldsymbol{\theta} + n_m\right], m = 1, 2, \ldots, M, \tag{6.7}$$

	$M = 5$	$M = 11$	$M = 21$	$M = 41$	$M = 81$
Sensor selection	1.47	0.47	0.28	0.20	0.18
Continuous sparse sensing (proposed)	1.32	0.36	0.22	0.18	0.17

Table 6.1: Mean squared error with 5 selected sensors for different grid densities.

where $\boldsymbol{h}'_m = \frac{\partial \boldsymbol{h}(t)}{\partial t}\Big|_{t=t_m}$. Note that by using this idea, gridding actually results in binning, where we take at most one sample per bin. In what follows, we will derive the risk function that depends on $\boldsymbol{w}$ and $\boldsymbol{p}$ for the linear inverse problem.

6.4 Risk function for continuous sparse sensing

As discussed in Chapter 3, for statistical estimation problems, we use scalar measures of the Fisher information matrix to quantify the estimation accuracy. The computation of the FIM for a linear model (6.7) is straightforward (cf. (3.5) from Chapter 3). It is given by

$$\begin{aligned}\boldsymbol{F}(\boldsymbol{w},\boldsymbol{p}) = \sum_{m=1}^{M} & w_m \boldsymbol{h}_m \boldsymbol{h}_m^H + w_m p_m^2 \boldsymbol{h}'_m \boldsymbol{h}'^H_m \\ & + w_m p_m (\boldsymbol{h}'_m \boldsymbol{h}_m^H + \boldsymbol{h}_m \boldsymbol{h}'^H_m).\end{aligned}$$

The A-optimality criterion (cf. Section 3.1.1 from Chapter 3), which is the trace of the inverse FIM, for linear Gaussian models corresponds to the mean squared error of the least squares estimate. Introducing variables $\boldsymbol{u} = [u_1, u_2, \ldots, u_M]^T$ with $u_m = w_m p_m^2$ (which can also be written as $u_m = w_m^2 p_m^2$), and $\boldsymbol{v} = [v_1, v_2, \ldots, v_M]^T$ with $v_m = w_m p_m$, we can write the risk function that we optimize as

$$\begin{aligned} f(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}) &= \sigma^2 \mathrm{tr}\Bigg\{\Bigg(\textstyle\sum_{m=1}^{M} w_m \boldsymbol{h}_m \boldsymbol{h}_m^H + u_m \boldsymbol{h}'_m \boldsymbol{h}'^H_m \\ &\quad + v_m(\boldsymbol{h}'_m \boldsymbol{h}_m^H + \boldsymbol{h}_m \boldsymbol{h}'^H_m)\Bigg)^{-1}\Bigg\}, \end{aligned} \tag{6.8}$$

where $\boldsymbol{u} = \boldsymbol{v}^{\odot 2}$.

In contrast to discrete sparse sensing, the risk function in this case, depends on additional parameters that are related to the perturbation of the discrete sampling points.

6.5 Solver

The optimization variables $(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w})$ in (6.8) are related through a structure. Recall that $w_m = 1$ indicates that the mth sample is selected. Only when w_m is nonzero, the corresponding continuous variables u_m and v_m are nonzero[1]. In other words, the vectors $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{w}$ all share the same support set. Hence, instead of simply minimizing the cardinality of $\boldsymbol{w}$ as in (P0) that was introduced in Chapter 2, we can exploit the structure and jointly optimize their cardinality to minimize the number of samples, and thus selecting the smallest set of perturbations associated with the discrete sampling points. Defining the matrix $\boldsymbol{Z} = [\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}] \in \mathbb{R}^{M\times 3}$, the proposed continuous sensor placement problem can be formulated as

$$\begin{aligned}
&\arg\min_{\boldsymbol{Z}} \ \|\boldsymbol{Z}\|_{2,0} && (6.9a)\\
&\text{s.to} \quad f(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) \le \lambda_{\text{c}}, && (6.9b)\\
&\qquad \boldsymbol{Z} = [\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}], &&\\
&\qquad \boldsymbol{u} = \boldsymbol{v}^{\odot 2}, && (6.9c)\\
&\qquad w_m \in \{0,1\}, m = 1, 2, \ldots, M, && (6.9d)\\
&\qquad -0.5\delta < v_m < 0.5\delta, m = 1, 2, \ldots, M, && (6.9e)\\
&\qquad 0 < u_m < 0.25\delta^2, m = 1, 2, \ldots, M, && (6.9f)
\end{aligned}$$

where the ℓ_2/ℓ_0-(quasi) norm counts the number of nonzero rows of $\boldsymbol{Z}$ as follows $\|\boldsymbol{Z}\|_{2,0} := |\{m : \sqrt{u_m^2 + v_m^2 + w_m^2} \neq 0\}|$, and the convex constraint (6.9b) specifies the estimation accuracy through a threshold λ_{c} (c denotes continuous). Since the continuous variable p_m takes values in the range $[-0.5\delta, 0.5\delta]$, we obtain the convex box constraints (6.9e) and (6.9f). The optimization problem (6.9) is nonconvex due to: (a) cardinality cost, (b) Boolean constraint (6.9d), and (c) quadratic equality (6.9c). Therefore, it is (in general) difficult to solve (6.9) optimally.

6.5.1 Convex relaxation

We now use some standard convex relaxation techniques to simplify (6.9) and solve it sub-optimally. The ℓ_2/ℓ_0-(quasi) norm is relaxed with its best convex approximation, i.e., an ℓ_2/ℓ_1-mixed norm defined as $\|\boldsymbol{Z}\|_{2,1} := \sum_{m=1}^{M} \sqrt{u_m^2 + v_m^2 + w_m^2}$. The Boolean $w_m \in \{0,1\}$ constraint is replaced with a convex set $w_m \in [0,1]$. The constraint (6.9c) is equivalently expressed as $\boldsymbol{u} = \text{diag}(\boldsymbol{U})$, where $\boldsymbol{U} = \boldsymbol{v}\boldsymbol{v}^H$ is a rank-1 matrix with $[\boldsymbol{U}]_{i,j\neq i} = 0,\ \forall i, j$. Dropping the rank constraint on $\boldsymbol{U}$ and replacing the equality with an inequality as

$$\boldsymbol{U} \ge \boldsymbol{v}\boldsymbol{v}^H \Leftrightarrow \begin{bmatrix} \boldsymbol{U} & \boldsymbol{v} \\ \boldsymbol{v}^H & 1 \end{bmatrix} \ge 0,$$

[1] We are not interested in a nonzero u_m or v_m when $w_m = 0$.

we arrive at the relaxed continuous sensor placement problem:

$$\begin{aligned} \underset{\boldsymbol{Z},\boldsymbol{U}}{\arg\min} \quad & \|\boldsymbol{Z}\|_{2,1} \\ \text{s.to} \quad & f(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}) \leq \lambda_{\mathrm{c}}, \\ & \boldsymbol{Z} = [\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}], \\ & \begin{bmatrix} \boldsymbol{U} & \boldsymbol{v} \\ \boldsymbol{v}^H & 1 \end{bmatrix} \geq 0, \\ & \mathrm{diag}(\boldsymbol{U}) = \boldsymbol{u}, [\boldsymbol{U}]_{i,j\neq i} = 0, \forall i,j, \\ & 0 \leq w_m \leq 1, m = 1,2,\ldots,M, \\ & -0.5\delta < v_m < 0.5\delta, m = 1,2,\ldots,M, \\ & 0 < u_m < 0.25\delta^2, m = 1,2,\ldots,M. \end{aligned} \tag{6.10}$$

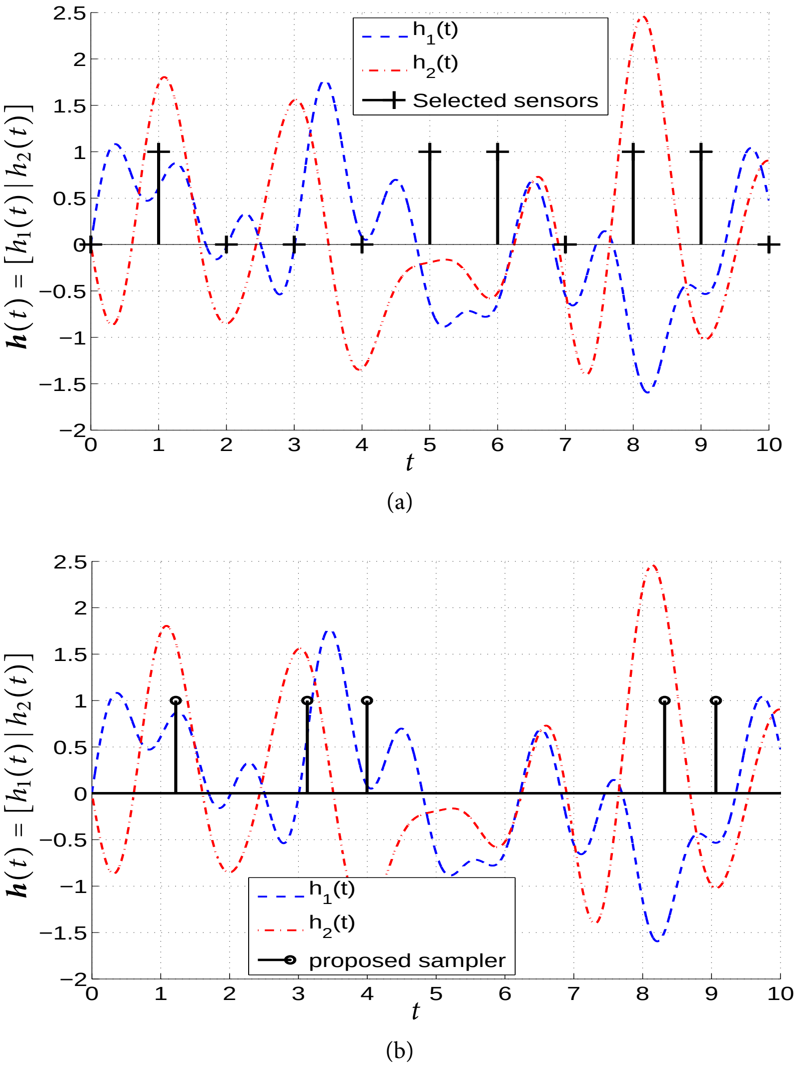
Subsequently, an approximate Boolean solution for $\boldsymbol{w}$ has to be recovered either by deterministic or randomized rounding as discussed in Chapter 3. Finally, the sensor placements are given by shifting the locations of the selected sensors according to $\boldsymbol{v}$. The relaxed optimization problem can be solved using off-the-shelf solvers like SeDuMi [Sturm, 1999]. We underline here that the proposed sensor placement is not limited to the initial chosen grid points, and we basically replace grid points with bins allowing one sensor per bin. However, this feature comes at an additional complexity compared to that of solving the sensor selection problem with a fixed discrete grid. The increase in complexity is due to the additional variables like the continuous perturbation parameter and the associated box constraints.

Furthermore, the proposed algorithm is a single step approach. However, it is also possible to compute the initial points from discrete sparse sensing in step-1, and based on these initial points, iteratively solve for the sampling locations using gradient descent techniques to find good local solutions in step-2. But, such a two-step approach can result in a local optimum. Moreover, it will be a complex iterative approach, which can be completely different from the solution of the proposed approach.

6.5.2 Numerical example

To validate the proposed continuous sparse sensing approach, we refer to the sum of sinusoids example introduced earlier in Section 6.2. Let the initial coarse grid include $M = 11$ sampling locations $\{t_m = (m-1)\delta \,|\, m \in \{1,2,\ldots,11\}\}$ with $\delta = 1$. Note that the proposed framework is not limited to the sum of sinusoids model, but is valid for any general known model.

Fig. 6.2(a) illustrates the sensor placement via discrete sparse sensing (specifically, sensor selection [Joshi and Boyd, 2009]). The best subset of sensors is computed by solving the relaxed version of (6.3). For sensor selection, we choose $\lambda := 0.47$ such that 5 sensors are selected out of 11 available sensors. For the considered scenario, the mean squared error achieved with such a sensor placement is ≈ 0.47.



(a)

(b)

Figure 6.2: The field is measured at $M = 11$ discrete locations with $\delta = 1$ and $T = 10$. (a.) Sensor placement via sensor selection. (b.) Proposed continuous sparse sensing. A different threshold λ is used for (a) and (b), such that 5 sensors are selected.

Fig. 6.2(b) illustrates the results from the proposed continuous sensor placement obtained by solving the relaxed optimization problem (6.10). We use $\lambda_c = 0.064$, which has also been chosen such that 5 sensors are selected. The mean squared error achieved with the proposed placement is ≈ 0.36, which is lower than the mean squared error obtained by the sensor placement through sensor selection. The threshold λ_c is an underestimate of the mean squared error (unlike λ), and this is due to the approximation in (6.4). The threshold corresponding to a certain mean squared error can be chosen by computing the solution path for different λ_c values.

Finally, in Table 6.1 we evaluate the mean squared error with 5 selected sensors obtained by solving the sensor selection problem and the proposed continuous placement for different grid densities. The optimization problems are solved in MATLAB using SeDuMi [Sturm, 1999]. We consider different grid densities $M = \{5, 11, 21, 41, 81\}$ and in each case we use a threshold that selects 5 sensors. The continuous sensor placement offers better mean squared error with a reasonable increase in complexity.

6.6 Discussion

We have proposed a framework of continuous sparse sensing in this chapter, where we select sparse sensing patterns from a continuous domain instead of a discrete one. We model an off-the-grid sampling point as an on-the-grid sampling point plus a perturbation assuming that the continuous-domain function is sufficiently smooth. In other words, we can take samples in between the grid points. Expressing the inference quality determining risk as a function of the discrete sampling points and their perturbations, we have designed a continuous sparse sensing operator by solving a convex program.

Part II

Applications

Part of this chapter was published as: S.P. Chepuri *et al.* Joint Clock Synchronization and Ranging: Asymmetrical Time-Stamping and Passive Listening. *IEEE Signal Processing Letters*, 20 (1): 51 - 54, Jan 2013.

Chapter 7

Wireless Clock Synchronization

Contents

7.1 Introduction

Sensor networks can be designed to faithfully represent distributed signals (e.g., a spatially varying phenomenon such as the temperature field). In other words, a sensor network can be used as a spatial sampling device. Further, to acquire multidimensional distributed signals that exist in space and time, we also need to perform sampling in time. The temporal sampling is achieved using the sensor's analog-to-digital converters or time-to-digital converters, for example. Each sensor has an independent sample clock (i.e., oscillator), and its stability basically determines the alignment of the temporal sampling grid across the sensors. This temporal sampling grid is perfectly aligned if all the sensors share a common clock. When uncommon, the individual clocks drift from each other due to imperfections in the oscillator, aging, and other environmental variations. This drift will result in the misalignment of the temporal sampling grid across the sensors. Therefore, clock synchronization among different nodes each having its own autonomous clock forms a key component of a sensor network.

A plethora of clock synchronization algorithms based on the time-of-flight measurements of the messages have been proposed [Noh et al., 2008, Freris et al., 2010, Wu et al., 2011, Rajan and van der Veen, 2011], which could operate via a two-way time-stamp exchange [Wu et al., 2011] or pairwise broadcast synchronization (PBS) [Noh et al., 2008]. Assuming an affine (i.e., first order) clock model and one of the nodes as a reference, the unknown clock skews and clock offsets of the remaining nodes in a network can be estimated using least squares based on the two-way time-stamp exchange protocol [Rajan and van der Veen, 2011]. This is achieved by exploiting the redundancy due to all possible pairwise links in the network. Sensor nodes are usually battery powered. Thus, all the tasks of a sensor network, including synchronization, should be carefully performed to ensure longer operating lifetime. For synchronization, this means to minimize the number of transmissions between nodes during which the time-stamps are recorded.

In this chapter, we extend the joint clock synchronization and ranging algorithm of [Rajan and van der Veen, 2011] to fully harness the broadcast nature of the wireless medium. By doing so, the number of active transmissions between the nodes can be significantly reduced for a fixed synchronization accuracy. In other words, the synchronization accuracy can be improved for a fixed number of active transmissions. To realize this, we propose an asymmetrical time-stamping and passive listening (ATPL) protocol. The ATPL protocol is based on the protocols proposed in [Wang et al., 2011] and [Noh et al., 2008]. The main goal of [Noh et al., 2008] was synchronization and did not focus on ranging (distance between the nodes was assumed to be known). The algorithm proposed in [Wang et al., 2011] also exploits the broadcast property and focused on localization of a target node in an asynchronous network, however, estimation of the clock parameters was not explicitly considered. In the ATPL protocol, during communication between a pair of nodes, time-stamps are recorded and exchanged. Besides this, the remaining nodes in the network also passively listen and record the time-stamps with their respective clocks, in a cooperative way. However, they do not respond back to either of the active node pair. In addition, the protocol does not put any constraint on the sequence of transmissions, and this together with passive listening results in asymmetrical links, and hence, asymmetrical time-stamps. The ATPL protocol is energy-efficient in the sense that we obtain more information just by passive listening, and reception usually consumes less power than transmission. For a fully asynchronous network with one sensor and M anchors, we propose a least squares estimator based on the time stamps recorded using the ATPL protocol for jointly estimating all the unknown clock skews, clock offsets, and pairwise distances of the sensor to each anchor.

7.2 System model

We consider a fully asynchronous network with M anchors (nodes with known relative locations) and one sensor (*node* 0). We assume that one of the nodes has a relatively stable clock oscillator and is used as a clock reference. All the other nodes suffer from clock skews and clock offsets. The network model considered here is a special case of the model in [Rajan and van der Veen, 2011], as

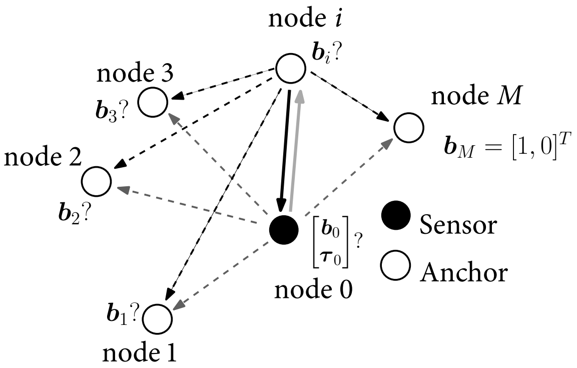


Figure 7.1: The ATPL protocol with the ith anchor transmitting. Solid (dotted) lines refer to the active (passive) links. Dark (light) shaded lines refer to the forward (reverse) link.

the pairwise distances of certain nodes (anchors) are now assumed to be known.

The distance between the ith and the jth node is denoted by $d_{i,j} = d_{j,i}$. The distance between the sensor and the ith anchor is denoted by $d_{0,i} = d_{i,0}$, and is unknown. Let t_i be the local time at the ith node and t be the reference time. We assume that the relation between the local time and the reference time can be given by a first order affine clock model [Rajan and van der Veen, 2011],

$$t_i = \omega_i t + \phi_i \quad \Leftrightarrow \quad t = \alpha_i t_i + \beta_i \tag{7.1}$$

where $\omega_i \in \mathbb{R}_+$ is the clock skew, $\phi_i \in \mathbb{R}$ is the clock offset, $\alpha_i = \omega_i^{-1}$ and $\beta_i = -\omega_i^{-1}\phi_i$ are the synchronization parameters of the ith node. Without loss of generality, we assume that *node M* has a stable clock, i.e., $[\omega_M, \phi_M] = [1, 0]$. The unknown synchronization parameters are collected in $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \ldots, \alpha_{M-1}]^T$ and $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_{M-1}]^T$. The unknown clock skews and clock offsets are, respectively, given by

$$\boldsymbol{\omega} = \mathbf{1}_M \oslash \boldsymbol{\alpha} \quad \text{and} \quad \boldsymbol{\phi} = -\boldsymbol{\beta} \oslash \boldsymbol{\alpha}. \tag{7.2}$$

The transmission and reception time-stamps are recorded both during the forward link (ith active anchor to the sensor) and the reverse link (sensor to the ith active anchor). The time-stamp recorded at the ith node when the kth iteration message departs is denoted by $T_i^{(k)}$, and on arrival of the corresponding message, the jth node records the time-stamp $R_{i,j}^{(k)}$. Note that the time-stamps recorded at the sensor will be either $T_0^{(k)}$ or $R_{i,0}^{(k)}$.

7.3 Passive listening protocol

In the two-way time-stamp exchange protocol between the ith anchor and the sensor, the remaining nodes of the network are idle. In the ATPL protocol, we propose that all the remaining $M-1$ anchors passively listen to the communication between the ith anchor and the sensor, and record the time-stamps $R_{i,j\neq 0}^{(k)}$ of their respective local clocks. By doing so, we obtain more information with extra

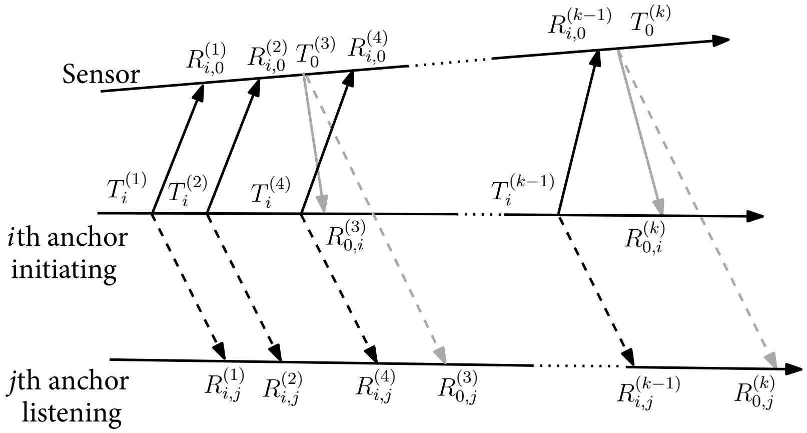


Figure 7.2: An example sequence of the recorded time-stamps. Solid (dotted) lines refer to the active (passive) links. Dark (light) shaded lines refer to the forward (reverse) link.

equations corresponding to transmissions between a) active anchor and other passive anchors, and b) sensor and remaining passive anchors. This is additional to the equations corresponding to the active anchor sensor pair as compared to the two-way time-stamp exchange. The ATPL protocol initiated by the ith anchor is illustrated in Fig. 7.1. An illustration of the sequence of time-stamps recorded during the ATPL protocol is shown in Fig. 7.2.

In the proposed protocol, we do not put any constraints on the sequence of forward links and reverse links [Rajan and van der Veen, 2011], i.e., the reverse link need not always follow the forward link as in [Freris et al., 2010, Wu et al., 2011, Noh et al., 2008]. This means that the sensor need not respond to the request from the anchor immediately. Therefore the processing time at the sensor or the network delay typically considered in clock synchronization algorithms [Freris et al., 2010, Wu et al., 2011, Wang et al., 2011, Noh et al., 2008] need not be taken into account as long as the clock parameters are stable within certain reasonable limits.

Remark 7.1. *(Protocol modes): Possible ways of executing the ATPL protocol are **mode a)** Anchor node i makes K_i transmissions and the sensor replies back with K_0 messages, with K_0 not necessarily equal to K_i and the transmissions need not be sequential. This is repeated by all the remaining anchors; **mode b)** each anchor node i makes K_i transmissions, and, in the end the sensor replies only once with K_0 messages; and **mode c)** m-out-of-M anchors $(m \leq M)$ make transmissions, and the sensor replies as described in either mode a or mode b.*

A suitable protocol mode can be adopted depending on the performance requirement and the energy constraint per node.

Remark 7.2. *(Centralized or distributed): The computation can be done in a centralized way in a fusion center (FC). However, there is an involved communication load in transmitting the time-stamps recorded at each node to a FC.*

An FC based approach can be avoided by including the time-stamps $R_{i,0}^{(k)}$, $k = 1, 2, \ldots, K_i$ in the payload when the sensor responds to the ith anchor. However, additional broadcast messages to distribute the time-stamps a) $R_{0,i}^{(k)}$, b) $R_{i,j}^{(k)}$ and $R_{0,j}^{(k)}$ are still required. This approach would avoid transmission of the computed unknown parameters to the nodes, that is required with an FC based approach. Moreover, it allows each node to independently perform computations in a distributed fashion.

7.4 Estimator

The time-of-flight for a line-of-sight (LOS) transmission between the ith and the jth node can be defined as $\tau_{i,j} = v^{-1} d_{i,j}$, where v denotes the speed of a wave (electromagnetic or acoustic) in a medium. Using (7.1), $\tau_{i,j}$ can be written in terms of the time-stamps recorded using respective local clocks of the ith and jth node as

$$\tau_{i,j} = (\alpha_j R_{i,j}^{(k)} + \beta_j) - (\alpha_i T_i^{(k)} + \beta_i) + n_{i,j}^{(k)} \tag{7.3}$$

where $n_{i,j}^{(k)}$ denotes the aggregate measurement error on the time-stamps.

The transmission and reception time-stamps recorded at the ith and the jth node are, respectively, collected in vectors

$$\boldsymbol{t}_i = [T_i^{(1)}, T_i^{(2)}, \ldots, T_i^{(K_i)}]^T \in \mathbb{R}^{K_i \times 1}$$

and

$$\boldsymbol{r}_{i,j} = [R_{i,j}^{(1)}, R_{i,j}^{(2)}, \ldots, R_{i,j}^{(K_i)}]^T \in \mathbb{R}^{K_i \times 1},$$

where K_i is the number of transmissions made by the ith node. The error vector is denoted by $\boldsymbol{n}_{i,j} = [n_{i,j}^{(1)}, n_{i,j}^{(2)}, \ldots, n_{i,j}^{(K_i)}]^T \in \mathbb{R}^{K_i \times 1}$.

For the sake of exposition, we consider a network with one sensor (*node* 0) and $M = 2$ anchors (*node* 1 and *node* 2) and the following example protocol: (i) *node* 1 makes K_1 transmissions, *node* 0 and *node* 2 passively listen, (ii) *node* 2 makes K_2 transmissions, *node* 0 and *node* 1 passively listen, and finally (iii) sensor *node* 0 responds with K_0 messages and *node* 1 and *node* 2 passively listen. This is an example of protocol *mode b* that was described earlier.

Collecting the clock parameters of the ith node in a vector $\boldsymbol{b}_i = [\alpha_i, \beta_i]^T$, we can now write the equations of the form given in (7.3), obtained for all the $K = K_0 + K_1 + K_2$ time-stamps recorded in a matrix-vector form as

$$\begin{array}{ll} \text{Node 0 responds} & j=1 \\ (i=0) & j=2 \\ \hdashline \text{Node 1 transmits} & j=0 \\ (i=1) & j=2 \\ \hdashline \text{Node 2 transmits} & j=0 \\ (i=2) & j=1 \end{array} \overbrace{\left[\begin{array}{cc|cc|cc|c|c|c} \boldsymbol{t}_0 & \mathbf{1}_{K_0} & -\boldsymbol{r}_{0,1} & -\mathbf{1}_{K_0} & \mathbf{0}_{K_0} & \mathbf{0}_{K_0} & \mathbf{1}_{K_0} & \mathbf{0}_{K_0} & \mathbf{0}_{K_0} \\ \boldsymbol{t}_0 & \mathbf{1}_{K_0} & \mathbf{0}_{K_0} & \mathbf{0}_{K_0} & -\boldsymbol{r}_{0,2} & -\mathbf{1}_{K_0} & \mathbf{0}_{K_0} & \mathbf{1}_{K_0} & \mathbf{0}_{K_0} \\ \hdashline -\boldsymbol{r}_{1,0} & -\mathbf{1}_{K_1} & \boldsymbol{t}_1 & \mathbf{1}_{K_1} & \mathbf{0}_{K_1} & \mathbf{0}_{K_1} & \mathbf{1}_{K_1} & \mathbf{0}_{K_1} & \mathbf{0}_{K_1} \\ \mathbf{0}_{K_1} & \mathbf{0}_{K_1} & \boldsymbol{t}_1 & \mathbf{1}_{K_1} & -\boldsymbol{r}_{1,2} & -\mathbf{1}_{K_1} & \mathbf{0}_{K_1} & \mathbf{0}_{K_1} & \mathbf{1}_{K_1} \\ \hdashline -\boldsymbol{r}_{2,0} & -\mathbf{1}_{K_2} & \mathbf{0}_{K_2} & \mathbf{0}_{K_2} & \boldsymbol{t}_2 & \mathbf{1}_{K_2} & \mathbf{0}_{K_2} & \mathbf{1}_{K_2} & \mathbf{0}_{K_2} \\ \mathbf{0}_{K_2} & \mathbf{0}_{K_2} & -\boldsymbol{r}_{2,1} & -\mathbf{1}_{K_2} & \boldsymbol{t}_2 & \mathbf{1}_{K_2} & \mathbf{0}_{K_2} & \mathbf{0}_{K_2} & \mathbf{1}_{K_2} \end{array}\right]}^{\boldsymbol{A}\in\mathbb{R}^{2K\times 9},\ K=K_0+K_1+K_2} \begin{bmatrix} \boldsymbol{b}_0 \\ \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \tau_{0,1} \\ \tau_{0,2} \\ \tau_{1,2} \end{bmatrix}$$

$$= \overbrace{\left[\begin{array}{c} \boldsymbol{n}_{0,1} \\ \boldsymbol{n}_{0,2} \\ \hdashline \boldsymbol{n}_{1,0} \\ \boldsymbol{n}_{1,2} \\ \hdashline \boldsymbol{n}_{2,0} \\ \boldsymbol{n}_{2,1} \end{array}\right]}^{\boldsymbol{n}\,\in\mathbb{R}^{2K\times 1}} \begin{array}{l} \left.\vphantom{\begin{matrix}a\\a\end{matrix}}\right\}\boldsymbol{n}_0 \\ \left.\vphantom{\begin{matrix}a\\a\end{matrix}}\right\}\boldsymbol{n}_1 \\ \left.\vphantom{\begin{matrix}a\\a\end{matrix}}\right\}\boldsymbol{n}_2 \end{array} . \tag{7.4}$$

The ordering of the rows of the system matrix $\boldsymbol{A}$ is arbitrary and does not imply the order of transmission. The columns of $\boldsymbol{A}$ corresponding to $\tau_{0,1}$, $\tau_{0,2}$, and $\tau_{1,2}$ have two nonzero subvectors each as $\tau_{i,j} = \tau_{j,i}$.

Remark 7.3. *(Rank-deficiency): The linear model in (7.4) does not have a unique solution, unless we impose certain constraints. Here, we do that by assigning node* 2 *as the clock reference, i.e.,* $\boldsymbol{b}_2 = [1, 0]^T$.

We define the vector $\boldsymbol{\tau}_0 = [\tau_{0,1}, \tau_{0,2}, \ldots, \tau_{0,M}]^T \in \mathbb{R}^{M\times 1}$, where the entries of $\boldsymbol{\tau}_0$ are not known. Note that $\tau_{1,2} = v^{-1}d_{1,2}$ corresponds to the distance between the nodes 1 and 2, and is known. Moving all the knowns (columns corresponding to $\boldsymbol{b}_2$ and $\tau_{1,2}$) to one side, (7.4) simplifies to the generalized linear model given as

$$\begin{array}{ll} \text{Node 0 responds} & j=1 \\ (i=0) & j=2 \\ \hdashline \text{Node 1 transmits} & j=0 \\ (i=1) & j=2 \\ \hdashline \text{Node 2 transmits} & j=0 \\ (i=2) & j=1 \end{array} \overbrace{\left[\begin{array}{cc|cc|cc} \boldsymbol{t}_0 & \mathbf{1}_{K_0} & -\boldsymbol{r}_{0,1} & -\mathbf{1}_{K_0} & \mathbf{1}_{K_0} & \mathbf{0}_{K_0} \\ \boldsymbol{t}_0 & \mathbf{1}_{K_0} & \mathbf{0}_{K_0} & \mathbf{0}_{K_0} & \mathbf{0}_{K_0} & \mathbf{1}_{K_0} \\ \hdashline -\boldsymbol{r}_{1,0} & -\mathbf{1}_{K_1} & \boldsymbol{t}_1 & \mathbf{1}_{K_1} & \mathbf{1}_{K_1} & \mathbf{0}_{K_1} \\ \mathbf{0}_{K_1} & \mathbf{0}_{K_1} & \boldsymbol{t}_1 & \mathbf{1}_{K_1} & \mathbf{0}_{K_1} & \mathbf{0}_{K_1} \\ \hdashline -\boldsymbol{r}_{2,0} & -\mathbf{1}_{K_2} & \mathbf{0}_{K_2} & \mathbf{0}_{K_2} & \mathbf{0}_{K_2} & \mathbf{1}_{K_2} \\ \mathbf{0}_{K_2} & \mathbf{0}_{K_2} & -\boldsymbol{r}_{2,1} & -\mathbf{1}_{K_2} & \mathbf{0}_{K_2} & \mathbf{0}_{K_2} \end{array}\right]}^{\bar{\boldsymbol{A}}\in\mathbb{R}^{2K\times 6}} \overbrace{\begin{bmatrix} \boldsymbol{b}_0 \\ \boldsymbol{b}_1 \\ \boldsymbol{\tau}_0 \end{bmatrix}}^{\boldsymbol{\theta}\in\mathbb{R}^{6\times 1}}$$

$$= -\overbrace{\left[\begin{array}{cc|c} \mathbf{0}_{K_0} & \mathbf{0}_{K_0} & \mathbf{0}_{K_0} \\ -\boldsymbol{r}_{0,2} & -\mathbf{1}_{K_0} & \mathbf{0}_{K_0} \\ \hdashline \mathbf{0}_{K_1} & \mathbf{0}_{K_1} & \mathbf{0}_{K_1} \\ -\boldsymbol{r}_{1,2} & -\mathbf{1}_{K_1} & \mathbf{1}_{K_1} \\ \hdashline \boldsymbol{t}_2 & \mathbf{1}_{K_2} & \mathbf{0}_{K_2} \\ \boldsymbol{t}_2 & \mathbf{1}_{K_2} & \mathbf{1}_{K_2} \end{array}\right] \begin{bmatrix} \boldsymbol{b}_2 \\ \tau_{1,2} \end{bmatrix}}^{\boldsymbol{x}\in\mathbb{R}^{2K\times 1}} + \boldsymbol{n}. \tag{7.5}$$

The generalization of the data model (7.5) for any $M > 2$ is straightforward and can be easily derived along the same lines. The generalized linear model based on the ATPL protocol is given by

$$\bar{\boldsymbol{A}}\boldsymbol{\theta} = \boldsymbol{x} + \boldsymbol{n} \tag{7.6}$$

where $\bar{\boldsymbol{A}} \in \mathbb{R}^{KM\times 3M}$, $\boldsymbol{\theta} \in \mathbb{R}^{3M\times 1}$, $\boldsymbol{x} \in \mathbb{R}^{KM\times 1}$ and $\boldsymbol{n} \in \mathbb{R}^{KM\times 1}$, all having a similar structure as that of (7.5).

Remark 7.4. *(Correlated error vector): In case of broadcasting, the entries of the error vector* $\boldsymbol{n}$ *are not uncorrelated due to a common error on the transmit time-stamp* $T_i^{(k)}$*.*

We assume that the aggregate error $n_{i,j}^{(k)}$ in (7.3) is due to the additive stochastic noise components on the time-stamps, $T_i^{(k)}$ denoted by $e_i^{(k)}$ and the time-stamps, $R_{i,j}^{(k)}$ denoted by $e_{i,j}^{(k)}$. We model the aggregate error in (7.3) as

$$n_{i,j}^{(k)} = e_i^{(k)} + e_{i,j}^{(k)} \tag{7.7}$$

where both $e_i^{(k)}$ and $e_{i,j}^{(k)}$ are modeled as zero mean i.i.d. Gaussian [Patwari et al., 2005] with variance $0.5\sigma^2$, such that, $\mathbb{E}\left\{e_i^{(k)} e_{i,j}^{(k)}\right\} = 0$ for $i \neq j$. (This is a simplified noise model and more accurate models could be considered.)

We can compute the noise covariance matrix as $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\Sigma}_0, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_M) \in \mathbb{R}^{MK\times MK}$, where $\boldsymbol{\Sigma}_i = \mathbb{E}\left\{\boldsymbol{n}_i \boldsymbol{n}_i^T\right\}$. For $M = 2$, we find

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma^2 \boldsymbol{I}_{K_i} & 0.5\sigma^2 \boldsymbol{I}_{K_i} \\ 0.5\sigma^2 \boldsymbol{I}_{K_i} & \sigma^2 \boldsymbol{I}_{K_i} \end{bmatrix} \in \mathbb{R}^{2K_i\times 2K_i} \tag{7.8}$$

The structure of $\boldsymbol{\Sigma}$ can be generalized for any $M > 2$ in a similar way, leading to $\boldsymbol{\Sigma}_i = 0.5\sigma^2(\boldsymbol{1}_M\boldsymbol{1}_M^T + \boldsymbol{I}_M) \otimes \boldsymbol{I}_{K_i} \in \mathbb{R}^{MK_i\times MK_i}$.

We can now prewhiten the observation model in (7.5) by forming $\boldsymbol{\Sigma}^{-1/2}\bar{\boldsymbol{A}}$ and $\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}$. For $K \geq 3$, $\bar{\boldsymbol{A}}$ is tall and is left-invertible. Hence, the unknown parameters in $\boldsymbol{\theta}$ can be estimated using least squares, i.e.,

$$\widehat{\boldsymbol{\theta}} = (\bar{\boldsymbol{A}}^T\boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{A}})^{-1}\bar{\boldsymbol{A}}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}. \tag{7.9}$$

Subsequently, the clock skews $\boldsymbol{\omega}$, clock offsets $\boldsymbol{\phi}$ can be obtained using the relation in (7.2), and the pairwise distances of the sensor to each anchor using the relation $\boldsymbol{d}_0 = v\boldsymbol{\tau}_0$.

Remark 7.5. *(Sensor does not respond):*

When only one of the nodes transmits, say node 1*,* $\bar{\boldsymbol{A}}$ *in (7.5) will not have rows corresponding to transmissions of node* 0 *and node* 2*, and it is rank-deficient as the columns two and five are dependent. This also holds when only either node* 2 *or node* 0 *transmits.*

If only anchor nodes transmit, and the sensor does not respond, then $\bar{\boldsymbol{A}}$ *will not have rows corresponding to transmissions of node* 0*. In that case,* $\bar{\boldsymbol{A}}$ *will be again rank-deficient, as column two is a linear combination of columns five and six. Therefore, for (7.6) to have a unique solution, the sensor should respond at least once, i.e.,* $K_0 = 1$ *with* $K \geq 3$.

The possibility that the sensor node responds with only one message in the end makes the protocol energy-efficient.

In sum, the proposed algorithm takes into account: (1) known distances between anchors, and (2) the broadcast property, which results in additional observations and a correlated error as compared to the pairwise transmissions without passive listening.

7.5 Cramér-Rao lower bound

It follows from the Cramér-Rao lower bound (CRB) theorem that the covariance of any unbiased estimate $\widehat{\bar{\boldsymbol{\theta}}}$ of the unknown parameter $\bar{\boldsymbol{\theta}}$ satisfies the well-known inequality [Kay, 1993]

$$\mathbb{E}\left\{(\bar{\boldsymbol{\theta}} - \widehat{\bar{\boldsymbol{\theta}}})(\bar{\boldsymbol{\theta}} - \widehat{\bar{\boldsymbol{\theta}}})^T\right\} \geq \boldsymbol{F}^{-1},$$

where $\bar{\boldsymbol{\theta}} = [\boldsymbol{\omega}^T, \boldsymbol{\phi}^T, \boldsymbol{d}_0^T]^T$ and $\boldsymbol{F} \in \mathbb{R}^{3M \times 3M}$ is the Fisher information matrix. If the error $\boldsymbol{n}$ in (7.6) is zero-mean Gaussian with covariance matrix $\boldsymbol{\Sigma}$, then $\boldsymbol{F}$ can be computed as $\boldsymbol{F} = \boldsymbol{J}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{J}$, where $\boldsymbol{J}$ is a Jacobian matrix. The Jacobian matrix is given by

$$\boldsymbol{J} = \frac{\partial(\bar{\boldsymbol{A}}\boldsymbol{\theta} - \boldsymbol{x})}{\partial \bar{\boldsymbol{\theta}}^T} = [\boldsymbol{J}_{\boldsymbol{\omega}} \quad \boldsymbol{J}_{\boldsymbol{\phi}} \quad \boldsymbol{J}_{\boldsymbol{d}_0}] \in \mathbb{R}^{KM \times 3M} \tag{7.10}$$

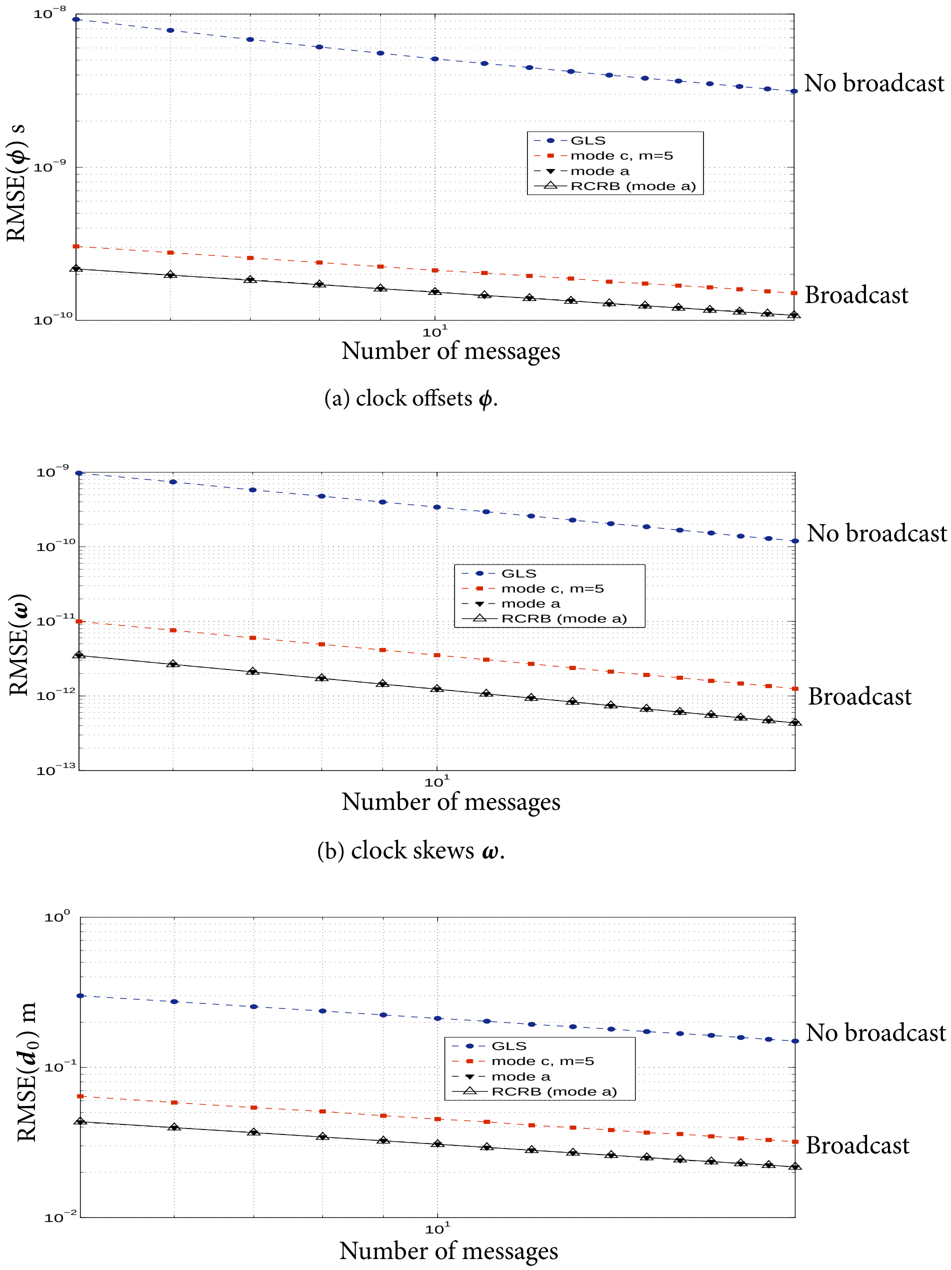
with sub-blocks

$$\begin{aligned}
\boldsymbol{J}_{\boldsymbol{\omega}} &= \frac{\partial(\bar{\boldsymbol{A}}\boldsymbol{\theta} - \boldsymbol{x})}{\partial \boldsymbol{\omega}^T} = -(\bar{\boldsymbol{A}}\boldsymbol{S}_{\boldsymbol{\alpha}} - \bar{\boldsymbol{A}}\boldsymbol{S}_{\boldsymbol{\beta}} \odot \mathbf{1}_{KM}\boldsymbol{\phi}^T) \oslash (\mathbf{1}_{KM}\boldsymbol{\omega}^T)^{\odot 2}, \\
\boldsymbol{J}_{\boldsymbol{\phi}} &= \frac{\partial(\bar{\boldsymbol{A}}\boldsymbol{\theta} - \boldsymbol{x})}{\partial \boldsymbol{\phi}^T} = -\bar{\boldsymbol{A}}\boldsymbol{S}_{\boldsymbol{\beta}} \oslash \mathbf{1}_{KM}\boldsymbol{\omega}^T, \\
\boldsymbol{J}_{\boldsymbol{d}_0} &= \frac{\partial(\bar{\boldsymbol{A}}\boldsymbol{\theta} - \boldsymbol{x})}{\partial \boldsymbol{d}_0^T} = \nu^{-1}\bar{\boldsymbol{A}}\boldsymbol{S}_{\boldsymbol{\tau}_0}
\end{aligned}$$

where $\boldsymbol{S}_{\boldsymbol{\alpha}}$, $\boldsymbol{S}_{\boldsymbol{\beta}}$, and $\boldsymbol{S}_{\boldsymbol{\tau}_0}$ are selection matrices to select the columns of $\bar{\boldsymbol{A}}$ corresponding to $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\tau}_0$, respectively.

7.6 Simulations

A network with one sensor and 10 anchors is considered for simulations. Both the sensor and the anchor nodes are deployed randomly within a range of 100 m. Clock skews $\boldsymbol{\omega}$ and clock offsets $\boldsymbol{\phi}$ are uniformly distributed in the range $[1 - 100\,\text{ppm}, 1 + 100\,\text{ppm}]$ and $[-1\,\text{s}, 1\,\text{s}]$, respectively. We use an observation interval of 100 s during which the clock parameters are fixed and $\nu = 3 \times 10^8$ m/s. The time-stamps are corrupted with an i.i.d. Gaussian process having a standard deviation $\sigma = 1$ ns [Patwari et al., 2005].



(a) clock offsets $\boldsymbol{\phi}$.

(b) clock skews $\boldsymbol{\omega}$.

(c) pairwise distances $\boldsymbol{d}_0$.

Figure 7.3: RMSE of the estimated unknown parameters.

The proposed estimator based on the ATPL protocol is compared with the global least squares (GLS) algorithm proposed in [Rajan and van der Veen, 2011, Fig. 3(c)], as it is already shown to outperform other existing synchronization algorithms. We apply the GLS algorithm based on two-way communication between each sensor-anchor pair.

Fig. 7.3 shows the root mean square error (RMSE) of the estimates $\boldsymbol{\phi}$ and $\boldsymbol{\omega}$, and $\boldsymbol{d}_0$ for different number of messages, K. We show simulations for *mode a* and *mode c* of the ATPL protocol described in Section 7.3. It can be seen from the figures that the proposed algorithm performs better than GLS in both the considered scenarios due the additional links obtained from passive listening. The proposed algorithm also achieves the theoretical root CRB (RCRB).

7.7 Discussion

In this chapter, we have considered a fully asynchronous network with one sensor and M anchors. We have proposed a least squares estimator to synchronize the sample clocks of wireless sensors based on the ATPL protocol that fully exploits the broadcast nature of the wireless medium. We estimate all the unknown clock skews and clock offsets along with the pairwise distances of the sensor to each anchor. The proposed estimator is shown to be efficient, asymptotically meeting the theoretical CRB, and it outperforms available algorithms. Pairwise distances form a major input to any localization scheme as we see in the next chapter.

Part of this chapter was published as: S.P. Chepuri *et al.* Rigid Body Localization Using Sensor Networks. *IEEE Trans. on Signal Processing*, 62(18): 4911-4924, Sept. 2014.

Chapter 8

Rigid Body Localization

Contents

8.1 Introduction

In this chapter, we provide a different flavor of localization, called *rigid body localization*. In rigid body localization, we use a few sensors on a rigid sensing platform and exploit the knowledge of how the sensors are mounted on the body (i.e., sensor placement on the platform) to jointly estimate the position as well as the orientation of the rigid body based on distance measurements.

8.1.1 Applications and prior works

Rigid body localization has potential applications in a variety of fields. To list a few, it is useful for location services involving underwater (or in-liquid) systems, orbiting satellites, mechatronic systems, aircrafts, underwater vehicles, ships, robotic systems, or automobiles. In such applications, classical localization of the node(s) is not sufficient. For example, in an autonomous underwater vehicle (AUV), or an orbiting satellite, the sensing platform is not only subject to motion but also to rotation. Hence, next to position, determining the orientation of the body also forms a key component, and is essential for controlling, maneuvering, and monitoring purposes.

The orientation is sometimes referred to as attitude (aerospace applications) or tilt (for industrial equipments and consumer devices). Traditionally, position and orientation are treated separately even though they are closely related. The orientation of a body is usually measured using inertial measurement units (IMUs) comprised of accelerometers [Salhuana, 2012] and gyroscopes. However, IMUs generally suffer from accumulated errors often referred to as drift errors. The drift calibration is typically achieved using different sensor technologies including vision, magnetometers, ultra wide band (UWB), or GPS [Hol et al., 2009, Hol, 2011], leading to dependencies between these technologies. Sometimes these different sensors cannot be coherently fused, for instance magnetometer based calibration needs an undistorted magnetic environment, which is typically difficult to guarantee.

GPS-based attitude determination [Wahba, 1965, Cohen, 1992, Juang and Huang, 1997] is closely related to our work, in which multiple antennas on a platform are used. Here, the attitude is estimated from GPS carrier phase measurements which involves a complicated integer problem with no unique solution in general.

8.1.2 Contributions

We propose a framework for joint position and orientation estimation of a rigid body in a three-dimensional space by borrowing techniques from classical sensor localization, i.e., using only *range* measurements between all the sensor-anchor pairs. We consider a rigid body on which a few sensor nodes are mounted. The absolute position of the rigid body is not known. However, the topology of how the sensors are mounted on the rigid body is known up to a certain accuracy. The orientation of the rigid body is expressed as a *rotation matrix* and the absolute position of the rigid body (instead of the absolute position of each individual sensor) as a *translation* vector. In other words, the absolute position of the sensors is expressed as an affine function of the Stiefel manifold.

The maximum likelihood (ML) estimators for the original problem involve solving a constrained (nonconvex) nonlinear least-squares (NLS) problem, which is in general difficult to solve. In order to simplify this problem, we linearize the problem by squaring the measurements. We use the linearized model in a least-squares (LS) estimator to jointly estimate the rotation matrix (to begin with, its structure is ignored) and the translation vector. Since rotation matrices are unitary matrices, we also propose a unitarily constrained least-squares (CLS) estimator and a simplified unitarily constrained least-squares (SCLS) estimator, both of which solve an optimization problem on the Stiefel manifold. The solutions from the proposed estimators can be used as an initialization to solve the maximum-likelihood estimators or the original nonlinear LS problem if needed. We also derive a unitarily constrained Cramér-Rao bound (CCRB), which is used as a benchmark for the proposed estimators.

In many applications, the sensor topology might not be accurately known, i.e., the known topology can be noisy. These perturbations are typically introduced while mounting the sensors during fabrication or if the body is not entirely rigid. To account for such perturbations, we propose a unitarily constrained total-least-squares (CTLS) estimator and a simplified unitarily constrained total-least-squares (SCTLS) estimator. The performance of the proposed estimators is analyzed using simulations. Using a sensor array with a known geometry not only enables orientation estimation, but also yields a better localization performance. The initial results on rigid body localization using range measurements, viz., SCLS and SCTLS were proposed in [Chepuri et al., 2013a].

The proposed framework of rigid body localization can also be used as an add-on to the existing IMU based systems to correct the drift errors, or in environments where inertial measurements and/or positioning via GPS is not feasible. The proposed framework is based on a static position and orientation, unlike most of the orientation estimators which are based on inertial measurements and a certain dynamical state-space model (e.g., see [Hol et al., 2009]). Hence, our approach is useful when there is no dynamic model available. We should stress, however, that the proposed framework is also suitable for the estimation (tracking) of dynamic position and orientation using a state-constrained Kalman filter for instance, and some initial results on this topic can be found in [Chepuri et al., 2013d].

8.2 Problem formulation and modeling

8.2.1 Problem formulation

Consider a network with M anchors (nodes with known absolute locations) and N sensors in a 3-dimensional space. The sensors are mounted on a rigid body as illustrated in Fig. 8.1. The wireless sensors are mounted on the rigid body (e.g., at the factory), and the topology of how these sensors are mounted is known up to a certain accuracy. In other words, we connect a so-called *reference frame* to the rigid body, as illustrated in Fig. 8.1, and in that reference frame, the coordinates of the nth sensor are given by the known 3×1 vector $\boldsymbol{c}_n = [c_{n,1}, c_{n,2}, c_{n,3}]^T$. The sensor topology is basically determined by the matrix $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_N] \in \mathbb{R}^{3\times N}$. Let the absolute coordinates of the mth anchor and the nth sensor be denoted by a 3×1 vector $\boldsymbol{a}_m$ and $\boldsymbol{s}_n$, respectively, where $\boldsymbol{s}_n$ is not known. The absolute positions of the anchors and the sensors are collected in the matrices $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_M] \in \mathbb{R}^{3\times M}$ and $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \dots, \boldsymbol{s}_N] \in \mathbb{R}^{3\times N}$, respectively.

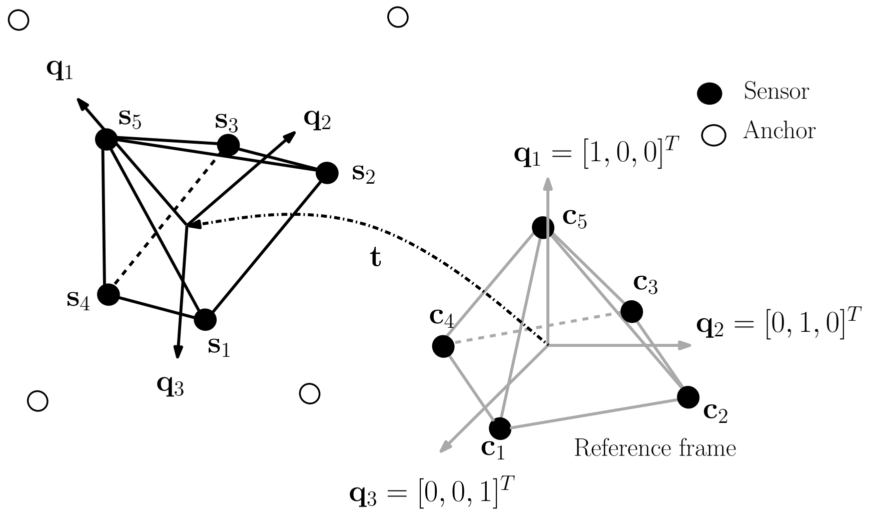


Figure 8.1: An illustration of the sensors on a rigid body undergoing a rotation and translation.

Rigid body transformation

A Stiefel manifold [Eldén and Park, 1999] in three dimensions, denoted by $\mathcal{V}_{3,3}$, is the set of all 3 × 3 unitary matrices $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3] \in \mathbb{R}^{3\times 3}$, i.e.,

$$\mathcal{V}_{3,3} = \{\boldsymbol{Q} \in \mathbb{R}^{3\times 3} : \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}_3\}.$$

The absolute position of the nth sensor can be written as an affine function of a point on the Stiefel manifold, i.e.,

$$\begin{aligned} \boldsymbol{s}_n &= c_{n,1}\boldsymbol{q}_1 + c_{n,2}\boldsymbol{q}_2 + c_{n,3}\boldsymbol{q}_3 + \boldsymbol{t} \\ &= \boldsymbol{Q}\boldsymbol{c}_n + \boldsymbol{t}, \end{aligned} \tag{8.1}$$

where $\boldsymbol{t} \in \mathbb{R}^{3\times 1}$ denotes the unknown translation. More specifically, the parameter vector $\boldsymbol{t}$ refers to the unknown position of the rigid body. The combining weights $\boldsymbol{c}_n$ are the *known* coordinates of the nth sensor in the reference frame. This means that the unknown unitary matrix $\boldsymbol{Q}$ actually tells us how the rigid body has rotated in the reference frame. When there is no rotation, then we have $\boldsymbol{Q} = \boldsymbol{I}_3$. The relation in (8.1) is sometimes also referred to as the *rigid body transformation*. The rotation matrices can uniquely represent the orientation of a rigid body unlike Euler angles or unit quaternions (see [Chaturvedi et al., 2011] for more details). The rigid body transformation is also used in computer vision applications [Arun, 1992, Horn et al., 1988, Horn, 1987].

With (8.1), the absolute position of all the sensors can be written as

$$\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{C} + \boldsymbol{t}\boldsymbol{1}_N^T = \overbrace{\left[\begin{array}{c|c} \boldsymbol{Q} & \boldsymbol{t} \end{array}\right]}^{\boldsymbol{\Theta}} \overbrace{\left[\begin{array}{c} \boldsymbol{C} \\ \hline \boldsymbol{1}_N^T \end{array}\right]}^{\boldsymbol{C}_{\mathrm{e}}}, \tag{8.2}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{3\times4}$ is the unknown transformation matrix.

Range measurements

The proposed framework is general and can be applied to range estimates obtained from any one of the standard ranging techniques (e.g., two-way ranging based on TOA measurements [Patwari et al., 2005, Chepuri et al., 2013c]). The framework is valid as long as the range estimates between all the sensor-anchor pairs are available. Further, we assume that the body position is nearly static during the ranging process, i.e., the linear and angular velocities are negligible compared to the propagation speed.

Let the range (or the Euclidean distance) between the mth anchor and the nth sensor be denoted by $\rho_{mn} = \|\boldsymbol{a}_m - \boldsymbol{s}_n\|_2$. The noisy range measurement between the mth anchor and the nth sensor can be expressed as

$$\begin{aligned} y_{mn} &= \|\boldsymbol{a}_m - \boldsymbol{s}_n\|_2 + v_{mn} \\ &= \|\boldsymbol{a}_m - (\boldsymbol{Q}\boldsymbol{c}_n + \boldsymbol{t})\|_2 + v_{mn}, \end{aligned} \tag{8.3}$$

where $v_{mn} \sim \mathcal{N}(0, \sigma_{mn}^2)$ is the stochastic noise resulting from the ranging process. The ranging noise $v_{mn}, m = 1, 2, \dots, M$; $n = 1, 2, \dots, N$, is a sequence of independent random variables whose variance σ_{mn}^2 is assumed to be known or easily estimated.

Problem statement

Having introduced the rigid body transformation in (8.1) and the measurement model (8.3) we can now formally state the rigid body localization problem as follows.

Problem 8.1 (Rigid body localization). *Given one range measurement between each sensor-anchor pair, i.e., y_{mn} as in* (8.3)*, the ranging noise variance σ_{mn}^2, for $m = 1, 2, \dots, M$ and $n = 1, 2, \dots, N$, the positions of the anchors $\boldsymbol{A}$, and the topology of the sensors on the rigid body determined by the matrix $\boldsymbol{C}$, jointly estimate the position $\boldsymbol{t} \in \mathbb{R}^{3\times1}$ and orientation $\boldsymbol{Q} \in \mathcal{V}_{3,3}$ of the rigid body.*

The ML estimator for jointly estimating the orientation and translation is to solve the following optimization problem

$$\underset{\boldsymbol{Q},\boldsymbol{t}}{\arg\min} \sum_{m=1}^{M}\sum_{n=1}^{N} \sigma_{mn}^{-2}(y_{mn} - \|\boldsymbol{a}_m - (\boldsymbol{Q}\boldsymbol{c}_n + \boldsymbol{t})\|_2)^2 \tag{8.4a}$$

$$\text{s.to} \quad \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_3. \tag{8.4b}$$

The above problem is a nonlinear and a nonconvex optimization problem, and is in general difficult to solve. To simplify this problem, we next linearize the model in (8.3), which can then be solved using linear LS based estimators. The solution from the proposed estimators can then be used as an initialization to solve the above NLS problem if needed.

8.2.2 Squared-range measurements

The model in (8.3) is nonlinear in $\boldsymbol{s}_n$, $\boldsymbol{Q}$, and $\boldsymbol{t}$. Therefore, we linearize the nonlinear model in (8.3) by squaring it. Squaring the measurements in (8.3) results in a noise term with a nonnegative known mean[1] σ_{mn}^2. Subtracting that mean σ_{mn}^2 from the squared-range measurements between the mth anchor and the nth sensor, we obtain

$$\begin{aligned} d_{mn} &= y_{mn}^2 - \sigma_{mn}^2 \\ &= \|\boldsymbol{a}_m\|_2^2 - 2\boldsymbol{a}_m^T\boldsymbol{s}_n + \|\boldsymbol{s}_n\|_2^2 + n_{mn}, \end{aligned} \tag{8.5}$$

where

$$n_{mn} = 2\rho_{mn}v_{mn} + v_{mn}^2 - \sigma_{mn}^2 \tag{8.6}$$

is the new zero-mean noise term due to squaring. Collecting these new squared-range measurements between the nth sensor and all the anchors in

$$\boldsymbol{d}_n \quad = \quad [d_{1n}, d_{2n}, \dots, d_{Mn}]^T \in \mathbb{R}^{M\times 1},$$

we can write (8.5) in a vector form as

$$\boldsymbol{d}_n = \boldsymbol{\alpha} - 2\boldsymbol{A}^T\boldsymbol{s}_n + \|\boldsymbol{s}_n\|_2^2\mathbf{1}_M + \boldsymbol{n}_n, \tag{8.7}$$

where

$$\boldsymbol{\alpha} = [\|\boldsymbol{a}_1\|_2^2, \|\boldsymbol{a}_2\|_2^2, \dots, \|\boldsymbol{a}_M\|_2^2]^T \in \mathbb{R}^{M\times 1},$$

is known, and

$$\boldsymbol{n}_n = [n_{1n}, n_{2n}, \dots, n_{Mn}]^T \in \mathbb{R}^{M\times 1}.$$

Subtracting the knowns in (8.7) from the measurements, we arrive at

$$\boldsymbol{d}_n - \boldsymbol{\alpha} = -2\boldsymbol{A}^T\boldsymbol{s}_n + \|\boldsymbol{s}_n\|_2^2\mathbf{1}_M + \boldsymbol{n}_n. \tag{8.8}$$

We next eliminate the vector $\|\boldsymbol{s}_n\|_2^2\mathbf{1}_M$ in (8.8) using an isometry decomposition of the projection matrix

$$\boldsymbol{P}_M = \boldsymbol{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T = \boldsymbol{U}_M\boldsymbol{U}_M^T \in \mathbb{R}^{M\times M},$$

where $\boldsymbol{U}_M$ is an $M \times (M-1)$ matrix obtained by collecting orthonormal basis vectors of the null-space of $\mathbf{1}_M$ such that $\boldsymbol{U}_M^T\mathbf{1}_M = \mathbf{0}_{M-1}$. Pre-multiplying both sides of (8.8) with $\boldsymbol{U}_M^T$, we arrive at

$$\boldsymbol{U}_M^T(\boldsymbol{d}_n - \boldsymbol{\alpha}) = -2\boldsymbol{U}_M^T\boldsymbol{A}^T\boldsymbol{s}_n + \boldsymbol{U}_M^T\boldsymbol{n}_n. \tag{8.9}$$

Stacking (8.9) for all the N sensors, we obtain

$$\boldsymbol{U}_M^T\boldsymbol{D} = -2\boldsymbol{U}_M^T\boldsymbol{A}^T\boldsymbol{S} + \boldsymbol{U}_M^T\boldsymbol{N}, \tag{8.10}$$

[1] For low noise levels, this nonnegative mean which is simply the variance of the range error in (8.3) can be ignored.

where we define the following $M \times N$ matrices:

$$\begin{aligned} \boldsymbol{D} &= [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N] - \boldsymbol{\alpha} \mathbf{1}_N^T, \\ \text{and} \quad \boldsymbol{N} &= [\boldsymbol{n}_1, \boldsymbol{n}_2, \ldots, \boldsymbol{n}_N]. \end{aligned}$$

The linear model in (8.10) can then be compactly expressed as

$$\bar{\boldsymbol{D}} = \bar{\boldsymbol{A}} \boldsymbol{S} + \bar{\boldsymbol{N}}, \tag{8.11}$$

where we have introduced the following matrices:

$$\begin{aligned} \bar{\boldsymbol{D}} &= \boldsymbol{U}_M^T \boldsymbol{D} \in \mathbb{R}^{(M-1)\times N}, \\ \bar{\boldsymbol{A}} &= -2\boldsymbol{U}_M^T \boldsymbol{A}^T \in \mathbb{R}^{(M-1)\times 3}, \\ \text{and} \quad \bar{\boldsymbol{N}} &= \boldsymbol{U}_M^T \boldsymbol{N} \in \mathbb{R}^{(M-1)\times N}. \end{aligned}$$

Vectorizing[2] (8.11) leads to

$$\bar{\boldsymbol{d}} = \overbrace{(\boldsymbol{I}_N \otimes \bar{\boldsymbol{A}})}^{(M-1)N\times 3N} \boldsymbol{s} + \bar{\boldsymbol{n}}, \tag{8.12}$$

where

$$\begin{aligned} \boldsymbol{s} &= \text{vec}(\boldsymbol{S}) \in \mathbb{R}^{3N}, \\ \bar{\boldsymbol{d}} &= \text{vec}(\bar{\boldsymbol{D}}) \in \mathbb{R}^{(M-1)N}, \\ \text{and} \quad \bar{\boldsymbol{n}} &= \text{vec}(\bar{\boldsymbol{N}}) = (\boldsymbol{I}_N \otimes \boldsymbol{U}_M^T)\text{vec}(\boldsymbol{N}) \in \mathbb{R}^{(M-1)N}. \end{aligned}$$

Using the rigid body transformation in (8.2), we can relate the measurements $\bar{\boldsymbol{D}}$ in (8.11) and the transformation matrix $\boldsymbol{\Theta}$. Substituting (8.2) in (8.11), we arrive at the following linear model

$$\bar{\boldsymbol{D}} = \bar{\boldsymbol{A}} \boldsymbol{\Theta} \boldsymbol{C}_\text{e} + \bar{\boldsymbol{N}}, \tag{8.13}$$

which can then be vectorized to

$$\bar{\boldsymbol{d}} = \overbrace{(\boldsymbol{C}_\text{e}^T \otimes \bar{\boldsymbol{A}})}^{(M-1)N\times 9} \boldsymbol{\theta} + \bar{\boldsymbol{n}}, \tag{8.14}$$

where

$$\boldsymbol{\theta} = \text{vec}(\boldsymbol{\Theta}) = [\boldsymbol{q}_1^T, \boldsymbol{q}_2^T, \boldsymbol{q}_3^T, \boldsymbol{t}^T]^T \in \mathbb{R}^{12\times 1}$$

is the unknown parameter vector that has to be estimated.

The covariance matrix of the noise $\bar{\boldsymbol{n}}$ in (8.14) is denoted by

$$\boldsymbol{R}_{\bar{\boldsymbol{n}}} = (\boldsymbol{I}_N \otimes \boldsymbol{U}_M^T) \boldsymbol{R}_{\boldsymbol{n}} (\boldsymbol{I}_N \otimes \boldsymbol{U}_M) \in \mathbb{R}^{(M-1)N\times(M-1)N},$$

[2] We use the matrix property $\text{vec}(\boldsymbol{ABC}) = (\boldsymbol{C}^T \otimes \boldsymbol{A})\text{vec}(\boldsymbol{B})$.

where $\boldsymbol{R}_{\boldsymbol{n}}$ is the covariance matrix of $\boldsymbol{n} = \mathrm{vec}(\boldsymbol{N})$, and is developed in Appendix 8.A. To whiten[3] the noise, the vectorized model in (8.14) (equivalently the model in (8.12)) is transformed to

$$\begin{aligned} \bar{\boldsymbol{d}}' &= \boldsymbol{R}_{\bar{\boldsymbol{n}}}^{-1/2}\bar{\boldsymbol{d}} \\ &= \boldsymbol{R}_{\bar{\boldsymbol{n}}}^{-1/2}((\boldsymbol{I}_N \otimes \bar{\boldsymbol{A}})\boldsymbol{s} + \bar{\boldsymbol{n}}) \\ &= \boldsymbol{R}_{\bar{\boldsymbol{n}}}^{-1/2}((\boldsymbol{C}_{\mathrm{e}}^T \otimes \bar{\boldsymbol{A}})\boldsymbol{\theta} + \bar{\boldsymbol{n}}). \end{aligned} \tag{8.15}$$

Here, the notation $\boldsymbol{R}_{\boldsymbol{n}}^{1/2}$ is defined from the Cholesky decomposition $\boldsymbol{R}_{\boldsymbol{n}} := \boldsymbol{R}_{\boldsymbol{n}}^{1/2}\boldsymbol{R}_{\boldsymbol{n}}^{T/2}$.

In the next section, we propose several estimators of $\boldsymbol{\theta}$ from the processed squared-range measurements $\bar{\boldsymbol{d}}'$.

8.3 Linear least-squares estimators

To begin with, we first look at the topology-agnostic classical LS-based location estimator (i.e., ignoring the prior sensor placement information).

8.3.1 Classical LS-based localization (topology-agnostic)

We use the classical (weighted) LS estimator of $\boldsymbol{s}$ from $\bar{\boldsymbol{d}}'$ in (8.15) to estimate the absolute position of the sensors as

$$\begin{aligned} \widehat{\boldsymbol{s}}_{\mathrm{LS}} &= \underset{\boldsymbol{s}\in\mathbb{R}^{3N}}{\arg\min} \, \|\bar{\boldsymbol{d}}' - \boldsymbol{R}_{\bar{\boldsymbol{n}}}^{-1/2}(\boldsymbol{I}_N \otimes \bar{\boldsymbol{A}})\boldsymbol{s}\|_2^2 \\ &= (\boldsymbol{R}_{\bar{\boldsymbol{n}}}^{-1/2}(\boldsymbol{I}_N \otimes \bar{\boldsymbol{A}}))^{\dagger}\bar{\boldsymbol{d}}', \end{aligned} \tag{8.16}$$

which is unique if $\boldsymbol{I}_N \otimes \bar{\boldsymbol{A}}$ has full column-rank. This requires $M \geq 4$. Finally, we have

$$\widehat{\boldsymbol{S}}_{\mathrm{LS}} = \mathrm{unvec}(\widehat{\boldsymbol{s}}_{\mathrm{LS}}) \in \mathbb{R}^{3\times N}.$$

In this classical LS-based localization, the knowledge about the known sensor topology is not exploited, and the absolute position of each sensor is estimated separately.

8.3.2 Unconstrained LS estimator

Note that the unknown parameter vector $\boldsymbol{\theta}$ has a structure because $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3]$ is a unitary matrix. We propose to estimate $\boldsymbol{\theta}$, ignoring its structure, from $\bar{\boldsymbol{d}}'$ in (8.15) using the following (weighted) LS estimator

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_{\mathrm{LS}} &= \underset{\boldsymbol{\theta}}{\arg\min} \, \|\bar{\boldsymbol{d}}' - \boldsymbol{R}_{\bar{\boldsymbol{n}}}^{-1/2}(\boldsymbol{C}_{\mathrm{e}}^T \otimes \bar{\boldsymbol{A}})\boldsymbol{\theta}\|_2^2 \\ &= (\boldsymbol{R}_{\bar{\boldsymbol{n}}}^{-1/2}(\boldsymbol{C}_{\mathrm{e}}^T \otimes \bar{\boldsymbol{A}}))^{\dagger}\bar{\boldsymbol{d}}'. \end{aligned} \tag{8.17}$$

[3] The noise covariance is parameter dependent, and hence, for whitening it we use an estimated covariance matrix $\widehat{\boldsymbol{R}}_{\boldsymbol{n}}$ as discussed in Sec. 8.6.

The estimator in (8.17) will have a unique solution if the matrix $\boldsymbol{C}_{\mathrm{e}}^T \otimes \bar{\boldsymbol{A}}$ has full column-rank, i.e., $\boldsymbol{C}_{\mathrm{e}}^T$ and $\bar{\boldsymbol{A}}$ are both full-column rank, and this requires $(M-1)N \geq 12$. Finally, we have

$$\widehat{\boldsymbol{\Theta}}_{\mathrm{LS}} = \mathrm{unvec}(\widehat{\boldsymbol{\theta}}_{\mathrm{LS}}) = \left[\ \widehat{\boldsymbol{Q}}_{\mathrm{LS}} \,\middle|\, \widehat{\boldsymbol{t}}_{\mathrm{LS}}\ \right]. \tag{8.18}$$

8.3.3 Unitarily constrained estimators

The LS estimate $\widehat{\boldsymbol{Q}}_{\mathrm{LS}}$ obtained in (8.18) is typically (in presence of noise) not a rotation matrix. Hence, we next propose two LS estimators with a unitary constraint on $\boldsymbol{Q}$. Both these estimators solve an optimization problem on the Stiefel manifold.

For this purpose, we decouple the rotations and the translations in (8.2) by eliminating the all-one vector $\mathbf{1}_N^T$, and hence the matrix $\boldsymbol{t}\mathbf{1}_N^T$. In order to eliminate $\boldsymbol{t}\mathbf{1}_N^T$, we use an isometry matrix $\boldsymbol{U}_N$, and as earlier, this matrix is obtained by the isometry decomposition of $\boldsymbol{P}_N$, given by

$$\boldsymbol{P}_N = \boldsymbol{I}_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T = \boldsymbol{U}_N\boldsymbol{U}_N^T,$$

where $\boldsymbol{U}_N$ is an $N \times (N-1)$ matrix obtained by collecting orthonormal basis vectors of the null-space of $\mathbf{1}_N$ such that $\mathbf{1}_N^T\boldsymbol{U}_N = \mathbf{0}_{N-1}^T$. Right-multiplying both sides of (8.2) with $\boldsymbol{U}_N$ leads to

$$\boldsymbol{S}\boldsymbol{U}_N = \boldsymbol{Q}\boldsymbol{C}\boldsymbol{U}_N. \tag{8.19}$$

Combining (8.11) and (8.19) we get the following linear model

$$\bar{\boldsymbol{D}}\boldsymbol{U}_N = \bar{\boldsymbol{A}}\boldsymbol{Q}\boldsymbol{C}\boldsymbol{U}_N + \bar{\boldsymbol{N}}\boldsymbol{U}_N,$$

which can be further simplified as

$$\tilde{\boldsymbol{D}} = \bar{\boldsymbol{A}}\boldsymbol{Q}\bar{\boldsymbol{C}} + \tilde{\boldsymbol{N}}, \tag{8.20}$$

where we have introduced the following matrices:

$$\begin{aligned} \tilde{\boldsymbol{D}} &= \boldsymbol{U}_M^T\boldsymbol{D}\boldsymbol{U}_N \in \mathbb{R}^{(M-1)\times(N-1)}, \\ \bar{\boldsymbol{C}} &= \boldsymbol{C}\boldsymbol{U}_N \in \mathbb{R}^{3\times(N-1)}, \\ \text{and} \quad \tilde{\boldsymbol{N}} &= \boldsymbol{U}_M^T\boldsymbol{N}\boldsymbol{U}_N \in \mathbb{R}^{(M-1)\times(N-1)}. \end{aligned}$$

Vectorizing (8.20), we obtain

$$\tilde{\boldsymbol{d}} = \overbrace{(\bar{\boldsymbol{C}}^T \otimes \bar{\boldsymbol{A}})}^{K\times 9}\,\boldsymbol{q} + \tilde{\boldsymbol{n}}, \tag{8.21}$$

where $K = (M-1)(N-1)$, $\tilde{\boldsymbol{d}} = \mathrm{vec}(\tilde{\boldsymbol{D}})$, $\boldsymbol{q} = \mathrm{vec}(\boldsymbol{Q})$, and $\tilde{\boldsymbol{n}} = \mathrm{vec}(\tilde{\boldsymbol{N}})$. The covariance matrix of the noise $\tilde{\boldsymbol{n}}$ in (8.21) is denoted by

$$\boldsymbol{R}_{\tilde{\boldsymbol{n}}} = (\boldsymbol{U}_N^T \otimes \boldsymbol{U}_M^T)\boldsymbol{R}_{\boldsymbol{n}}(\boldsymbol{U}_N \otimes \boldsymbol{U}_M) \in \mathbb{R}^{K\times K}.$$

We will estimate $\boldsymbol{Q}$ based on a (weighted) LS formulation with a *unitary* constraint, as given by

$$\arg\min_{\boldsymbol{Q}} \ \|\boldsymbol{R}_{\tilde{\boldsymbol{n}}}^{-1/2}(\tilde{\boldsymbol{d}} - (\bar{\boldsymbol{C}}^T \otimes \bar{\boldsymbol{A}})\boldsymbol{q})\|_2^2 \tag{8.22a}$$

$$\text{s.to} \quad \boldsymbol{q} = \text{vec}(\boldsymbol{Q}), \ \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_3. \tag{8.22b}$$

The optimization problem in (8.22) is *nonconvex* due to the quadratic equality constraint, and does not generally admit a known closed-form solution. However, such optimization problems can be solved iteratively as will be discussed later on. Before presenting the iterative algorithm, we will first look at a simplified version of (8.22).

Simplified unitarily constrained LS (SCLS) estimator

The optimization problem in (8.22) can be simplified and brought to the standard form of an *orthogonal Procrustes problem* (OPP) with a noniterative known solution. The OPP is generally used to compute rotations between subspaces.

Assuming that $\bar{\boldsymbol{A}}$ has full column-rank (this can be ensured with optimal anchor placement [Chepuri et al., 2013b]), and multiplying both sides of (8.20) with $\bar{\boldsymbol{A}}^\dagger$, we obtain

$$\check{\boldsymbol{D}} = \boldsymbol{Q}\bar{\boldsymbol{C}} + \check{\boldsymbol{N}}, \tag{8.23}$$

where $\check{\boldsymbol{D}} := \bar{\boldsymbol{A}}^\dagger\tilde{\boldsymbol{D}}$ and $\check{\boldsymbol{N}} := \bar{\boldsymbol{A}}^\dagger\tilde{\boldsymbol{N}}$. The simplified unitarily constrained LS (SCLS) problem is then given as

$$\begin{aligned} \widehat{\boldsymbol{Q}}_{\text{SCLS}} = \arg\min_{\boldsymbol{Q}} \quad & \|\check{\boldsymbol{D}} - \boldsymbol{Q}\bar{\boldsymbol{C}}\|_F^2 \\ \text{s.to} \quad & \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_3. \end{aligned} \tag{8.24}$$

The SCLS estimator in (8.24) is *suboptimal* for the problem in (8.22) due to the colored noise $\check{\boldsymbol{N}}$ in (8.23).

Theorem 8.1 (Solution to the SCLS problem)**.** *The constrained LS problem in (8.24) admits a noniterative known solution given by $\widehat{\boldsymbol{Q}}_{\text{SCLS}} = \boldsymbol{V}\boldsymbol{U}^T$, where $\boldsymbol{V}$ and $\boldsymbol{U}$ are obtained from the singular value decomposition (SVD) of $\check{\boldsymbol{D}}\bar{\boldsymbol{C}}^T =: \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T$ in which matrices $\boldsymbol{V} \in \mathbb{R}^{3\times3}$, $\boldsymbol{U} \in \mathbb{R}^{3\times3}$ are unitary, and $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$ is diagonal. The obtained solution is unique if and only if $\check{\boldsymbol{D}}\bar{\boldsymbol{C}}^T$ is nonsingular.*

Proof. See [Golub and Van Loan, 1996, pg. 601]. □

Remark 8.1 (Alternative SCLS formulation)**.** *Instead of pseudo inverting $\bar{\boldsymbol{A}}$ in* (8.20) *to arrive at (8.23), we can alternatively pseudo-invert $\bar{\boldsymbol{C}}$ in* (8.20) *to arrive at another OPP given by*

$$\begin{aligned} \widehat{\boldsymbol{Q}}_{\text{A-SCLS}} = \arg\min_{\boldsymbol{Q}} & \|\check{\boldsymbol{D}} - \bar{\boldsymbol{A}}\boldsymbol{Q}\|_F^2 \\ \text{s.to} \quad & \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_3, \end{aligned} \tag{8.25}$$

where $\check{\boldsymbol{D}} := \tilde{\boldsymbol{D}}\bar{\boldsymbol{C}}^{\dagger}$*. The OPP in* (8.25) *has a closed-form solution* $\widehat{\boldsymbol{Q}}_{\text{A-SCLS}} = \boldsymbol{U}\boldsymbol{V}^T$*, where the unitary matrices* $\boldsymbol{U} \in \mathbb{R}^{3\times 3}$ *and* $\boldsymbol{V} \in \mathbb{R}^{3\times 3}$ *are obtained from the SVD of* $\bar{\boldsymbol{A}}^T\check{\boldsymbol{D}} =: \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$*.*

Pseudo inverting $\bar{\boldsymbol{C}}$ can often assure better conditioning as the topology matrix is usually designed at the factory. However, the alternative SCLS formulation in (8.25) cannot be used in case of perturbations on the sensor positions, which is discussed in Sec. 8.5. Hence, from now on we will not consider the approach in Remark 8.1.

Subsequently, the SCLS estimate of the translation $\boldsymbol{t}$ can be computed using $\widehat{\boldsymbol{Q}}_{\text{SCLS}}$ obtained by solving (8.24). We can write (8.14) equivalently as

$$\bar{\boldsymbol{d}} = \left[\ (\boldsymbol{C}^T \otimes \bar{\boldsymbol{A}}) \ \middle|\ (\boldsymbol{1}_N \otimes \bar{\boldsymbol{A}})\ \right] \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{t} \end{bmatrix} + \bar{\boldsymbol{n}}, \tag{8.26}$$

where $(\boldsymbol{C}^T \otimes \bar{\boldsymbol{A}}) \in \mathbb{R}^{(M-1)N\times 9}$, and $(\boldsymbol{1}_N \otimes \bar{\boldsymbol{A}}) \in \mathbb{R}^{(M-1)N\times 3}$. Substituting $\widehat{\boldsymbol{q}}_{\text{SCLS}} := \text{vec}(\widehat{\boldsymbol{Q}}_{\text{SCLS}})$ in the above model, and moving the knowns to the left side, we get the following linear model

$$\bar{\boldsymbol{d}} - (\boldsymbol{C}^T \otimes \bar{\boldsymbol{A}})\widehat{\boldsymbol{q}}_{\text{SCLS}} = (\boldsymbol{1}_N \otimes \bar{\boldsymbol{A}})\boldsymbol{t} + \bar{\boldsymbol{n}}.$$

The SCLS estimate for the translations is given by the following LS estimate

$$\begin{aligned} \widehat{\boldsymbol{t}}_{\text{SCLS}} &= \arg\min_{\boldsymbol{t}} \quad \|\bar{\boldsymbol{d}} - (\boldsymbol{C}^T \otimes \bar{\boldsymbol{A}})\widehat{\boldsymbol{q}}_{\text{SCLS}} - (\boldsymbol{1}_N \otimes \bar{\boldsymbol{A}})\boldsymbol{t}\|_2^2 \\ &= (\boldsymbol{1}_N \otimes \bar{\boldsymbol{A}})^{\dagger}(\bar{\boldsymbol{d}} - (\boldsymbol{C}^T \otimes \bar{\boldsymbol{A}})\widehat{\boldsymbol{q}}_{\text{SCLS}}). \end{aligned} \tag{8.27}$$

Since we solve the SCLS estimates in an unweighted LS sense, we need not compute the related noise covariance estimates.

Algorithm 8.1 CLS based on Gauss-Newton's method.

```
Compute initial value $\boldsymbol{Q}_0 := \widehat{\boldsymbol{Q}}_{\text{SCLS}}$,
initialize $\boldsymbol{L}$, $\boldsymbol{z}$, iteration counter $k = 0$, $\varepsilon$, and $\varepsilon_0$.
while $\varepsilon_k > \varepsilon$
    compute $\boldsymbol{J}$ using (8.48)
    compute a Gauss-Newton step

        $\boldsymbol{x}_k = (\boldsymbol{J}^T\boldsymbol{J})^{-1}\boldsymbol{J}^T(\boldsymbol{z} - \boldsymbol{L}\text{vec}(\boldsymbol{Q}_k))$

    compute the optimal step-length $\alpha_k$ using (8.50).
    update $\boldsymbol{Q}_{k+1} = \boldsymbol{Q}_k\boldsymbol{\Omega}(\alpha_k\boldsymbol{x}_k)$,
                        where $\boldsymbol{\Omega}(\cdot)$ is defined in (8.46).
    increment $k = k + 1$.
    compute $\varepsilon_{k+1} = \frac{\|\boldsymbol{J}^T(\boldsymbol{z}-\boldsymbol{L}\text{vec}(\boldsymbol{Q}_k))\|_2}{\|\boldsymbol{J}\|_F\|\boldsymbol{z}-\boldsymbol{L}\text{vec}(\boldsymbol{Q}_k)\|_2}$.
end while.
```

Unitarily constrained LS (CLS) estimator

The CLS estimates are obtained by solving the optimization problem that was introduced earlier in (8.22), which we recall as

$$\widehat{\boldsymbol{Q}}_{\mathrm{CLS}} = \arg\min_{\boldsymbol{Q}} \|\boldsymbol{R}_{\tilde{\boldsymbol{n}}}^{-1/2}(\tilde{\boldsymbol{d}} - (\bar{\boldsymbol{C}}^T \otimes \bar{\boldsymbol{A}})\boldsymbol{q})\|_2^2 \tag{8.28a}$$

$$\text{s.to} \quad \boldsymbol{q} = \mathrm{vec}(\boldsymbol{Q}),\ \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_3. \tag{8.28b}$$

The optimization problem in (8.28) is a generalization of the OPP, and is sometimes also referred to as the *weighted orthogonal Procrustes problem* (WOPP) [Viklands, 2008]. Unlike the OPP of (8.24), which has a closed-form analytical solution, the optimization problem (8.28) does not admit a closed-form solution. However, it can be solved using iterative methods based on either Gauss-Newton's method, Newton's method [Viklands, 2008], or steepest descent [Manton, 2002]. In this chapter, we restrict ourselves to Gauss-Newton's method for solving (8.28) because of the availability of a good initial value (e.g., the closed-form solution from Theorem 8.1) for the iterative algorithm.

The optimization problem in (8.28) is a LS problem on the Stiefel manifold. To simplify the notations we write (8.28) in a more general form:

$$\begin{aligned} &\arg\min_{\boldsymbol{Q}} \|\boldsymbol{z} - \boldsymbol{L}\mathrm{vec}(\boldsymbol{Q})\|_2^2 \\ &\text{s.to} \quad \boldsymbol{Q} \in \mathcal{V}_{3,3}, \end{aligned} \tag{8.29}$$

and for solving (8.28) we use

$$\begin{aligned} \boldsymbol{z} &:= \boldsymbol{R}_{\tilde{\boldsymbol{n}}}^{-1/2}\tilde{\boldsymbol{d}} \in \mathbb{R}^{K\times 1}, \\ \text{and} \qquad \boldsymbol{L} &:= \boldsymbol{R}_{\tilde{\boldsymbol{n}}}^{-1/2}(\bar{\boldsymbol{C}}^T \otimes \bar{\boldsymbol{A}}) \in \mathbb{R}^{K\times 9}. \end{aligned} \tag{8.30}$$

The algorithm is derived in Appendix 8.B, and it is summarized as Algorithm 8.1. A more profound treatment on WOPP is found in [Viklands, 2008].

As before, the estimate for the translation $\boldsymbol{t}$ can then be computed using $\widehat{\boldsymbol{q}}_{\mathrm{CLS}} := \mathrm{vec}(\widehat{\boldsymbol{Q}}_{\mathrm{CLS}})$ in (8.26). The CLS estimate for the translation is given by the following (weighted) LS estimate

$$\begin{aligned} \widehat{\boldsymbol{t}}_{\mathrm{CLS}} &= \arg\min_{\boldsymbol{t}} \|\bar{\boldsymbol{d}}' - \boldsymbol{R}_{\bar{\boldsymbol{n}}}^{-1/2}((\boldsymbol{C}^T \otimes \bar{\boldsymbol{A}})\widehat{\boldsymbol{q}}_{\mathrm{SCLS}} + (\boldsymbol{1}_N \otimes \bar{\boldsymbol{A}})\boldsymbol{t})\|_2^2 \\ &= (\boldsymbol{R}_{\bar{\boldsymbol{n}}}^{-1/2}(\boldsymbol{1}_N \otimes \bar{\boldsymbol{A}}))^{\dagger}(\bar{\boldsymbol{d}}' - \boldsymbol{R}_{\bar{\boldsymbol{n}}}^{-1/2}(\boldsymbol{C}^T \otimes \bar{\boldsymbol{A}})\widehat{\boldsymbol{q}}_{\mathrm{CLS}}). \end{aligned}$$

8.3.4 Topology-aware (TA) localization

A complementary by-product of the rigid body localization is the *topology-aware* localization. In this case, the position and orientation estimation is not the main interest, but the absolute position of each sensor node has to be estimated, given that the sensors follow a certain topology. This latter information can be used as a constraint for estimating the sensor positions rather than estimating them separately. For the rigidity constraint, using $\widehat{\boldsymbol{Q}}$ and $\widehat{\boldsymbol{t}}$ obtained from either the SCLS or CLS estimator, we can then compute the absolute positions of each sensor on the rigid body as

$$\widehat{\boldsymbol{S}}_{\text{TA}} = \widehat{\boldsymbol{Q}}\boldsymbol{C} + \widehat{\boldsymbol{t}}\mathbf{1}_N^T. \tag{8.31}$$

8.4 Unitarily constrained Cramér-Rao bound

Suppose we want to estimate the unknown vector $\boldsymbol{\theta} = [\boldsymbol{q}_1^T, \boldsymbol{q}_2^T, \boldsymbol{q}_3^T, \boldsymbol{t}^T]^T \in \mathbb{R}^{12\times 1}$ from the range measurements y_{mn} corrupted by independent noise $v_{mn} \sim \mathcal{N}(0, \sigma_{mn}^2)$ for $n = 1, 2, \ldots, N$, and $m = 1, 2, \ldots, M$, where the observations follow the nonlinear model (8.3). We can compute the CRB for any unbiased estimator of $\boldsymbol{\theta}$ as described next.

8.4.1 Unconstrained CRB

The covariance matrix of any unbiased estimate of the parameter vector $\boldsymbol{\theta}$ satisfies [Kay, 1993]

$$\mathbb{E}\{(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T\} \geq \boldsymbol{C}_{\text{CRB}} = \boldsymbol{F}^{-1},$$

where the Fisher information matrix (FIM) $\boldsymbol{F}$ is given by

$$\boldsymbol{F} = \sum_{m=1}^{M}\sum_{n=1}^{N}\mathbb{E}\left\{\left(\frac{\partial \ln p(y_{mn};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right)\left(\frac{\partial \ln p(y_{mn};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right)^T\right\}.$$

This is the *Cramér-Rao bound theorem* and $\boldsymbol{C}_{\text{CRB}}$ is the Cramér-Rao lower bound (CRB). Let us define $\boldsymbol{c}_{e,n} = [\boldsymbol{c}_n^T, 1]^T \in \mathbb{R}^4$ for $n = 1, 2, \ldots, N$. The computation of the CRB for observations with Gaussian likelihoods is straightforward, and is given as

$$\boldsymbol{F}(\boldsymbol{\theta}) = \sum_{m=1}^{M}\sum_{n=1}^{N}\frac{(\boldsymbol{c}_{e,n}\otimes\boldsymbol{I}_3)(\boldsymbol{a}_m - (\boldsymbol{Q}\boldsymbol{c}_n + \boldsymbol{t}))(\boldsymbol{a}_m - (\boldsymbol{Q}\boldsymbol{c}_n + \boldsymbol{t}))^T(\boldsymbol{c}_{e,n}^T\otimes\boldsymbol{I}_3)}{\sigma_{mn}^2\|\boldsymbol{a}_m - (\boldsymbol{Q}\boldsymbol{c}_n + \boldsymbol{t})\|_2^2} \in \mathbb{R}^{12\times 12}. \tag{8.32}$$

8.4.2 Constrained CRB

The FIM in (8.32), does not take into account the unitary constraint on the matrix $\boldsymbol{Q}$, i.e., $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_3$. Generally, if the parameter vector $\boldsymbol{\theta}$ is subject to some P continuously differentiable (nonredundant) constraints $\boldsymbol{h}(\boldsymbol{\theta}) = \mathbf{0}$, then with these constraints, the resulting constrained CRB is lower than the unconstrained CRB. In [Stoica and Ng, 1998], it is shown that the constrained CRB (CCRB) has the form

$$\mathbb{E}\{(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T\} \geq \boldsymbol{C}_{\text{CCRB}}(\boldsymbol{\theta}) = \boldsymbol{M}(\boldsymbol{M}^T\boldsymbol{F}\boldsymbol{M})^{-1}\boldsymbol{M}^T, \tag{8.33}$$

where $\boldsymbol{F}$ is the FIM for the unconstrained estimation problem as in (8.32), and an isometry matrix $\boldsymbol{M} \in \mathbb{R}^{12\times(12-P)}$ is obtained by collecting orthonormal basis vectors of the null-space of the gradient matrix

$$\boldsymbol{G}(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{h}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \in \mathbb{R}^{P\times 12}.$$

The nonredundant constraints ensure that the matrix $\boldsymbol{G}(\boldsymbol{\theta})$ is full row-rank, and implies

$$\boldsymbol{G}(\boldsymbol{\theta})\boldsymbol{M} = \boldsymbol{0}$$

while $\boldsymbol{M}^T\boldsymbol{M} = \boldsymbol{I}_{12-P}$. For the unitarily constrained CRB (CCRB) denoted by $\boldsymbol{C}_{\text{CCRB}}(\boldsymbol{\theta})$, we have to consider the unitary constraint $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_3$, which can be expressed by the following $P = 6$ nonredundant constraints

$$\begin{aligned}\boldsymbol{h}(\boldsymbol{\theta}) =&[\boldsymbol{q}_1^T\boldsymbol{q}_1 - 1, \boldsymbol{q}_2^T\boldsymbol{q}_1, \boldsymbol{q}_3^T\boldsymbol{q}_1, \boldsymbol{q}_2^T\boldsymbol{q}_2 - 1,\\ &\boldsymbol{q}_3^T\boldsymbol{q}_2, \boldsymbol{q}_3^T\boldsymbol{q}_3 - 1]^T = \boldsymbol{0}_6 \in \mathbb{R}^{6\times 1},\end{aligned} \tag{8.34}$$

where the symmetric redundant constraints are discarded. The gradient matrix for the constraints in (8.34) can be computed as follows

$$\begin{aligned}\boldsymbol{G}(\boldsymbol{\theta}) &= \frac{\partial \boldsymbol{h}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\\ &= \begin{bmatrix} 2\boldsymbol{q}_1^T & \boldsymbol{0}_3^T & \boldsymbol{0}_3^T & \boldsymbol{0}_3^T \\ \boldsymbol{q}_2^T & \boldsymbol{q}_1^T & \boldsymbol{0}_3^T & \boldsymbol{0}_3^T \\ \boldsymbol{q}_3^T & \boldsymbol{0}_3^T & \boldsymbol{q}_1^T & \boldsymbol{0}_3^T \\ \boldsymbol{0}_3^T & 2\boldsymbol{q}_2^T & \boldsymbol{0}_3^T & \boldsymbol{0}_3^T \\ \boldsymbol{0}_3^T & \boldsymbol{q}_3^T & \boldsymbol{q}_2^T & \boldsymbol{0}_3^T \\ \boldsymbol{0}_3^T & \boldsymbol{0}_3^T & 2\boldsymbol{q}_3^T & \boldsymbol{0}_3^T \end{bmatrix} \in \mathbb{R}^{6\times 12}.\end{aligned}$$

An orthonormal basis of the null-space of the gradient matrix is finally given by

$$\boldsymbol{M} = \frac{1}{\sqrt{2}}\left[\begin{array}{ccc|c} -\boldsymbol{q}_3 & \boldsymbol{0}_3 & \boldsymbol{q}_2 & \\ \boldsymbol{0}_3 & -\boldsymbol{q}_3 & -\boldsymbol{q}_1 & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{q}_1 & \boldsymbol{q}_2 & \boldsymbol{0}_3 & \\ \hline \multicolumn{3}{c|}{\boldsymbol{0}_{3\times 3}} & \sqrt{2}\boldsymbol{I}_3 \end{array}\right].$$

The CCRB for TA-localization can be easily derived from $\boldsymbol{C}_{\text{CCRB}}(\boldsymbol{\theta})$ using the transformation of parameters [Kay, 1993].

Theorem 8.2 (Biased estimator)**.** *An unbiased constrained estimator for* **Q** *does not exist, except for the noiseless case.*

Proof. See Appendix 8.C. □

Due to Theorem 8.2, the mean-squared-error (MSE) of any unitarily constrained estimator will be lower than the CCRB for high noise levels. However, at low noise levels, the bias tends to zero, and the CCRB gives a good lower bound on the MSE of the unitarily constrained estimators.

8.5 Unitarily constrained total-least-squares

In the previous section, we assumed that the position of the sensors on the rigid body in the reference frame, i.e., the matrix $\boldsymbol{C}$, is accurately known. In practice, there is no reason to believe that errors are restricted only to the range measurements and there are no perturbations on the initial sensor positions. Such perturbations can be introduced for instance during fabrication or if the body is not entirely rigid (e.g., wing flexing of an aircraft).

Let us assume that the position of the nth sensor in the reference frame $\boldsymbol{c}_n$ is noisy, and let us denote the perturbation on $\boldsymbol{c}_n$ by $\boldsymbol{e}_n$, and the perturbation on $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_N]$ by the matrix $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N] \in \mathbb{R}^{3\times N}$. The covariance matrix of the perturbation $\boldsymbol{e} = \mathrm{vec}(\boldsymbol{E})$ is denoted by $\boldsymbol{R}_{\boldsymbol{e}} = \sigma_{\mathrm{e}}^2 \boldsymbol{I}_{3N}$. In other words, we assume that the perturbations $\boldsymbol{e}_n, n = 1, 2, \ldots, N$, are a sequence of independent and identically distributed (i.i.d.) random vectors. To account for such errors in the model, we next propose total-least-squares (TLS) estimators again with unitary constraints.

8.5.1 Simplified unitarily constrained TLS (SCTLS) estimator

Taking the perturbations on the known topology into account, the data model in (8.20) will be modified as

$$\tilde{\boldsymbol{D}} = \bar{\boldsymbol{A}}\boldsymbol{Q}(\bar{\boldsymbol{C}} + \bar{\boldsymbol{E}}) + \tilde{\boldsymbol{N}},$$

where $\bar{\boldsymbol{E}} = \boldsymbol{E}\boldsymbol{U}_N$. Multiplying both sides of the above equation with $\bar{\boldsymbol{A}}^{\dagger}$, we get

$$\check{\boldsymbol{D}} = \boldsymbol{Q}(\bar{\boldsymbol{C}} + \bar{\boldsymbol{E}}) + \check{\boldsymbol{N}}. \tag{8.35}$$

The solution to the data model in (8.35) admits a classical TLS formulation, but with a unitary constraint. The SCTLS optimization problem is given by

$$\begin{aligned} \widehat{\boldsymbol{Q}}_{\text{SCTLS}} = \arg\min_{\boldsymbol{Q}} \quad & \|\bar{\boldsymbol{E}}\|_F^2 + \|\check{\boldsymbol{N}}\|_F^2 \\ \text{s.to} \quad & \check{\boldsymbol{D}} = \boldsymbol{Q}(\bar{\boldsymbol{C}} + \bar{\boldsymbol{E}}) + \check{\boldsymbol{N}}, \text{ and } \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_3. \end{aligned} \tag{8.36}$$

Similar to SCLS, the optimization problem in (8.36) admits a known closed-form solution which makes it simplified compared to the weighted problem discussed in the next section. Also, the noise $\check{\boldsymbol{N}}$ in (8.35) is not white leading to a suboptimal solution.

Theorem 8.3 (Solution to SCTLS)**.** *The SCTLS problem in (8.36) has the same solution as the simplified unitarily constrained LS problem, i.e.,* $\widehat{\boldsymbol{Q}}_{\text{SCTLS}} = \widehat{\boldsymbol{Q}}_{\text{SCLS}}$.

Proof. See Appendix 8.D. □

We next estimate the translation vector $\boldsymbol{t}$. Taking the perturbations into account, we can modify the model in (8.13) as

$$\bar{\boldsymbol{D}} = \bar{\boldsymbol{A}}\boldsymbol{Q}\boldsymbol{C} + \bar{\boldsymbol{A}}\boldsymbol{t}\boldsymbol{1}_N^T + \bar{\boldsymbol{A}}\boldsymbol{Q}\boldsymbol{E} + \bar{\boldsymbol{N}},$$

which can be vectorized and further simplified to

$$\bar{\boldsymbol{d}} - (\boldsymbol{C}^T \otimes \bar{\boldsymbol{A}})\boldsymbol{q} = (\boldsymbol{1}_N \otimes \bar{\boldsymbol{A}})\boldsymbol{t} + \overbrace{(\boldsymbol{I} \otimes \bar{\boldsymbol{A}}\boldsymbol{Q})\boldsymbol{e} + \bar{\boldsymbol{n}}}^{\boldsymbol{v} \in \mathbb{R}^{(M-1)N}}. \tag{8.37}$$

Using $\widehat{\boldsymbol{q}}_{\text{SCLS}}$ in the above model, the SCTLS estimator for the translation taking perturbations into account, is then given by the following (unweighted) LS problem

$$\begin{aligned} \widehat{\boldsymbol{t}}_{\text{SCLS}} &= \arg\min_{\boldsymbol{t}} \|\bar{\boldsymbol{d}} - (\boldsymbol{C}^T \otimes \bar{\boldsymbol{A}})\widehat{\boldsymbol{q}}_{\text{SCLS}} - (\boldsymbol{1}_N \otimes \bar{\boldsymbol{A}})\boldsymbol{t}\|_2^2 \\ &= (\boldsymbol{1}_N \otimes \bar{\boldsymbol{A}})^{\dagger}(\bar{\boldsymbol{d}} - (\boldsymbol{C}^T \otimes \bar{\boldsymbol{A}})\widehat{\boldsymbol{q}}_{\text{SCLS}}). \end{aligned}$$

The algorithms to compute the SCTLS estimates are the same as that of the SCLS estimator. As before, we solve SCTLS in an unweighted LS sense, therefore, we need not estimate the related noise covariance matrix for whitening.

8.5.2 Unitarily constrained TLS (CTLS) estimator

Similar to the CLS formulation, the TLS estimator can be derived without pseudo inverting the matrix $\bar{\boldsymbol{A}}$. The data model in (8.20) taking into account the perturbations on the known sensor topology is then given by

$$\tilde{\boldsymbol{D}} = \bar{\boldsymbol{A}}\boldsymbol{Q}(\bar{\boldsymbol{C}} + \bar{\boldsymbol{E}}) + \tilde{\boldsymbol{N}}, \tag{8.38}$$

which can be further vectorized as

$$\tilde{\boldsymbol{d}} = (\boldsymbol{I}_{3(N-1)} \otimes \bar{\boldsymbol{A}}\boldsymbol{Q})\bar{\boldsymbol{c}} + (\boldsymbol{I}_{3(N-1)} \otimes \bar{\boldsymbol{A}}\boldsymbol{Q})\bar{\boldsymbol{e}} + \tilde{\boldsymbol{n}}, \tag{8.39}$$

where $\bar{\boldsymbol{c}} = \text{vec}(\bar{\boldsymbol{C}}) \in \mathbb{R}^{3(N-1)}$, and $\bar{\boldsymbol{e}} = \text{vec}(\bar{\boldsymbol{E}}) \in \mathbb{R}^{3(N-1)}$.

Assuming that the pre-whitening matrix takes the block diagonal form

$$\boldsymbol{R}_{\boldsymbol{k}}^{-1/2} := \text{diag}(\sigma_{\text{e}}^{-1}\boldsymbol{I}_{3(N-1)}, \boldsymbol{R}_{\tilde{\boldsymbol{n}}}^{-1/2})$$

with $\boldsymbol{k} = [\bar{\boldsymbol{e}}^T, \tilde{\boldsymbol{n}}^T]^T$, the CTLS optimization problem is given by

$$\begin{aligned} \arg\min_{\boldsymbol{Q}} \quad & \|\sigma_{\text{e}}^{-1}\bar{\boldsymbol{e}}\|_2^2 + \|\boldsymbol{R}_{\tilde{\boldsymbol{n}}}^{-1/2}\tilde{\boldsymbol{n}}\|_2^2 \\ \text{s.to} \quad & \tilde{\boldsymbol{D}} = \bar{\boldsymbol{A}}\boldsymbol{Q}(\bar{\boldsymbol{C}} + \bar{\boldsymbol{E}}) + \tilde{\boldsymbol{N}}, \\ & \bar{\boldsymbol{e}} = \text{vec}(\bar{\boldsymbol{E}}), \quad \tilde{\boldsymbol{n}} = \text{vec}(\tilde{\boldsymbol{N}}), \\ & \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_3. \end{aligned} \tag{8.40}$$

Theorem 8.4 (Solution to CTLS)**.** *Assuming that the covariance matrix of the perturbation vector is a scaled identity matrix, the unitarily constrained TLS problem in* (8.40) *has the same solution as a specifically weighted CLS, i.e., it is the solution to*

$$\begin{aligned} \widehat{\boldsymbol{Q}}_{\text{CTLS}} = \; & \underset{\boldsymbol{Q}}{\arg\min} \; \|\boldsymbol{\Lambda}^{1/2}(\tilde{\boldsymbol{d}} - (\bar{\boldsymbol{C}}^T \otimes \bar{\boldsymbol{A}})\boldsymbol{q})\|_2^2 \\ & \text{s.to} \quad \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_3, \end{aligned} \tag{8.41}$$

where matrix $(\bar{\boldsymbol{C}}^T \otimes \bar{\boldsymbol{A}}) \in \mathbb{R}^{K\times 9}$ *was defined earlier, and* $\boldsymbol{\Lambda} \in \mathbb{R}^{K\times K}$ *is the specific weighting matrix defined in* (8.57).

Proof. See Appendix 8.E. □

The optimization problem (8.41) does not have a closed-form solution, and has to be solved iteratively using for instance Gauss-Newton's method (summarized in Algorithm 8.1) with

$$\begin{aligned} & \boldsymbol{z} := \boldsymbol{\Lambda}^{1/2}\tilde{\boldsymbol{d}} \in \mathbb{R}^{K\times 1}, \\ \text{and} \qquad & \boldsymbol{L} := \boldsymbol{\Lambda}^{1/2}(\bar{\boldsymbol{C}}^T \otimes \bar{\boldsymbol{A}}) \in \mathbb{R}^{K\times 9}. \end{aligned} \tag{8.42}$$

The CTLS estimates for the translations can be computed similar to SCTLS as earlier, however, for CTLS we first pre-whiten the noise. The covariance of the noise $\boldsymbol{v}$ in (8.37) is denoted by an $(M-1)N \times (M-1)N$ matrix $\boldsymbol{R}_{\boldsymbol{v}} := \sigma_{\text{e}}^2(\boldsymbol{I} \otimes \bar{\boldsymbol{A}}\bar{\boldsymbol{A}}^T) + \boldsymbol{R}_{\bar{\boldsymbol{n}}}$. We can then use a weighted LS estimator to estimate the translations accounting for the perturbations. The CTLS translation estimates are given by

$$\begin{aligned} \widehat{\boldsymbol{t}}_{\text{CTLS}} &= \underset{\boldsymbol{t}}{\arg\min} \; \|\boldsymbol{R}_{\boldsymbol{v}}^{-1/2}(\bar{\boldsymbol{d}} - (\boldsymbol{C}^T \otimes \bar{\boldsymbol{A}})\widehat{\boldsymbol{q}}_{\text{CTLS}} - (\boldsymbol{1}_N \otimes \bar{\boldsymbol{A}})\boldsymbol{t})\|_2^2 \\ &= (\boldsymbol{R}_{\boldsymbol{v}}^{-1/2}(\boldsymbol{1}_N \otimes \bar{\boldsymbol{A}}))^{\dagger}\boldsymbol{R}_{\boldsymbol{v}}^{-1/2}(\bar{\boldsymbol{d}} - (\boldsymbol{C}^T \otimes \bar{\boldsymbol{A}})\widehat{\boldsymbol{q}}_{\text{CTLS}}). \end{aligned}$$

8.6 Simulation results

We consider $N = 5$ sensors mounted on the vertices of a rigid body (rectangle based pyramid as in Fig. 8.1) with

$$\boldsymbol{C} = \begin{bmatrix} 0.5 & 1.5 & 1.5 & 0.5 & 1 \\ 0 & 0 & 1.5 & 1.5 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{m},$$

and $M = 4$ anchors fixed at location

$$\boldsymbol{A} = \begin{bmatrix} 0 & 100 & 0 & 100 \\ 100 & 100 & 0 & 0 \\ 0 & 100 & 100 & 0 \end{bmatrix} \text{m}.$$

Let us define a function $\mathcal{R}(\cdot) : \mathbb{R}^3 \to \mathcal{V}_{3,3}$ that maps angles in degrees along each dimension into a rotation matrix, and its inverse $\mathcal{R}^{-1}(\cdot) : \mathcal{V}_{3,3} \to \mathbb{R}^3$ which maps a rotation matrix into corresponding angles in degrees (see [Diebel, 2006] for details on converting angles to a rotation matrix and vice versa). Collecting the angles in a 3×1 vector $\boldsymbol{\phi} = [20, -25, 10]^T$ deg, the rotation matrix is then generated according to $\boldsymbol{Q} = \mathcal{R}(\boldsymbol{\phi})$. We use a translation vector $\boldsymbol{t} = [15, 5, 10]^T$ m. The simulations are averaged over $N_{\text{exp}} = 1000$ independent Monte-Carlo experiments.

The performance of the proposed estimators is analyzed in terms of the root-mean-squared-error (RMSE) of the estimates of $\boldsymbol{Q}$, $\boldsymbol{\phi}$, and $\boldsymbol{t}$, and are respectively given as

$$\begin{aligned}
\text{RMSE}(\boldsymbol{Q}) &= \left(\frac{1}{N_{\text{exp}}} \sum_{n=1}^{N_{\text{exp}}} \| \boldsymbol{Q} - \widehat{\boldsymbol{Q}}^{(n)} \|_F^2 \right)^{1/2}, \\
\text{RMSE}(\boldsymbol{\phi}) &= \left(\frac{1}{N_{\text{exp}}} \sum_{n=1}^{N_{\text{exp}}} \| \boldsymbol{\phi} - \mathcal{R}^{-1}(\widehat{\boldsymbol{Q}}^{(n)}) \|_2^2 \right)^{1/2} \text{deg}, \\
\text{RMSE}(\boldsymbol{t}) &= \left(\frac{1}{N_{\text{exp}}} \sum_{n=1}^{N_{\text{exp}}} \| \boldsymbol{t} - \widehat{\boldsymbol{t}}^{(n)} \|_2^2 \right)^{1/2} \text{m},
\end{aligned}$$

where $\widehat{\boldsymbol{Q}}^{(n)}$ and $\widehat{\boldsymbol{t}}^{(n)}$ denote the estimates during the nth Monte-Carlo experiment. Since the rotation matrix estimated using the unconstrained LS estimator is not a valid rotation matrix, we first project it onto $\mathcal{V}_{3,3}$ using (8.51) before converting it into corresponding angles.

We use the same noise variance for all the range measurements, i.e., $\sigma_{mn} = \sigma$ m for $m = 1, 2, \dots, M$, and $n = 1, 2, \dots, N$. The covariance matrix of the noise $\boldsymbol{n}$, i.e., (see Appendix 8.A)

$$\boldsymbol{R_n} = \text{diag}(2\boldsymbol{\rho})\boldsymbol{R_v}\text{diag}(2\boldsymbol{\rho}) + \boldsymbol{R}_{\boldsymbol{v}^{\odot 2}} - \boldsymbol{\mu}\boldsymbol{\mu}^T,$$

depends on the unknown parameter

$$\boldsymbol{\rho} = [\rho_{11}, \rho_{12}, \dots, \rho_{M1}, \dots, \rho_{MN}]^T \in \mathbb{R}^{MN}.$$

Hence, to whiten it, we use the noisy range measurements

$$\boldsymbol{y} = [y_{11}, y_{12}, \dots, y_{M1}, \dots, y_{MN}]^T \in \mathbb{R}^{MN}$$

in (8.3) instead of $\boldsymbol{\rho}$ to compute the estimated covariance matrix

$$\widehat{\boldsymbol{R}}_{\boldsymbol{n}} = \text{diag}(2\boldsymbol{y})\boldsymbol{R_v}\text{diag}(2\boldsymbol{y}) + \boldsymbol{R}_{\boldsymbol{v}^{\odot 2}} - \boldsymbol{\mu}\boldsymbol{\mu}^T.$$

We use this estimated covariance matrix for pre-whitening the noise. Simulations are provided for different values of the nominal ranging noise σ m.

In Fig. 8.2, the RMSE of the estimates $\boldsymbol{Q}$, $\boldsymbol{\phi}$, and $\boldsymbol{t}$ for different values of σ is shown. The estimators in Fig. 8.2 are LS based where the topology of the sensors is assumed to be accurately known. Due to Theorem 8.2, the RMSE of $\boldsymbol{Q}$ for the constrained estimators is lower than the CCRB at high noise levels. The saturation of the RMSE in Fig. 8.2(a) for $\sigma > 1$ m follows from the following lemma.

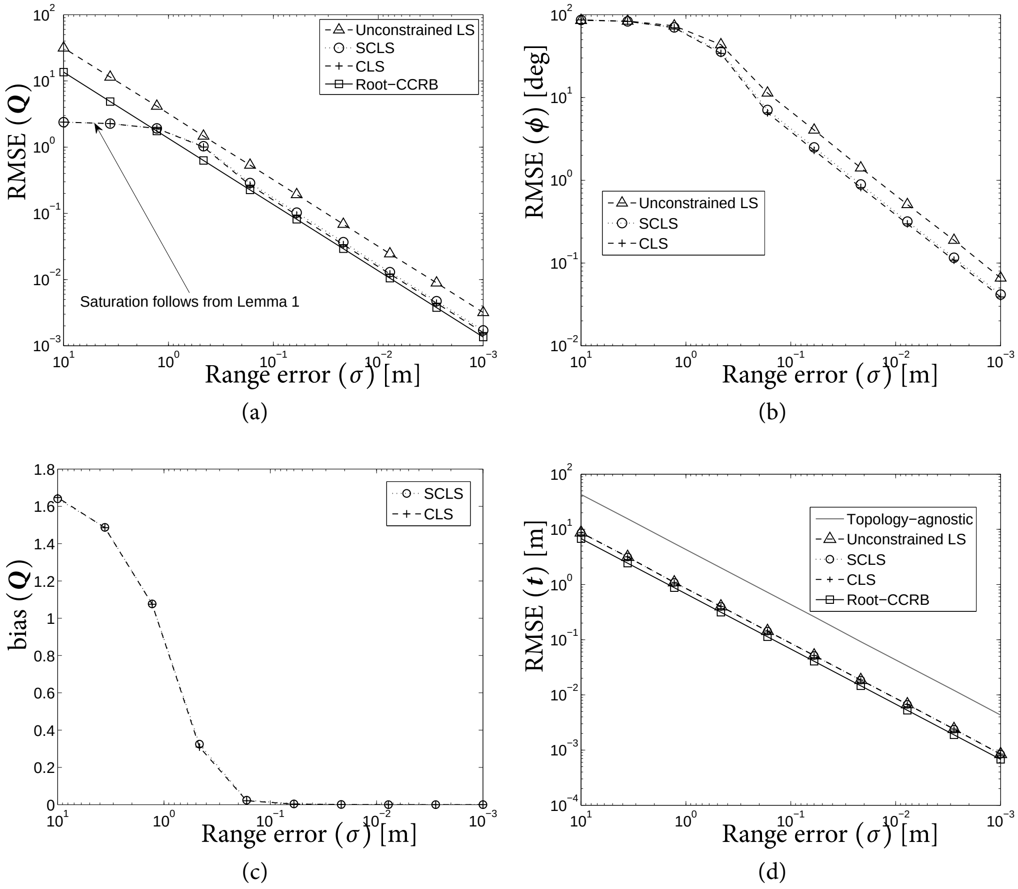
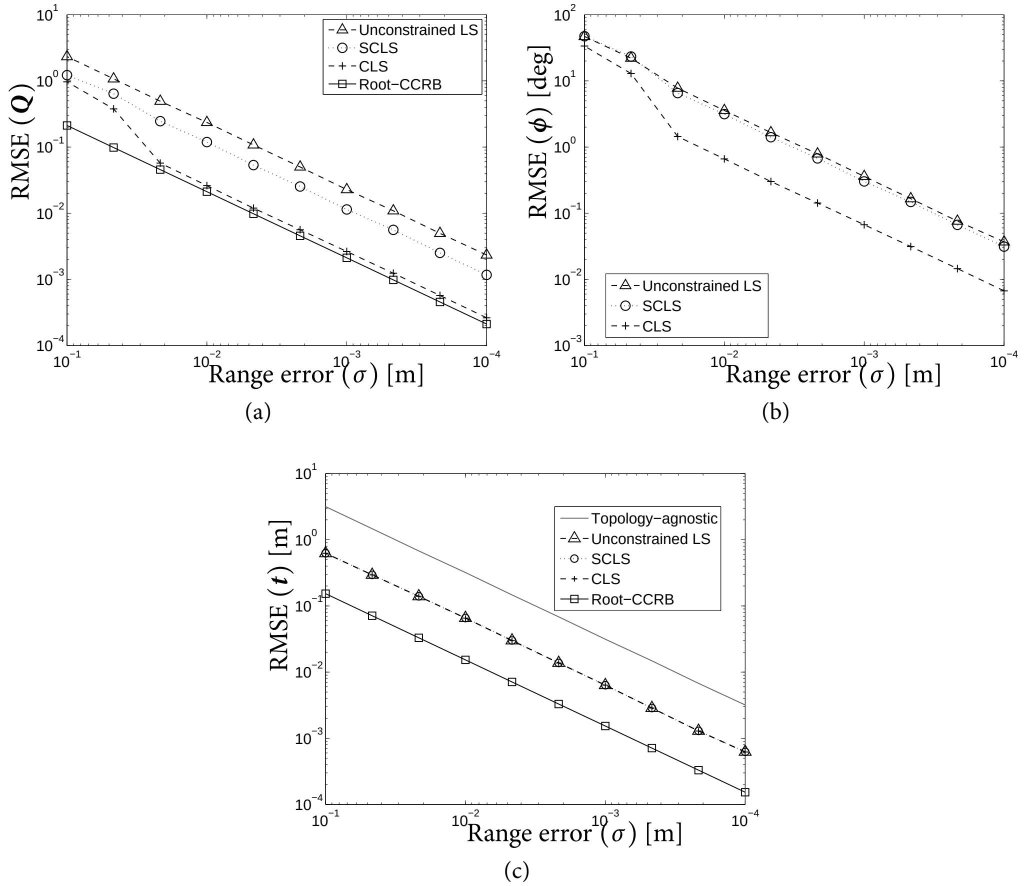


Figure 8.2: (a) RMSE of the estimated rotation matrix $\mathbf{Q}$. The RMSE of the constrained estimators is upper bounded as discussed in Remark 1. (b) RMSE in degrees of the estimated rotations. (c) Bias in the SCLS and CLS estimators for $\mathbf{Q}$. Bias tends to zero for low noise variance. (d) RMSE of the estimated translation vector $\boldsymbol{t}$ along with the solution from the classical LS-based localization.

Theorem 1 (Frobenius norm induced distance)**.** *For any matrix $\mathbf{Q}_i$ and $\mathbf{Q}_j$, such that, $\mathbf{Q}_i \in \mathcal{V}_{n,n}$ and $\mathbf{Q}_j \in \mathcal{V}_{n,n}$, the Frobenius norm induced distance is always upper bounded by $\sqrt{2n}$, i.e., $\|\mathbf{Q}_i - \mathbf{Q}_j\|_F \leq \sqrt{\|\mathbf{Q}_i\|_F + \|\mathbf{Q}_j\|_F} = \sqrt{2n}$.*

However, the CCRB computed using (8.33) does not saturate in this range as there exists no unbiased estimator for high noise values as discussed in Theorem 8.2. Fig. 8.2(b) shows the RMSE in degrees, which gives an insight into how the error on the range measurements translates to the error on the estimated rotations. For the considered scenario, the range error that yields a positioning accuracy of the order of 10 cm also yields an orientation error accuracy of the order of 0.1 deg.

The bias of both the SCLS and CLS estimators is shown in Fig. 8.2(c), and it can be seen that the



(a)

(b)

(c)

Figure 8.3: Effect of bad anchor geometry (cond$(\boldsymbol{A})$ = 100): (a) RMSE of the estimated rotation matrix $\boldsymbol{Q}$. (b) RMSE in degrees of the estimated rotations. (c) RMSE of the estimated translation vector $\boldsymbol{t}$.

bias tends to zero for $\sigma < 0.1$ m (as discussed in Theorem 8.2), whereas the unconstrained LS is an unbiased estimator. The bias is computed as

$$\text{bias}(\boldsymbol{Q}) = \|\frac{1}{N_{\text{exp}}}\sum_{n=1}^{N_{\text{exp}}} \text{vec}(\widehat{\boldsymbol{Q}}^{(n)}) - \text{vec}(\boldsymbol{Q})\|_2.$$

We can see a significant (close to an order of magnitude) improvement in the performance of the location estimates when the knowledge of the topology is exploited as compared to the topology-agnostic case (see Fig. 8.2(d)). The performance of the SCLS estimator is similar (slightly worse) to that of the iterative CLS estimator. The resulting gap between the RMSE and the root-CRB is reasonable, and thus the proposed estimators are robust to the linearization via squaring.

Fig. 8.3 illustrates the effect of a bad anchor geometry, where we use an ill-conditioned matrix $\boldsymbol{A}$ with a condition number of 100. The performance of the SCLS estimators (based on pseudo inverting

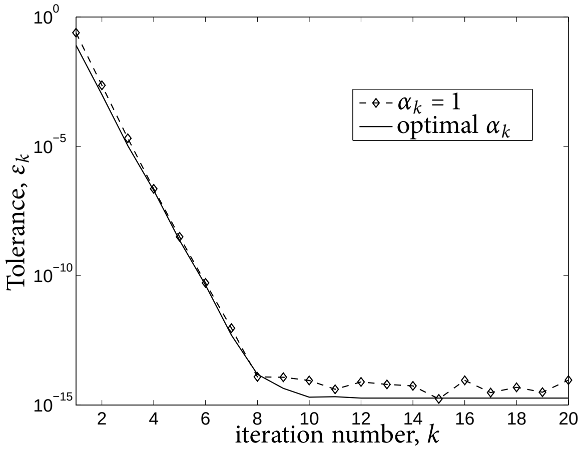


Figure 8.4: Convergence of Gauss-Newton iterations for range error $\sigma = 10^{-2}$ m.

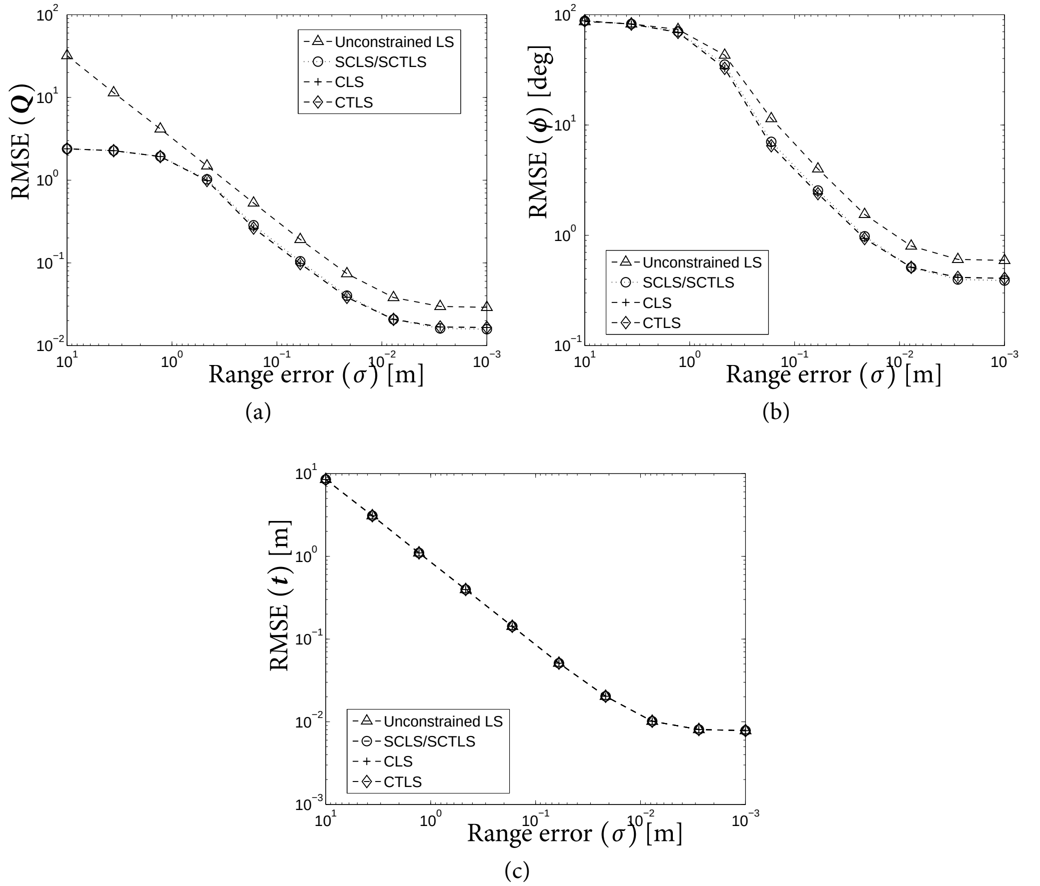
$\bar{\boldsymbol{A}}$) deteriorates for scenarios with a bad anchor geometry, however, the performance of the CLS estimators is not affected. Nevertheless, if the topology is not subject to perturbations, the solution proposed in Remark 8.1 can be used for scenarios with a bad anchor geometry, in which a well-conditioned $\boldsymbol{C}$ can always be designed.

The convergence (i.e., $\varepsilon_k := \frac{\|\boldsymbol{J}^T(\boldsymbol{z} - \boldsymbol{L}\text{vec}(\boldsymbol{Q}_k))\|_2}{\|\boldsymbol{J}\|_F \|\boldsymbol{z} - \boldsymbol{L}\text{vec}(\boldsymbol{Q}_k)\|_2}$, where $\boldsymbol{J}$ defined in (8.48)) of Gauss-Newton's method with $\sigma = 10^{-2}$ m for the optimal step-size and a fixed step-size is shown in Fig. 8.4. We can see that it is sufficient to use a fixed step-size $\alpha_k = 1$ at low noise levels. As observed from the simulations, the iterative algorithm requires typically ten or fewer iterations.

In order to analyze the performance of the estimators for the case when the sensor topology is perturbed, we corrupt the sensor coordinates in the reference frame with a zero mean i.i.d. Gaussian random process of standard deviation $\sigma_\text{e} = 1$ cm. The RMSE of the estimates of $\boldsymbol{Q}$, $\boldsymbol{\phi}$, and $\boldsymbol{t}$ using the unconstrained LS, SCLS/SCTLS, CLS and CTLS estimators is shown in Fig. 8.5. The performance of these estimators is similar to that of the LS-based estimators, except for the error floor, and this is due to the model error (perturbations on the sensor topology). The estimators hit the error floor for lower values of σ as σ_e^2 increases. With an ill-conditioned matrix $\boldsymbol{A}$, the performance of the SCLS/SCTLS (and the unconstrained LS) estimator is worse than the CLS and CTLS estimators.

8.7 Discussion

A framework for joint position and orientation estimation of a rigid body based on range measurements is proposed. We refer to this problem as rigid body localization. Sensor nodes can be mounted on the rigid bodies (e.g., satellites, robots) during fabrication at the factory, and the geometry of how these sensors are mounted is known *a priori* up to a certain accuracy. However, the absolute position of the rigid body is not known. The original nonlinear problem is linearized via squaring of the



(a)

(b)

(c)

Figure 8.5: Results for a perturbed topology $\boldsymbol{C}$. We use $\sigma_e = 1$ cm. (a) RMSE of the estimated rotation matrix $\boldsymbol{Q}$. (b) RMSE in degrees of the estimated rotations. (c) RMSE of the estimated translation vector $\boldsymbol{t}$.

range measurements. The squared-range measurements between the anchors and the sensors on the rigid body can then be used to estimate the position and the orientation of the body. The position and orientation of the rigid body is determined by a translation vector and a rotation matrix, respectively. Ignoring the fact that the rotation matrix is unitary, an unconstrained estimator is proposed. A number of unitarily constrained LS estimators is also proposed, all of which solve an optimization problem on the Stiefel manifold. All the proposed estimators are robust to linearization via squaring. For good anchor geometries, the performance of the SCLS estimator with a closed-form solution is already reasonable. The gap between the RMSE and root-CCRB of the SCLS and CLS estimates is negligible, however, the estimators do not achieve the CCRB. In addition to this, constrained TLS estimators that take into account the inaccuracies in the known sensor topology are also proposed.

8.A Derivation of the covariance matrix $\boldsymbol{R_n}$

In this section, we develop the covariance matrix $\boldsymbol{R_n}$ used for pre-whitening. Modeling the noise on the range measurements v_{mn} as a zero-mean additive white Gaussian process having a variance σ_{mn}^2, we can compute the statistics (up to the second-order) of the zero-mean noise in (8.6), i.e.,

$$n_{mn} = 2\rho_{mn}v_{mn} + v_{mn}^2 - \sigma_{mn}^2 \tag{8.43}$$

as

$$\begin{aligned} &\mathbb{E}\{n_{mn}\} = 0,\\ &\mathbb{E}\{(n_{mn} - \mathbb{E}\{n_{mn}\})^2\} = 4\rho_{mn}^2\sigma_{mn}^2 + 2\sigma_{mn}^4,\\ &\mathbb{E}\{(n_{mn} - \mathbb{E}\{n_{mn}\})(n_{ln} - \mathbb{E}\{n_{ln}\})\} = 0, \quad m \neq l,\\ \text{and}\quad &\mathbb{E}\{(n_{mn} - \mathbb{E}\{n_{mn}\})(n_{ml} - \mathbb{E}\{n_{ml}\})\} = 0, \quad n \neq l, \end{aligned}$$

where we use the fact that, if $v_{mn} \sim \mathcal{N}(0, \sigma_{mn}^2)$ then $\mathbb{E}\{v_{mn}^4\} = 3\sigma_{mn}^4$ and $\mathbb{E}\{v_{mn}^3\} = 0$. Collecting $\rho_{mn}, m = 1, 2, \ldots, M; n = 1, 2, \ldots, N$, in the vector

$$\boldsymbol{\rho} = [\rho_{11}, \rho_{12}, \ldots, \rho_{M1}, \ldots, \rho_{MN}]^T \in \mathbb{R}^{MN},$$

$v_{mn}, m = 1, 2, \ldots, M; n = 1, 2, \ldots, N$, in the vector

$$\boldsymbol{v} = [v_{11}, v_{12}, \ldots, v_{M1}, \ldots, v_{MN}]^T \in \mathbb{R}^{MN},$$

and $\sigma_{mn}^2, m = 1, 2, \ldots, M; n = 1, 2, \ldots, N$, in the vector

$$\boldsymbol{\mu} = [\sigma_{11}^2, \sigma_{12}^2, \ldots, \sigma_{M1}^2, \ldots, \sigma_{MN}^2]^T \in \mathbb{R}^{MN},$$

we can write (8.43) compactly as

$$\boldsymbol{n} = \operatorname{diag}(2\boldsymbol{\rho})\boldsymbol{v} + \boldsymbol{v}^{\odot 2} - \boldsymbol{\mu},$$

where

$$\boldsymbol{n} = [n_{11}, n_{12}, \ldots, n_{M1}, \ldots, n_{MN}]^T \in \mathbb{R}^{MN}.$$

We can then compute the mean of $\boldsymbol{n}$ as $\mathbb{E}\{\boldsymbol{n}\} = \boldsymbol{0}_{MN}$, and the covariance matrix of $\boldsymbol{n}$ as

$$\begin{aligned} \boldsymbol{R_n} &= \mathbb{E}\{(\boldsymbol{n} - \mathbb{E}\{\boldsymbol{n}\})(\boldsymbol{n} - \mathbb{E}\{\boldsymbol{n}\})^T\}\\ &= \operatorname{diag}(2\boldsymbol{\rho})\boldsymbol{R_v}\operatorname{diag}(2\boldsymbol{\rho}) + \boldsymbol{R}_{\boldsymbol{v}^{\odot 2}} - \boldsymbol{\mu}\boldsymbol{\mu}^T \in \mathbb{R}^{MN \times MN}, \end{aligned}$$

where

$$\boldsymbol{R_v} = \operatorname{diag}(\boldsymbol{\mu}) \in \mathbb{R}^{MN \times MN},$$

and

$$\boldsymbol{R}_{\boldsymbol{v}^{\odot 2}} = \begin{bmatrix} 3\sigma_{11}^4 & \sigma_{11}^2\sigma_{12}^2 & \cdots & \sigma_{11}^2\sigma_{MN}^2 \\ \sigma_{11}^2\sigma_{12}^2 & 3\sigma_{12}^4 & \cdots & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ \sigma_{11}^2\sigma_{MN}^2 & \cdots & \cdots & 3\sigma_{MN}^4 \end{bmatrix} \in \mathbb{R}^{MN \times MN}.$$

8.B Gauss-Newton iterations on the Stiefel manifold

The Newton method to solve the WOPP is developed in [Viklands, 2008], and it is adapted to suit our problem. In this section, we derive the Gauss-Newton iterations to solve the LS problems on the Stiefel manifold.

The CLS and CTLS problems solve an optimization problem on the Stiefel manifold of the form

$$\arg\min_{\boldsymbol{Q}} \quad \|\boldsymbol{z} - \boldsymbol{L}\mathrm{vec}(\boldsymbol{Q})\|_2^2 \quad \text{s.to} \quad \boldsymbol{Q} \in \mathcal{V}_{3,3}. \tag{8.44}$$

We can represent any unitary matrix $\boldsymbol{Q}$ in the vicinity of a given unitary matrix $\boldsymbol{Q}_k$ as

$$\boldsymbol{Q} = \boldsymbol{Q}_k \, \boldsymbol{\Omega}(\boldsymbol{x}), \tag{8.45}$$

where the operator $\boldsymbol{\Omega}(\cdot) : \mathbb{R}^3 \to \mathcal{V}_{3,3}$ is defined as

$$\boldsymbol{\Omega}(\boldsymbol{x}) = \exp(\boldsymbol{X}(\boldsymbol{x})), \tag{8.46}$$

and $\boldsymbol{X}(\boldsymbol{x})$ is a skew-symmetric matrix

$$\boldsymbol{X}(\boldsymbol{x}) = \begin{bmatrix} 0 & -x_1 & -x_2 \\ x_1 & 0 & -x_3 \\ x_2 & x_3 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3},$$

with $\boldsymbol{x} = [x_1, x_2, x_3]^T \in \mathbb{R}^{3\times 1}$. We use the matrix exponential $\exp(\boldsymbol{X})$ to map a point $\boldsymbol{x} \in \mathbb{R}^3$ onto the Stiefel manifold $\mathcal{V}_{3,3}$.

We linearize the matrix exponential by using a first-order expansion of the matrix exponential

$$\boldsymbol{\Omega}(\boldsymbol{x}) \approx \boldsymbol{I}_3 + \boldsymbol{X}.$$

Using this linearization[4] in (8.45) we get

$$\boldsymbol{Q} \approx \boldsymbol{Q}_k(\boldsymbol{I}_3 + \boldsymbol{X}).$$

We can then express $\boldsymbol{L}\mathrm{vec}(\boldsymbol{Q})$ as

$$\boldsymbol{L}\mathrm{vec}(\boldsymbol{Q}) \approx \boldsymbol{L}\mathrm{vec}(\boldsymbol{Q}_k) + \boldsymbol{L}\mathrm{vec}(\boldsymbol{Q}_k\boldsymbol{X}),$$

which is a function of $\boldsymbol{x}$, i.e.,

$$f_k(\boldsymbol{x}) = \boldsymbol{L}\mathrm{vec}(\boldsymbol{Q}_k) + \boldsymbol{J}\boldsymbol{x}, \tag{8.47}$$

where

$$\boldsymbol{J} = \frac{\partial \boldsymbol{L}\mathrm{vec}(\boldsymbol{Q}_k\boldsymbol{X})}{\partial \boldsymbol{x}^T} \in \mathbb{R}^{K\times 3}. \tag{8.48}$$

[4] Instead of a matrix exponential, a Cayley transformation $\boldsymbol{\Omega}(\boldsymbol{x}) = (\boldsymbol{I}_3 + \boldsymbol{X})(\boldsymbol{I}_3 + \boldsymbol{X})^{-1}$ can be alternatively used, which can be then linearized by using a first-order expansion of $(\boldsymbol{I}_3 + \boldsymbol{X})^{-1} \approx \boldsymbol{I}_3 + \boldsymbol{X}$ (see [Viklands, 2008]). As a result, we get a similar expression $\boldsymbol{Q} \approx \boldsymbol{Q}_k(\boldsymbol{I}_3 + 2\boldsymbol{X})$.

Next, we solve the optimization problem in (8.44) iteratively as follows. Using (8.47) in (8.44), we can transform the unitarily constrained optimization problem into an unconstrained minimization problem. More specifically, during the k-th iteration we compute the Gauss-Newton search direction by minimizing the following unconstrained LS problem

$$\begin{aligned} \boldsymbol{x}_k &= \arg\min_{\boldsymbol{x}\in\mathbb{R}^3} \quad \psi(\boldsymbol{x}) = \|\boldsymbol{z} - f_k(\boldsymbol{x})\|_2^2 \\ &= (\boldsymbol{J}^T\boldsymbol{J})^{-1}\boldsymbol{J}^T(\boldsymbol{z} - \boldsymbol{L}\text{vec}(\boldsymbol{Q}_k)), \end{aligned} \tag{8.49}$$

and subsequently compute the rotation update

$$\boldsymbol{Q}_{k+1} = \boldsymbol{Q}_k\boldsymbol{\Omega}(\alpha_k\boldsymbol{x}_k).$$

Here, α_k is the step size. The optimal step size is obtained by solving

$$\alpha_k = \arg\min_{\alpha\in(0,1]} \quad \|\boldsymbol{z} - \boldsymbol{L}\text{vec}(\boldsymbol{Q}_k\boldsymbol{\Omega}(\alpha\boldsymbol{x}_k))\|_2^2, \tag{8.50}$$

whose solution is the root of the polynomial equation obtained by expanding the matrix exponential [Viklands, 2008], or can be computed simply by line search. With a good initial point and at low noise levels we can take $\alpha_k = 1$.

The solution $\widehat{\boldsymbol{Q}}_{\text{SCLS}}$ of the SCLS algorithm can be used as an initial point for the iterative algorithm. Alternatively, the initial point can be computed by orthonormalizing the solution of the unconstrained LS solution $\widehat{\boldsymbol{Q}}_{\text{LS}}$. The latter orthonormalization procedure solves a special case of OPP, and is given as

$$\begin{aligned} \boldsymbol{Q}_0 &:= \arg\min_{\boldsymbol{Q}} \|\boldsymbol{Q} - \widehat{\boldsymbol{Q}}_{\text{LS}}\|_F^2 \quad \text{s.to} \quad \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_3 \\ &= (\widehat{\boldsymbol{Q}}_{\text{LS}}\widehat{\boldsymbol{Q}}_{\text{LS}}^T)^{-1/2}\widehat{\boldsymbol{Q}}_{\text{LS}}. \end{aligned} \tag{8.51}$$

8.C Proof of Theorem 8.2

We prove the claim of Theorem 8.2 by contradiction. Let there exist an unbiased constrained estimator $\widehat{\boldsymbol{Q}}$ such that $\widehat{\boldsymbol{Q}} \in \mathcal{V}_{3,3}$. Then $\widehat{\boldsymbol{Q}} = \boldsymbol{Q} + \boldsymbol{\Psi}$ where $\boldsymbol{\Psi}$ is the estimation error such that $\mathbb{E}\{\widehat{\boldsymbol{Q}}\} = \boldsymbol{Q}$ or $\mathbb{E}\{\boldsymbol{\Psi}\} = \boldsymbol{0}$. Since $\widehat{\boldsymbol{Q}} \in \mathcal{V}_{3,3}$, we have $\widehat{\boldsymbol{Q}}\widehat{\boldsymbol{Q}}^T = \boldsymbol{I}_3$, and hence

$$(\boldsymbol{Q} + \boldsymbol{\Psi})(\boldsymbol{Q} + \boldsymbol{\Psi})^T = \boldsymbol{I}_3. \tag{8.52}$$

Using $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}_3$ and taking expectations on both sides, (8.52) can be further simplified to

$$\text{tr}\{\mathbb{E}\{\boldsymbol{\Psi}\}\boldsymbol{Q}^T\} + \text{tr}\{\boldsymbol{Q}\mathbb{E}\{\boldsymbol{\Psi}^T\}\} = -\mathbb{E}\{\|\boldsymbol{\Psi}\|_F^2\}. \tag{8.53}$$

Due to the assumption that $\mathbb{E}\{\boldsymbol{\Psi}\} = \boldsymbol{0}$, the left-hand side of (8.53) is zero, but, the right-hand side is strictly less than zero. Hence a contradiction occurs, unless the noise is zero. □

8.D Proof of Theorem 8.3

The proof from [Arun, 1992] is provided here to aid the understanding of the proof of the next theorem. For any $\boldsymbol{Q}$, we can re-write the constraint in (8.36) as

$$\left[\begin{array}{c|c} \boldsymbol{Q} & -\boldsymbol{I}_3 \end{array}\right]\begin{bmatrix} \bar{\boldsymbol{E}} \\ \check{\boldsymbol{N}} \end{bmatrix} = -\left[\begin{array}{c|c} \boldsymbol{Q} & -\boldsymbol{I}_3 \end{array}\right]\begin{bmatrix} \bar{\boldsymbol{C}} \\ \check{\boldsymbol{D}} \end{bmatrix}.$$

Using the unitary constraint on $\boldsymbol{Q}$, and pseudo-inverting the wide matrix $\left[\begin{array}{c|c} \boldsymbol{Q} & -\boldsymbol{I}_{3(N-1)} \end{array}\right]$ we get

$$\begin{aligned} \begin{bmatrix} \bar{\boldsymbol{E}} \\ \check{\boldsymbol{N}} \end{bmatrix} &= -\frac{1}{2}\begin{bmatrix} \boldsymbol{Q}^T \\ -\boldsymbol{I}_3 \end{bmatrix}\left[\begin{array}{c|c} \boldsymbol{Q} & -\boldsymbol{I}_3 \end{array}\right]\begin{bmatrix} \bar{\boldsymbol{C}} \\ \check{\boldsymbol{D}} \end{bmatrix} \\ &= -\frac{1}{2}\begin{bmatrix} \boldsymbol{I}_3 & -\boldsymbol{Q}^T \\ -\boldsymbol{Q} & \boldsymbol{I}_3 \end{bmatrix}\begin{bmatrix} \bar{\boldsymbol{C}} \\ \check{\boldsymbol{D}} \end{bmatrix} \\ &= -\frac{1}{2}\begin{bmatrix} \bar{\boldsymbol{C}} - \boldsymbol{Q}^T\check{\boldsymbol{D}} \\ \check{\boldsymbol{D}} - \boldsymbol{Q}\bar{\boldsymbol{C}} \end{bmatrix}. \end{aligned}$$

We can now re-write the objective in (8.36) to compute the minimum-norm square solution

$$\begin{aligned} &\operatorname{tr}\left\{\left[\begin{array}{c|c} \bar{\boldsymbol{E}}^T & \check{\boldsymbol{N}}^T \end{array}\right]\begin{bmatrix} \bar{\boldsymbol{E}} \\ \check{\boldsymbol{N}} \end{bmatrix}\right\} \\ &= \frac{1}{2}\operatorname{tr}(\bar{\boldsymbol{C}}^T\bar{\boldsymbol{C}} - \check{\boldsymbol{D}}^T\boldsymbol{Q}\bar{\boldsymbol{C}} - \bar{\boldsymbol{C}}^T\boldsymbol{Q}^T\check{\boldsymbol{D}} + \check{\boldsymbol{D}}^T\check{\boldsymbol{D}}) \\ &= \frac{1}{2}\|\bar{\boldsymbol{C}}\|_F^2 - \operatorname{tr}(\boldsymbol{Q}\bar{\boldsymbol{C}}\check{\boldsymbol{D}}^T) + \frac{1}{2}\|\check{\boldsymbol{D}}\|_F^2. \end{aligned}$$

The solution to the SCTLS problem is then obtained by optimizing the term depending only on $\boldsymbol{Q}$, i.e., by maximizing $\operatorname{tr}\{\boldsymbol{Q}\bar{\boldsymbol{C}}\check{\boldsymbol{D}}^T\}$. This is the same cost as that of the SCLS problem (see [Golub and Van Loan, 1996, pg. 601]). Hence, the solution to the unitarily constrained TLS problem is

$$\widehat{\boldsymbol{Q}}_{\text{SCTLS}} = \boldsymbol{V}\boldsymbol{U}^T, \tag{8.54}$$

where the matrices $\boldsymbol{V}$ and $\boldsymbol{U}$ are obtained by computing the SVD of $\check{\boldsymbol{D}}\bar{\boldsymbol{C}}^T = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T$. □

8.E Proof of Theorem 8.4

For any $\boldsymbol{Q}$ the constraint in the optimization problem (8.39) can be written as

$$\begin{aligned} &\left[\begin{array}{c|c} (\boldsymbol{I}_{3(N-1)} \otimes \bar{\boldsymbol{A}}\boldsymbol{Q}) & -\boldsymbol{I}_K \end{array}\right]\begin{bmatrix} \bar{\boldsymbol{e}} \\ \tilde{\boldsymbol{n}} \end{bmatrix} \\ &\qquad = -\left[\begin{array}{c|c} (\boldsymbol{I}_{3(N-1)} \otimes \bar{\boldsymbol{A}}\boldsymbol{Q}) & -\boldsymbol{I}_K \end{array}\right]\begin{bmatrix} \bar{\boldsymbol{c}} \\ \tilde{\boldsymbol{d}} \end{bmatrix}. \end{aligned} \tag{8.55}$$

Multiplying both sides of (8.55) with the pseudo-inverse of the wide-matrix $[(\boldsymbol{I}_{3(N-1)} \otimes \bar{\boldsymbol{A}}\boldsymbol{Q}) \,|\, -\boldsymbol{I}_K]$ given by

$$\begin{aligned}&\left[\begin{array}{c|c} (\boldsymbol{I}_{3(N-1)} \otimes \bar{\boldsymbol{A}}\boldsymbol{Q}) & -\boldsymbol{I}_K \end{array}\right]^{\dagger}\\ &\quad= \begin{bmatrix} \boldsymbol{I}_{3(N-1)} \otimes \boldsymbol{Q}^T\bar{\boldsymbol{A}}^T \\ -\boldsymbol{I} \end{bmatrix} \left((\boldsymbol{I}_{3(N-1)} \otimes \bar{\boldsymbol{A}}\bar{\boldsymbol{A}}^T) + \boldsymbol{I}_K\right)^{-1},\end{aligned}$$

we get the minimum-norm solution $\boldsymbol{k} = [\bar{\boldsymbol{e}}^T, \tilde{\boldsymbol{n}}^T]^T$ to the system of equations in (8.55) which is given by

$$\begin{bmatrix} \bar{\boldsymbol{e}} \\ \tilde{\boldsymbol{n}} \end{bmatrix} = -\begin{bmatrix} (\boldsymbol{I}_{3(N-1)} \otimes \boldsymbol{Q}^T\bar{\boldsymbol{A}}^T) \\ -\boldsymbol{I}_K \end{bmatrix} \left((\boldsymbol{I}_{3(N-1)} \otimes \bar{\boldsymbol{A}}\bar{\boldsymbol{A}}^T) + \boldsymbol{I}_K\right)^{-1} \left[\begin{array}{c|c} (\boldsymbol{I}_{3(N-1)} \otimes \bar{\boldsymbol{A}}\boldsymbol{Q}) & -\boldsymbol{I}_K \end{array}\right] \begin{bmatrix} \bar{\boldsymbol{c}} \\ \tilde{\boldsymbol{d}} \end{bmatrix}. \tag{8.56}$$

Assuming that the covariance matrix of the perturbation vector is a scaled identity matrix, it is straightforward to verify that the objective in (8.40) using (8.56) simplifies to

$$\operatorname{tr}\left\{\left[\begin{array}{c|c} \bar{\boldsymbol{e}}^T & \tilde{\boldsymbol{n}}^T \end{array}\right] \boldsymbol{R}_{\boldsymbol{k}} \begin{bmatrix} \bar{\boldsymbol{e}} \\ \tilde{\boldsymbol{n}} \end{bmatrix}\right\} = \|\boldsymbol{\Lambda}^{1/2}(\tilde{\boldsymbol{d}} - (\bar{\boldsymbol{C}}^T \otimes \bar{\boldsymbol{A}})\boldsymbol{q})\|_2^2,$$

where

$$\boldsymbol{\Lambda}^{1/2} = \left(\sigma_{\mathrm{e}}^{-2}(\boldsymbol{I}_{3(N-1)} \otimes \bar{\boldsymbol{A}}\bar{\boldsymbol{A}}^T) + \boldsymbol{R}_{\tilde{\boldsymbol{n}}}^{-1}\right)^{1/2} \left((\boldsymbol{I}_{3(N-1)} \otimes \bar{\boldsymbol{A}}\bar{\boldsymbol{A}}^T) + \boldsymbol{I}_K\right)^{-1}. \tag{8.57}$$

Hence, the solution to the optimization problem (8.40) is equivalent to a specifically weighted CLS.

□

Part III

Outlook

Chapter 9

Conclusions and Future Research Directions

Contents

9.1 Concluding remarks

Pervasive sensors collect massive amounts of data. As a consequence, it is becoming increasingly difficult to store and ship the acquired data to a central location for signal/data processing. Therefore, to overcome these problems, we have focused on developing sensing mechanisms to extract as much information as possible yet collecting fewer data, and thereby reducing storage and communication requirements. In particular, in the first part of this thesis, we have addressed the question "*How can task-cognition be exploited to reduce the costs of sensing as well as the related storage and communications requirements?*" To answer this question, the tool that we have exploited for data reduction is sparse sensing. It consists of a deterministic sparse sensing function (guided by a sparse vector) that is optimally designed to achieve a desired inference performance with the reduced number of samples.

The sparse sensing model to acquire continuous-domain and discrete-domain observations, which we term as *continuous sparse sensing* and *discrete sparse sensing*, respectively, has been derived in Chapter 2. One of the fundamental differences with compressive sensing (a well-known data reduction mechanism used for sparse signal recovery) is that in sparse sensing the underlying signals need not be sparse. This allows us to focus on general signal processing tasks such as estimation, filtering, and detection. The developed theory can be applied to sensor placement, sensor/sample selection, data compression, with an impact on a range of signal processing applications.

Chapters 3, 4, and 5 have been, respectively, dedicated to discrete sparse sensing for estimation, filtering, and detection tasks. In all these chapters, the main questions that we have answered are: *"What are the reasonable inference performance metrics for estimation, filtering, and detection tasks?"* and *"Can we efficiently optimize (e.g., using a polynomial time algorithm) such inference performance metrics to obtain the sparse samplers of interest?"*. To this end, we have derived convex risk functions that include the sparse vector for specific inference tasks and thus, can be optimized via convex programming. Relying on the assumption that the data model is perfectly known, we have shown that measures such as the Cramér-Rao bound (CRB) for nonlinear estimation, posterior CRB for filtering, and Ali-Silvey distances (i.e., Bhattacharyya distance, Kullback-Leibler distance, or J-divergence) for binary hypothesis testing can be used as reasonable inference performance metrics. Furthermore, we have also shown that all these (ensemble) inference performance measures (A) can be computed in closed form and are independent of the observations, thus enabling offline (or data-independent) designs, and (B) can be optimized using a convex program, in many cases of interest. The optimization problem is relatively easier to solve when the observations are independent, irrespective of their distributions. By choosing appropriate instantaneous (instead of ensemble) inference performance metrics (that depend on data), the unifying approach presented in this thesis can be extended to *data-driven sparse sensing* for machine learning problems, like clustering or classification, which provides room for further research.

In Chapter 6, we have presented the *continuous sparse sensing* framework. Here, unlike the discrete case, we have shown that it is possible to sample in between the grid points, that is, anywhere in the continuous observation domain. We have done this by modeling an off-the-grid sampling point as a discrete sampling point plus a perturbation. Then, the smallest set of possible discrete sampling points is searched for, along with the best possible perturbations, in order to reach the desired inference performance. We have solved the continuous sparse sensing problem via convex programming for linear inverse problems. There is still room for extending this framework (especially the solver) to other inference tasks, like hypothesis testing and nonlinear estimation/filtering with correlated inputs.

Next, in the second part of this thesis, we have focused on some signal processing problems related to distributed sensing and sensor networks. More specifically, we have addressed the questions: *"How can wireless communications be exploited to synchronize spatially separated sample clocks?"*, and *"How can we extend the classical localization paradigm to localize a sensing platform by exploiting the knowledge of the sensor placement on the platform?"*.

In Chapter 7, we have provided a solution to synchronize the sample clocks of the sensors by fully exploiting the broadcast nature of the wireless medium, and using the time-of-flight measurements of the messages. We have assumed an affine clock model and that there exists at least one line-of-sight path between the nodes, which are reasonable in many cases of interest. Specifically, we have solved for all the unknown clock skews and clock offsets using least squares. The proposed estimator is shown to be efficient, asymptotically meeting the theoretical CRB. However, when there is no line-of-sight path, the wireless clock synchronization remains challenging.

Finally, in Chapter 8, we have solved the rigid body localization (i.e., position and orientation estimation) problem using only the distance measurements. To achieve this, we have exploited the prior information about the sensor placement on the platform. We have shown that the position and orientation of the rigid platform can be represented by a rotation matrix and translation vector. We have proposed a number of constrained least squares estimators, and we have also derived the theoretical limits (i.e., CRB) for the position and orientation estimation problem using distance measurements.

9.2 Directions for future research

The research for this thesis has generated a number of new challenging research questions, which remain open. In what follows, we conclude this thesis by listing some directions for future research.

9.2.1 Sparse sensing for constrained inverse problems

In Chapter 3, the focus was on designing sparse sensing mechanisms for nonlinear estimation problems. In particular, the goal was to sparsely sense the signal $\boldsymbol{x}$ that is related to an unknown (nonrandom) parameter vector $\boldsymbol{\theta}$ through a known nonlinear model in order to achieve a desired estimation accuracy. We have used scalar functions of the CRB as the inference performance metric to quantify the estimation accuracy. In many applications, the parameter space may be restricted to a known subset of the Euclidean space through smooth functional constraints on the parameters. The question then is how can we design sparse sensing functions that take into account such constraints on the unknown parameter. Note that this is different from the Bayesian approach that we have discussed in Remark 3.4 of Chapter 3, where the assumption was that the parameter is random and the prior information was related to its distribution.

If the parameter vector $\boldsymbol{\theta}$ is subject to some continuously differentiable constraints $\boldsymbol{g}(\boldsymbol{\theta}) = \mathbf{0}$, then with these constraints, the resulting constrained CRB (CCRB) is lower than the unconstrained CRB (cf. Section 8.4.2 from Chapter 8). In [Stoica and Ng, 1998], it is shown that the CCRB has the form

$$\mathbb{E}\{(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T\} \geq \boldsymbol{C}_{\text{CCRB}}(\boldsymbol{w}, \boldsymbol{\theta}) = \boldsymbol{M}(\boldsymbol{M}^T\boldsymbol{F}(\boldsymbol{w}, \boldsymbol{\theta})\boldsymbol{M})^{-1}\boldsymbol{M}^T,$$

where $\widehat{\boldsymbol{\theta}}$ denotes the estimate of $\boldsymbol{\theta}$, $\boldsymbol{F}(\boldsymbol{w}, \boldsymbol{\theta})$ is the FIM for the unconstrained estimation problem as defined in (3.2), and $\boldsymbol{M}$ is an isometry matrix whose columns form an orthonormal basis for the nullspace of the gradient matrix $\partial\boldsymbol{g}(\boldsymbol{\theta})/\partial\boldsymbol{\theta}^T$. Here, $\boldsymbol{w}$ is the sparse sensing vector to be designed. The question is, can we optimize the risk functions (cf. Section 3.3 of Chapter 3) $f(\boldsymbol{w}) := \text{tr}\{\boldsymbol{C}_{\text{CCRB}}(\boldsymbol{w}, \boldsymbol{\theta})\}$ or $f(\boldsymbol{w}) := \lambda_{\max}\{\boldsymbol{C}_{\text{CCRB}}(\boldsymbol{w}, \boldsymbol{\theta})\}$ using a convex program.

9.2.2 Sparse sensing for composite detection problems

In Chapter 5, the focus was on choosing the best subset of observations for a binary hypothesis testing problem. In particular, the assumption was that the model parameters were perfectly known.

That is, our goal was to sparsely sense the signal $\boldsymbol{x}$ that follows, for example, the conditional distributions:

$$\begin{aligned}\mathcal{H}_0: &\quad \boldsymbol{x} \sim h_0(\boldsymbol{\theta}_0, \boldsymbol{n}) \\ \mathcal{H}_1: &\quad \boldsymbol{x} \sim h_1(\boldsymbol{\theta}_1, \boldsymbol{n}),\end{aligned}$$

where we assumed that the parameter vectors $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ of the functions h_0 and h_1 as well as the probability density function of the noise $\boldsymbol{n}$ are perfectly known. When the parameter vectors $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ are unknown, we generally solve the so-called composite binary hypothesis testing problem [Scharf, 1991].

For composite hypothesis testing problems, the question is, how to design sparse sensing mechanisms. Clearly, the inference performance metrics that were discussed in Section 5.4 of Chapter 5 will now depend on the unknown parameters. Consequently, the risk function of the form $f(\boldsymbol{w}, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)$ has to be optimized over the sparse sensing vector $\boldsymbol{w}$, $\forall \boldsymbol{\theta}_0$ and $\forall \boldsymbol{\theta}_1$. This is reminiscent of the risk functions of the nonlinear estimation problem discussed in Chapter 3. Therefore, a way to solve the sparse sensing problem would be to grid the parameter space (every grid point would result in an additional performance constraint), or alternatively marginalize the unknown parameters using the prior statistical information, when available. This leads to a number of interesting questions, such as: "is optimizing the signal-to-noise ratio optimal in terms of the error probabilities for Gaussian composite hypothesis testing?" and "how much data reduction can be achieved via sparse sensing for composite hypothesis testing as compared to (nonlinear) parameter estimation?".

9.2.3 Distributed sparse sensing

Throughout this thesis, we have assumed that the signals were available only at a single instance in space, frequency, or time. Suppose now that they are available at multiple instances, that is to say, the signals are distributed. For example, imagine spatially separated sensors observing continuous-time phenomena. The question then is, how we can jointly design the different sparse sampling functions at different physical sensors to reach the best compression rate per sensor, subject to specific performance guarantees.

While deriving sparse sensing mechanisms for estimation, filtering, or detection, we always considered a single realization of the signal $\boldsymbol{x}$ was available and designed the sparse sensing function $\boldsymbol{\Phi}(\boldsymbol{w})$ to obtain the sparsely sensed signal $\boldsymbol{y}$. Now if multiple realizations are available, e.g., when $\boldsymbol{x}$ is a space-domain signal, we could gather this signal at different time intervals or in different frequency bands. All these realizations can be sparsely sensed as before, by employing the same sensing function at different sensors. This might not be the best thing to do, as we do not fully exploit the available diversity to acquire the available multiple realizations using different sensing functions. In order to exploit the available diversity, clearly we should have the freedom to design different sparse sensing functions at different physical sensors. For example, let $\boldsymbol{x}_1 = [x_{1,1}, x_{2,1}, \dots, x_{M,1}]^T$ and $\boldsymbol{x}_2 = [x_{1,2}, x_{2,2}, \dots, x_{M,2}]^T$ be the two available temporal snapshots of the spatial signal $\boldsymbol{x} = [\boldsymbol{x}_1^T, \boldsymbol{x}_2^T]^T$. We acquire such distributed multidimensional signals using two sparse vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ related to two temporal snapshots (instead of one sparse vector) as

$$\boldsymbol{y} = \boldsymbol{\Phi}(\boldsymbol{w}_1, \boldsymbol{w}_2)\boldsymbol{x},$$

where $\boldsymbol{\Phi}(\cdot, \cdot)$ denotes the sparse sensing function. The question is how to design such sparse sensing functions.

9.2.4 Clock synchronization in non line-of-sight scenarios

The algorithms for wireless clock synchronization derived in Chapter 7 were based on time-of-flight measurements of messages between the sensor nodes. This relies on a fundamental assumption that there exists a line-of-sight path between the sensor nodes. The framework in Chapter 7 can be extended to multipath environments that include the line-of-sight path. This can, for example, be done by resolving the line-of-sight path using techniques like ray tracing or outlier rejection methods (i.e., to look for the consistent system of equations). However, when there is no line-of-sight path, synchronizing sample clocks is a challenging problem.

One possible way to synchronize clocks is to use a phased-lock loop — a well-known phase and frequency tracking tool used in wireless communications. Let us define $a(t)$ as the output of an oscillator [Allan, 1987]

$$a(t) = a_0 \sin(2\pi f_0 t + \psi(t)),$$

where f_0 is the nominal frequency, a_0 is the constant amplitude, and $\psi(t)$ captures all the residual phase deviations. For an ideal clock, $\psi(t) = 0, \forall t$. In Chapter 7, we have modeled only the first order clock deviations and ignored the higher-order terms. That is, the assumption was

$$\psi(t) \approx \phi + \omega t = \phi + 2\pi f t,$$

where t is the *true* time, ϕ is the phase offset (or the clock offset) and $\omega = 2\pi f$ is the frequency offset (or the clock skew). The question is, can we use a phased-lock loop (or rather a dual phased-lock loop) discussed in [Johnson and Sethares, 2004, pp. 200-202] at each sensor to track the phase and frequency offsets with respect to a synchronization beacon transmitted from a sensor with a relatively stable clock. The second question is, what are the theoretical limits on synchronization based on such an approach.

9.2.5 Sensor, source placement, and closing the loop

In the first part of this thesis, the focus was on sensor placement, that is, to choose the best subset of sensor locations, subject to specific inference performance guarantees. The dual problem of sensor placement is the source placement problem. The source placement problem can be interpreted as choosing the best subset of source locations in order to generate a desired field. Naturally, one would think of closing this loop through a joint sensor and source placement mechanism.

We will illustrate this with the rigid body localization problem that was discussed in Chapter 8. In particular, the question is, what are the optimal sensor/anchor placement (on the fixed world) and optimal source placement on the rigid platform to achieve the desired localization accuracy. To realize this, we can express the Fisher information matrix (cf. (8.32)) in terms of two sparse vectors, where one vector is associated with the candidate anchor positions and the other one is associated with the candidate sensor positions. The resulting constrained CRB (cf. (8.33)) is then optimized to jointly minimize the number of sensors and anchors. The question of interest is, how efficiently can we solve this optimization problem.

In addition to the above list, there is still room for studying greedy (e.g., submodular) algorithms for continuous and discrete sparse sensing problems that were considered in this thesis. Sparse sensing for multiple hypothesis testing and developing a theory on relative position and orientation estimation between rigid bodies in an anchorless scenario, also remain open.

The ideas presented in this thesis with innovative perspectives on sparse sensing and distributed sensing are certainly pertinent to the design of modern sensing systems. We hope that these ideas will also open up new lines of research within the broad field of signal processing.

Notation

Vectors and matrices

x	scalar
$\boldsymbol{x}$	vector denoted by lower bold face letters.
$\boldsymbol{X}$	matrix denoted by upper bold face letters.
$\boldsymbol{X}^T$	Transpose of matrix $\boldsymbol{X}$.
$\boldsymbol{X}^H$	Hermitian of matrix $\boldsymbol{X}$.
$\text{diag}(\cdot)$	block diagonal matrix with the elements in its argument on the main diagonal.
$\text{diag}_\text{r}(\cdot)$	diagonal matrix with the argument on its diagonal but with the all-zero rows removed.
$[\boldsymbol{X}]_{i,j}$	denotes the (i, j)th entry of matrix $\boldsymbol{X}$.
$[\boldsymbol{x}]_i$	denotes the ith entry of vector $\boldsymbol{x}$.
$\mathbf{1}_N$	$N \times 1$ vector of all ones.
$\mathbf{0}_N$	$N \times 1$ vector of all zeros.
$\mathbf{I}_N$	identity matrix of size N.
$\boldsymbol{x}^{\odot 2}$	element-wise squaring of vector $\boldsymbol{x}$.
$\text{tr}\{\boldsymbol{X}\}$	Trace of matrix $\boldsymbol{X}$.
$\det\{\boldsymbol{X}\}$	Determinant of matrix $\boldsymbol{X}$.
$\lambda_{\min}\{\boldsymbol{X}\}$	minimum eigenvalue of a symmetric matrix $\boldsymbol{X}$.
$\lambda_{\max}\{\boldsymbol{X}\}$	maximum eigenvalue of a symmetric matrix $\boldsymbol{X}$.
$\boldsymbol{X} \geq \boldsymbol{Y}$	$\boldsymbol{X} - \boldsymbol{Y}$ is a positive semidefinite matrix
$\boldsymbol{X} \otimes \boldsymbol{Y}$	Kronecker product of matrix $\boldsymbol{X}$ and matrix $\boldsymbol{Y}$.
$\boldsymbol{X}^\dagger = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$	Pseudo inverse (or the left-inverse) of a full column-rank tall matrix $\boldsymbol{X}$.
$\boldsymbol{X}^\dagger = \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}$	Pseudo inverse (or the right-inverse) of a full row-rank wide matrix $\boldsymbol{X}$.
$\text{vec}(\boldsymbol{X})$	$MN \times 1$ vector formed by stacking the columns of an $M \times N$ matrix $\boldsymbol{X}$.
$\text{unvec}(\boldsymbol{X})$	$M \times N$ matrix $\boldsymbol{X}$ formed by the inverse $\text{vec}(\boldsymbol{X})$ operation on an $MN \times 1$ vector.
$\text{cond}(\boldsymbol{X})$	condition number of a matrix $\boldsymbol{X}$.

Stochastic processes

$\mathbb{E}\{\boldsymbol{x}\}$	Expected value of random vector $\boldsymbol{x}$.
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution with mean vector $\boldsymbol{\mu}$, covariance matrix $\boldsymbol{\Sigma}$.
$\Pr(H)$	Probability of event H.

Norms and distances

$\\|\boldsymbol{w}\\|_0$	ℓ_0-(quasi) norm, i.e., number of non-zero entries of vector $\boldsymbol{w}$.
$\\|\boldsymbol{w}\\|_1$	ℓ_1-norm of vector $\boldsymbol{w}$.
$\\|\boldsymbol{w}\\|_2$	Euclidean (or ℓ_2-)norm of vector $\boldsymbol{w}$.
$\\|\boldsymbol{w}\\|_{\mathbf{A}}^2$	weighted squared ℓ_2-norm $\boldsymbol{w}^T\mathbf{A}\boldsymbol{w}$.
$\\|\boldsymbol{W}\\|_{2,0}$	ℓ_2/ℓ_0-(quasi) norm counts the number of nonzero rows of matrix $\boldsymbol{Z}$.

Sets

$\mathbb{R}$	Real numbers
$\mathbb{R}_+$	Nonnegative real numbers
$\mathbb{C}$	Complex numbers
$\mathbb{R}^N$	Real length-N vectors
$\mathbb{R}^{M\times N}$	Real $M \times N$ matrices
$\mathbb{C}^N$	Complex length-N vectors
$\mathbb{C}^{M\times N}$	Complex $M \times N$ vectors
$\{0,1\}^N$	Boolean length-N vectors
$\{0,1\}^{M\times N}$	Boolean $M \times N$ matrices
$\mathbb{Z}$	Integers
$\mathbb{N}$	Natural numbers
$\mathbb{S}^N$	Symmetric $N \times N$ matrices
$\mathbb{S}_+^N$	Symmetric positive semi-definite $N \times N$ matrices
$\mathbb{S}_{++}^N$	Symmetric positive definite $N \times N$ matrices
$\lvert\mathcal{A}\rvert$	cardinality of set $\mathcal{A}$

Bibliography

[Ali and Silvey, 1966] Ali, S. and Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *J. of the Royal Stat. Society. Series B (Methodological)*, pages 131–142.

[Allan, 1987] Allan, D. W. (1987). Time and frequency(time-domain) characterization, estimation, and prediction of precision clocks and oscillators. *IEEE Trans. on ultrasonics, ferroelectrics, and frequency control*, 34(6):647–654.

[Angelosante et al., 2009] Angelosante, D., Giannakis, G. B., and Grossi, E. (2009). Compressed sensing of time-varying signals. In *Proc. of 16th International Conference on Digital Signal Processing, 2009*, pages 1–8. IEEE.

[Appadwedula et al., 2008] Appadwedula, S., Veeravalli, V. V., and Jones, D. L. (2008). Decentralized detection with censoring sensors. *IEEE Trans. Signal Process.*, 56(4):1362–1373.

[Arun, 1992] Arun, K. (1992). A unitarily constrained total least squares problem in signal processing. *SIAM Journal on Matrix Analysis and Applications*, 13(3):729–745.

[Bahr and Bucklew, 1990] Bahr, R. K. and Bucklew, J. A. (1990). Optimal sampling schemes for the Gaussian hypothesis testing problem. *IEEE Trans. Acoust., Speech, Signal Process.*, 38(10):1677–1686.

[Bajovic et al., 2011] Bajovic, D., Sinopoli, B., and Xavier, J. (2011). Sensor selection for event detection in wireless sensor networks. *IEEE Trans. Signal Process.*, 59(10):4938–4953.

[Baraniuk, 2007] Baraniuk, R. G. (2007). Compressive sensing. *IEEE Signal Process. Mag.*, 24(4).

[Bertsekas, 1999] Bertsekas, D. P. (1999). *Nonlinear programming*. Athena Scientific optimization and computation series. Athena Scientific.

[Blum and Sadler, 2008] Blum, R. S. and Sadler, B. M. (2008). Energy efficient signal detection in sensor networks using ordered transmissions. *IEEE Trans. Signal Process.*, 56(7):3229–3235.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.

[Boyd et al., 2003] Boyd, S., Xiao, L., and Mutapcic, A. (2003). Subgradient methods. `http://www.stanford.edu/class/ee364b/notes/subgrad_method_notes.pdf`.

[Cambanis and Masry, 1983] Cambanis, S. and Masry, E. (1983). Sampling designs for the detection of signals in noise. *IEEE Trans. Inf. Theory*, 29(1):83–104.

[Candès and Wakin, 2008] Candès, E. and Wakin, M. (2008). An introduction to compressive sampling. *IEEE Signal Process. Mag.*, 25(2):21–30.

[Candès et al., 2008] Candès, E., Wakin, M., and Boyd, S. (2008). Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14:877–905.

[Carmi, 2010] Carmi, A. (2010). Sensor scheduling via compressed sensing. In *Proc. of 13th Conference on Information Fusion (FUSION)*, pages 1–8.

[Carmi et al., 2010] Carmi, A., Gurfil, P., and Kanevsky, D. (2010). Methods for sparse signal recovery using Kalman filtering with embedded pseudo-measurement norms and quasi-norms. *IEEE Trans. Signal Process.*, 58(4):2405–2409.

[Chamberland and Veeravalli, 2007] Chamberland, J. and Veeravalli, V. (2007). Wireless sensors in distributed detection applications. *IEEE Signal Process. Mag.*, 24(3):16–25.

[Chaturvedi et al., 2011] Chaturvedi, N. A., Sanyal, A. K., and Mc Clamroch, N. H. (2011). Rigid-body attitude control. *IEEE Control Syst. Mag.*, 31(3):30–51.

[Chepuri and Leus, 2015a] Chepuri, S. P. and Leus, G. (2015a). Sparse sensing for distributed Gaussian detection. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[Chepuri and Leus, 2015b] Chepuri, S. P. and Leus, G. (2015b). Sparsity-promoting sensor selection for non-linear measurement models. *IEEE Trans. Signal Process.*, 63(3):684–698.

[Chepuri et al., 2013a] Chepuri, S. P., Leus, G., and van der Veen, A.-J. (2013a). Position and orientation estimation of a rigid body: Rigid body localization. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5185–5189.

[Chepuri et al., 2013b] Chepuri, S. P., Leus, G., and van der Veen, A.-J. (2013b). Sparsity-exploiting anchor placement for localization in sensor networks. In *Proc. of the 21st European Signal Processing Conference (EUSIPCO)*, pages 1–5.

[Chepuri et al., 2013c] Chepuri, S. P., Rajan, R., Leus, G., and van der Veen, A.-J. (2013c). Joint clock synchronization and ranging: Asymmetrical time-stamping and passive listening. *IEEE Signal Process. Lett.*, 20(1):51–54.

[Chepuri et al., 2013d] Chepuri, S. P., Simonetto, A., Leus, G., and van der Veen, A.-J. (2013d). Tracking position and orientation of a mobile rigid body. In *Proc. of of the IEEE Workshop on Comp. Adv. in Multi-Sensor Adaptive Proc. (CAMSAP)*, pages 37–40.

[Cheung and So, 2005] Cheung, K. W. and So, H. C. (2005). A multidimensional scaling framework for mobile location using time-of-arrival measurements. *IEEE Trans. Signal Process.*, 53(2):460–470.

[Cohen, 1992] Cohen, C. (1992). Attitude determination using GPS. *Ph.D. dissertation, Dep. of Aeronautics and Astronautics, Stanford University*.

[Costa et al., 2006] Costa, J. A., Patwari, N., and Hero, III, A. O. (2006). Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Trans. Sen. Netw.*, 2(1):39–64.

[Cover and Thomas, 2012] Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

[Diebel, 2006] Diebel, J. (2006). Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58:15–16.

[Donoho, 2006a] Donoho, D. (2006a). For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics*, 59(6):797–829.

[Donoho, 2006b] Donoho, D. L. (2006b). Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306.

[Ekanadham et al., 2011] Ekanadham, C., Tranchina, D., and Simoncelli, E. (2011). Recovery of sparse translation-invariant signals with continuous basis pursuit. *IEEE Trans. Signal Process.*, 59(10):4735–4744.

[El Badawy et al., 2014] El Badawy, D., Ranieri, J., and Vetterli, M. (2014). Near-optimal sensor placement for signals lying in a union of subspaces. In *Proc. of 22nd European Signal Processing Conference (EUSIPCO 2014)*.

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

[Eldén and Park, 1999] Eldén, L. and Park, H. (1999). A Procrustes problem on the Stiefel manifold. *Numerische Mathematik*, 82:599–619. 10.1007/s002110050432.

[Farahmand et al., 2014] Farahmand, S., Giannakis, G. B., Leus, G., and Tian, Z. (2014). Tracking target signal strengths on a grid using sparsity. *EURASIP Journal on Advances in Signal Processing*, 2014(1):7.

[Ford et al., 1989] Ford, I., Titterington, D. M., and Kitsos, C. P. (1989). Recent advances in nonlinear experimental design. *Technometrics*, 31(1):49–60x.

[Freris et al., 2010] Freris, N., Kowshik, H., and Kumar, P. R. (2010). Fundamentals of large sensor networks: Connectivity, capacity, clocks, and computation. *Proc. of the IEEE*, 98(11):1828 –1846.

[Friedman et al., 2010] Friedman, J., Hastie, T., and Tibshirani, R. (2010). A note on the group LASSO and a sparse group LASSO. *arXiv preprint arXiv:1001.0736*.

[Fu et al., 2012] Fu, Y., Ling, Q., and Tian, Z. (2012). Distributed sensor allocation for multi-target tracking in wireless sensor networks. *IEEE Trans. Aerosp. Electron. Syst.*, 48(4):3538–3553.

[Gastpar et al., 2006] Gastpar, M., Vetterli, M., and Dragotti, P. L. (2006). Sensing reality and communicating bits: A dangerous liaison. *IEEE Signal Process. Mag.*, 23(4):70–83.

[Gezici et al., 2005] Gezici, S., Tian, Z., Giannakis, G. B., Kobayashi, H., Molisch, A. F., Poor, H. V., and Sahinoglu, Z. (2005). Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE Signal Process. Mag.*, 22(4):70–84.

[Golub and Van Loan, 1996] Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press.

[Grant and Boyd, 2012] Grant, M. and Boyd, S. (2012). CVX: Matlab software for disciplined convex programming, version 2.0 beta. `http://cvxr.com/cvx`.

[Grettenberg, 1963] Grettenberg, T. (1963). Signal selection in communication and radar systems. *IEEE Trans. Inf. Theory*, 9(4):265–275.

[Gustafsson and Gunnarsson, 2005] Gustafsson, F. and Gunnarsson, F. (2005). Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements. *IEEE Signal Process. Mag.*, 22(4):41 – 53.

[Haupt et al., 2009] Haupt, J., Nowak, R., and Castro, R. (2009). Adaptive sensing for sparse signal recovery. In *Proc. of IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009.*, pages 702–707. IEEE.

[Hernandez et al., 2004] Hernandez, M., Kirubarajan, T., and Bar-Shalom, Y. (2004). Multisensor resource deployment using Posterior Cramér-Rao bounds. *IEEE Trans. Aerosp. Electron. Syst.*, 40(2):399–416.

[Hoeffding, 1963] Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30.

[Hol, 2011] Hol, J. (2011). Sensor fusion and calibration of inertial sensors, vision, Ultra-Wideband and GPS. *Dissertations Linköping Studies in Science and Technology*, 1368.

[Hol et al., 2009] Hol, J., Dijkstra, F., Luinge, H., and Schon, T. (2009). Tightly coupled UWB/IMU pose estimation. In *Proc. of IEEE International Conference on Ultra-Wideband (ICUWB)*, pages 688–692.

[Horn, 1987] Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642.

[Horn et al., 1988] Horn, B. K. P., Hilden, H. M., and Negahdaripour, S. (1988). Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A*, 5(7):1127–1135.

[Ji et al., 2008] Ji, S., Xue, Y., and Carin, L. (2008). Bayesian compressive sensing. *IEEE Trans. Signal Process.*, 56(6):2346–2356.

[Jiang et al., 2013] Jiang, F., Chen, J., and Swindlehurst, A. L. (2013). Linearly reconfigurable Kalman filtering for a vector process. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[Johnson and Sethares, 2004] Johnson, C. R. and Sethares, W. A. (2004). *Telecommunication breakdown: concepts of communication transmitted via software-defined radio*, volume 1. Prentice Hall.

[Joshi and Boyd, 2009] Joshi, S. and Boyd, S. (2009). Sensor selection via convex optimization. *IEEE Trans. Signal Process.*, 57(2):451–462.

[Juang and Huang, 1997] Juang, J.-C. and Huang, G.-S. (1997). Development of GPS-based attitude determination algorithms. *IEEE Trans. Aerosp. Electron. Syst.*, 33(3):968 –976.

[Julier and LaViola, 2007] Julier, S. and LaViola, J. (2007). On Kalman filtering with nonlinear equality constraints. *IEEE Trans. Signal Process.*, 55(6):2774–2784.

[Kadota and Shepp, 1967] Kadota, T. and Shepp, L. A. (1967). On the best finite set of linear observables for discriminating two Gaussian signals. *IEEE Trans. Inf. Theory*, 13(2):278–284.

[Kailath, 1967] Kailath, T. (1967). The divergence and Bhattacharyya distance measures in signal selection. *IEEE Trans. Commun. Technol.*, 15(1):52–60.

[Kay, 1993] Kay, S. M. (1993). *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall.

[Kekatos and Giannakis, 2011] Kekatos, V. and Giannakis, G. (2011). From sparse signals to sparse residuals for robust sensing. *IEEE Trans. Signal Process.*, 59(7):3355–3368.

[Kekatos et al., 2012] Kekatos, V., Giannakis, G., and Wollenberg, B. (2012). Optimal placement of phasor measurement units via convex relaxation. *IEEE Trans. Power Syst.*, 27(3):1521–1530.

[Kobayashi and Thomas, 1967] Kobayashi, H. and Thomas, J. B. (1967). Distance measures and related criteria. In *Proc. 5th Annu. Allerton Conf. Circuit and System Theory*, pages 491–500.

[Krause, 2008] Krause, A. (2008). *Optimizing sensing: Theory and applications*. ProQuest.

[Krause and Guestrin, 2007] Krause, A. and Guestrin, C. (2007). Near-optimal observation selection using submodular functions. In *AAAI*, volume 7, pages 1650–1654.

[Krause et al., 2008a] Krause, A., McMahan, H. B., Guestrin, C., and Gupta, A. (2008a). Robust submodular observation selection. *Journal of Machine Learning Research*, 9(12).

[Krause et al., 2008b] Krause, A., Singh, A., and Guestrin, C. (2008b). Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284.

[Kullback, 2012] Kullback, S. (2012). *Information theory and statistics*. Courier Dover Publications.

[Kumar et al., 2015] Kumar, S., Zjajo, A., and van Leuken, R. (2015). Ctherm: An integrated framework for thermal-functional co-simulation of systems-on-chip. In *Proceedings of the 23rd IEEE/Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 674–681.

[Liu et al., 2014] Liu, S., Fardad, M., Varshney, P. K., and Masazade, E. (2014). Optimal periodic sensor scheduling in networks of dynamical systems. *IEEE Trans. Signal Process.*, 62(12):3055–3068.

[Lofberg, 2004] Lofberg, J. (2004). YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan.

[Malioutov et al., 2010] Malioutov, D. M., Sanghavi, S. R., and Willsky, A. S. (2010). Sequential compressed sensing. *IEEE J. Sel. Topics Signal Process.*, 4(2):435–444.

[Manton, 2002] Manton, J. H. (2002). Optimization algorithms exploiting unitary constraints. *IEEE Trans. Signal Process.*, 50(3):635–650.

[Marvasti, 2001] Marvasti, F. (2001). *Nonuniform sampling: theory and practice*, volume 1. Springer Science & Business Media.

[Marziliano and Vetterli, 2000] Marziliano, P. and Vetterli, M. (2000). Reconstruction of irregularly sampled discrete-time bandlimited signals with unknown sampling locations. *IEEE Transactions on Signal Processing*, 48(12):3462–3471.

[Masazade et al., 2012] Masazade, E., Fardad, M., and Varshney, P. (2012). Sparsity-promoting extended Kalman filtering for target tracking in wireless sensor networks. *IEEE Signal Process. Lett.*, 19(12):845–848.

[Matoušek and Gärtner, 2007] Matoušek, J. and Gärtner, B. (2007). *Understanding and using linear programming*, volume 168. Springer.

[Memik et al., 2008] Memik, S. O., Mukherjee, R., Ni, M., and Long, J. (2008). Optimizing thermal sensor allocation for microprocessors. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 27(3):516–527.

[Moon and Stirling, 2000] Moon, T. K. and Stirling, W. C. (2000). *Mathematical methods and algorithms for signal processing*, volume 1. Prentice hall New York.

[Nemhauser et al., 1978] Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions— I. *Mathematical Programming*, 14(1):265–294.

[Noh et al., 2008] Noh, K.-L., Serpedin, E., and Qaraqe, K. (2008). A new approach for time synchronization in wireless sensor networks: Pairwise broadcast synchronization. *IEEE Trans. Wireless Commun.*, 7(9):3318 –3322.

[Papadimitriou, 2003] Papadimitriou, C. H. (2003). *Computational complexity*. John Wiley and Sons Ltd.

[Patwari et al., 2005] Patwari, N., Ash, J. N., Kyperountas, S., Hero III, A. O., Moses, R., and Correal, N. (2005). Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Process. Mag.*, 22(4):54 – 69.

[Polyak et al., 2013] Polyak, B., Khlebnikov, M., and Shcherbakov, P. (2013). An LMI approach to structured sparse feedback design in linear control systems. In *Proc. of European Control Conference (ECC), 2013*, pages 833–838. IEEE.

[Pukelsheim, 1993] Pukelsheim, F. (1993). *Optimal design of experiments*, volume 50. SIAM.

[Quan et al., 2009] Quan, Z., Kaiser, W. J., and Sayed, A. H. (2009). Innovations diffusion: A spatial sampling scheme for distributed estimation and detection. *IEEE Trans. Signal Process.*, 57(2):738–751.

[Rago et al., 1996] Rago, C., Willett, P., and Bar-Shalom, Y. (1996). Censoring sensors: A low-communication-rate scheme for distributed detection. *IEEE Trans. Aerosp. Electron. Syst.*, 32(2):554–568.

[Rajan and van der Veen, 2011] Rajan, R. T. and van der Veen, A.-J. (2011). Joint ranging and clock synchronization for a wireless network. In *Proc. of 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 297 –300.

[Ranieri et al., 2014] Ranieri, J., Chebira, A., and Vetterli, M. (2014). Near-optimal sensor placement for linear inverse problems. *IEEE Trans. Signal Process.*, 62(5):1135–1146.

[Rothenberg, 1971] Rothenberg, T. (1971). Identification in parametric models. *Econometrica: Journal of the Econometric Society*, pages 577–591.

[Salhuana, 2012] Salhuana, L. (2012). Tilt sensing using linear accelerometers. In *Appl. note AN3461*. Freescale Semiconductor.

[Sandryhaila and Moura, 2014] Sandryhaila, A. and Moura, J. (2014). Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Process. Mag.*, 31(5):80–90.

[Scharf, 1991] Scharf, L. L. (1991). *Statistical signal processing*, volume 98. Addison-Wesley Reading, MA.

[Schizas, 2013] Schizas, I. (2013). Distributed informative-sensor identification via sparsity-aware matrix decomposition. *IEEE Trans. Signal Process.*, 61(18):4610–4624.

[Shamaiah et al., 2010] Shamaiah, M., Banerjee, S., and Vikalo, H. (2010). Greedy sensor selection: Leveraging submodularity. In *Proc. of 49th IEEE Conference on Decision and Control (CDC),*, pages 2572–2577. IEEE.

[Slavakis et al., 2014] Slavakis, K., Giannakis, G., and Mateos, G. (2014). Modeling and optimization for big data analytics:(statistical) learning tools for our era of data deluge. *IEEE Signal Process. Mag.*, 31(5):18–31.

[Stoica and Ng, 1998] Stoica, P. and Ng, B. C. (1998). On the Cramér-Rao bound under parametric constraints. *IEEE Signal Process. Lett.*, 5(7):177–179.

[Sturm, 1999] Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653.

[Sung et al., 2005] Sung, Y., Tong, L., and Poor, H. V. (2005). A large deviations approach to sensor scheduling for detection of correlated random fields. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, pages iii–649. IEEE.

[Tibshirani et al., 2005] Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused LASSO. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.

[Tichavsky et al., 1998] Tichavsky, P., Muravchik, C., and Nehorai, A. (1998). Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Trans. Signal Process.*, 46(5):1386–1396.

[Vaidyanathan, 2001] Vaidyanathan, P. (2001). Generalizations of the sampling theorem: Seven decades after Nyquist. *IEEE Trans. Circuits Syst. I: Fundamental Theory and Applications*, 48(9):1094–1109.

[Van Trees, 2004] Van Trees, H. L. (2004). *Detection, Estimation, and Modulation Theory, Optimum Array Processing.* John Wiley & Sons.

[Vaswani, 2008] Vaswani, N. (2008). Kalman filtered compressed sensing. In *Proc. of 15th IEEE International Conference on Image Processing, 2008. ICIP 2008.*, pages 893–896. IEEE.

[Viklands, 2008] Viklands, T. (2008). *Algorithms for the Weighted Orthogonal Procrustes problem and Other Least Squares Problems*. Ph.D. dissertation, Dep. Comput. Sci., Umea Univ., Umea, Sweden.

[Wahba, 1965] Wahba, G. (1965). A least squares estimate of satellite attitude. *SIAM review*, 7(3):409–409.

[Wang et al., 2009] Wang, T., Leus, G., and Huang, L. (2009). Ranging energy optimization for robust sensor positioning based on semidefinite programming. *IEEE Trans. Signal Process.*, 57(12):4777 –4787.

[Wang et al., 2011] Wang, Y., Ma, X., and Leus, G. (2011). Robust time-based localization for asynchronous networks. *IEEE Trans. Signal Process.*, 59(9):4397 –4410.

[Warnell et al., 2012] Warnell, G., Reddy, D., and Chellappa, R. (2012). Adaptive rate compressive sensing for background subtraction. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012*, pages 1477–1480. IEEE.

[Wu et al., 2011] Wu, Y.-C., Chaudhari, Q., and Serpedin, E. (2011). Clock synchronization of wireless sensor networks. *IEEE Signal Process. Mag.*, 28(1):124 –138.

[Xiao and Boyd, 2004] Xiao, L. and Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78.

[Yao et al., 1993] Yao, L., Sethares, W., and Kammer, D. (1993). Sensor placement for on-orbit modal identification via a genetic algorithm. *AIAA journal*, 31(10):1922–1928.

[Yu and Varshney, 1997] Yu, C.-T. and Varshney, P. K. (1997). Sampling design for Gaussian detection problems. *IEEE Trans. Signal Process.*, 45(9):2328–2337.

[Yuille and Rangarajan, 2003] Yuille, A. L. and Rangarajan, A. (2003). The concave-convex procedure. *Neural computation*, 15(4):915–936.

[Zhan et al., 2010] Zhan, P., Casbeer, D., and Swindlehurst, A. (2010). Adaptive mobile sensor positioning for multi-static target tracking. *IEEE Trans. Aerosp. Electron. Syst.*, 46(1):120–132.

[Zhu et al., 2011] Zhu, H., Leus, G., and Giannakis, G. (2011). Sparsity-cognizant total least-squares for perturbed compressive sampling. *IEEE Trans. Signal Process.*, 59(5):2002–2016.

[Zuo et al., 2007] Zuo, L., Niu, R., and Varshney, P. (2007). Posterior CRLB based sensor selection for target tracking in sensor networks. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages II–1041–II–1044.

Summary

Sparse Sensing for Statistical Inference

Theory, Algorithm, and Applications

In todays society we are flooded with massive volumes of data of the order of a billion gigabytes on a daily basis from pervasive sensors. It is becoming increasingly challenging to locally store and transport the acquired data to a central location for signal/data processing (i.e., for inference). Consequently, most of the data is discarded blindly, causing serious performance loss. It is evident that there is an urgent need for developing unconventional sensing mechanisms to extract as much information as possible yet collecting fewer data. Thus, reducing the costs of sensing as well as the related memory and bandwidth requirements.

The first aim of this thesis is to develop theory and algorithms for data reduction. We develop a data reduction tool called sparse sensing, which consists of a deterministic and structured sensing function (guided by a sparse vector) that is optimally designed to achieve a desired inference performance with the reduced number of data samples. The first part of this thesis is dedicated to the development of sparse sensing models and convex programs to efficiently design sparse sensing functions.

Sparse sensing offers a number of advantages over compressed sensing (a state-of-the-art data reduction method for sparse signal recovery). One of the major differences is that in sparse sensing the underlying signals need not be sparse. This allows us to consider general signal processing tasks (not just sparse signal recovery) under the proposed sparse sensing framework. Specifically, we focus on fundamental statistical inference tasks, like estimation, filtering, and detection. In essence, we present topics that transform classical (e.g., random or uniform) sensing methods to low-cost data acquisition mechanisms tailored for specific inference tasks. The developed framework can be applied to sensor selection, sensor placement, or sensor scheduling, for example.

In the second part of this thesis, we focus on some applications related to distributed sampling using sensor networks. Recent advances in wireless sensor technology have enabled the usage of sensors to connect almost everything as a network. Sensor networks can be used a spatial sampling device, that is, to faithfully represent distributed signals (e.g., a spatially varying phenomenon such as the temperature field). On top of that, the distributed signals can exist in space and time, where the temporal sampling is achieved using the sensor's analog-to-digital converters, for example. Each sensor has an independent sample clock, and its stability essentially determines the alignment of the temporal sampling grid across the sensors. Due to imperfection in the oscillators, the sample clocks drift from each other, resulting in the misalignment of the temporal sampling grids. To overcome this issue, we devise a mechanism to distribute the sample clock wirelessly. Specifically, we perform wireless clock synchronization based on the time-of-flight measurements of broadcast messages. In addition, clock synchronization also plays a central role in other time-based sensor network applications such as localization. Localization is increasingly gaining popularity in many applications, especially for monitoring environments beyond human reach, e.g., using robots or drones with several sensor units mounted on it. Consequently we now have to localize more than one sensor or even localize the whole sensing platform. Therefore, we extend the classical localization paradigm to localize a (rigid) sensing platform by exploiting the knowledge of the sensor placement on the platform. In particular, we develop algorithms for rigid body localization, i.e., for estimating the position and orientation of the rigid platform using distance measurements.

Given the central role of sensing and sensor networks, the results presented in this thesis impacts a wide range of applications.

Sundeep Prabhakar Chepuri

Biography

S.P. (Sundeep Prabhakar) Chepuri was born in Bangalore, India in 1986. He received the Bachelors in Engineering degree (*with distinction*) from the PES Institute of Technology, Bangalore, India, in 2007, and the Master of Science degree (*cum laude*) from the Delft University of Technology, The Netherlands, in 2011. He is currently with the Circuits and Systems group at the Faculty of Electrical Engineering, Mathematics and Computer Science of the Delft University of Technology, The Netherlands. During his PhD thesis work, he was employed on the STW FASTCOM project and later on an NWO/STW VICI project. He has held positions at Robert Bosch, India, during 2007-2009, and Holst Centre/imec-nl, The Netherlands, during 2010-2011. He has received the Best Student Paper Award for his publication at ICASSP 2015 conference in Australia (ICASSP is the largest and a very important conference in signal processing receiving around 3000 submissions every year). His general research interest lies in the field of mathematical signal processing, statistical inference, sensor networks, and wireless communications.