SPARSITY-AWARE WIRELESS NETWORKS: LOCALIZATION AND SENSOR SELECTION

HADI JAMALI-RAD

SPARSITY-AWARE WIRELESS NETWORKS: LOCALIZATION AND SENSOR SELECTION

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 van Promoties,
in het openbaar te verdedigen op maandag 22 december 2014 om 15.00 uur

door

HADI JAMALI-RAD

Master of Science in Electrical Engineering
Iran University of Science and Technology

geboren te Lahijan, Iran.

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. ir. G. J. T. Leus

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. ir. G. J. T. Leus	Delft University of Technology, promotor
Prof. dr. ir. M. Verhaegen	Delft University of Technology
Prof. dr. ir. C. Witteveen	Delft University of Technology
Prof. dr. M. Viberg	Chalmers University of Technology
Prof. dr. ir M. Moonen	University of Leuven
Prof. dr. X. Ma	Georgia Institute of Technology
Dr. X. Campman	Shell Global Solutions International B. V.
Prof. dr. ir. A.-J. van der Veen	Delft University of Technology (reserve lid)

The work presented in this thesis was financially supported by STW-NWO.

Printed by: Gildeprint Drukkerijen - The Netherlands.

ISBN: 978-94-6108-855-0

To my beloved family,
my sister Maryam,
and my parents,
Mahrokh and Bahman.

SUMMARY

Wireless networks have revolutionized nowadays world by providing real-time cost-efficient service and connectivity. Even such an unprecedented level of service could not fulfill the insatiable desire of the modern world for more advanced technologies. As a result, a great deal of attention has been directed towards (mobile) wireless sensor networks (WSNs) which are comprised of considerably cheap nodes that can cooperate to perform complex tasks in a distributed fashion in extremely harsh environments. Unique features of wireless environments, added complexity owing to mobility, distributed nature of the network setup, and tight performance and energy constraints, pose a challenge for researchers to devise systems which strike a proper balance between performance and resource utilization.

We study some of the fundamental challenges of wireless (sensor) networks associated with resource efficiency, scalability, and location-awareness. The pivotal point which distinguishes our studies from existing literature is employing the concept of sparse reconstruction and compressive sensing (CS) in our problem formulation and system design. We explore sparse structures embedded within the models we deal with and try to benefit from the undersampling offered by incorporating sparsity and thereby developing sparsity-aware system-level solutions. We prove that looking at these challenges from our perspective not only guarantees an expected cost efficiency due to taking less measurements, but also if properly designed, can promise an acceptable accuracy.

We start by looking at the location-awareness of mobile wireless networks as a key enabler for meaningful data extraction. Given the elegance and simplicity of multidimensional scaling (MDS) for network localization, we combine subspace perturbation expansion (SPE) with classical MDS and derive a model-independent dynamic version of MDS which can be employed to track a network of mobile nodes using only pairwise distance measurements. We further extend our low-complexity dynamic MDS paradigm in two different ways (adaptive inner-iterations and geometrical reconstruction) to be able to operate in partially connected networks where some of these pairwise distances are missing. We also study a model-dependent

case where the movement process of the nodes is known. In such a case, we propose to linearize the nonlinear set of measurements w.r.t. the location of the anchor nodes in the network and track the mobile nodes using a Kalman filter (KF) instead of a suboptimal extended KF (EKF). In both directions, we illustrate promising results confirming the efficiency of our proposed ideas.

We then study a related multi-source localization problem where some of the nodes in the network are considered to be pure signal emitters or so-called sources. The important feature of such sources, which is essentially the origin of the complexity of this problem, is the fact that these sources cannot be distinguished based on the signals they transmit. This introduces a complex assignment problem to decompose the received signals (typically the summation of the transmitted signals) and relate them to their respective sources to be able to localize them. We propose innovative ideas to solve this problem using time-difference-of-arrival (TDOA) and received-signal-strength (RSS) measurements. The general approach we propose is based on discretizing the area under consideration and performing fingerprinting. We then exploit the spatial sparsity of the sources in the discretized domain and propose sparsity-aware solutions which can offer a superb performance in terms of the number of identifiable sources. We further extend our TDOA-based approach to be able to localize off-grid sources with an acceptable accuracy. Both for the RSS-based approach in indoor environments as well as for an underground microseismic monitoring scenario, we further extend our proposed ideas to operate in a fully blind fashion w.r.t. the statistics of the source signals. We present extensive simulation results to corroborate our claims.

Finally, we turn our attention towards the sensor selection problem in WSNs in order to satisfy a specific estimation performance metric. In line with the main flavor of this thesis, this time, we explore the sparsity of the selected sensors among the total number of sensors in the network and propose sparsity-aware solutions for both cases where the noise experienced by the sensors is uncorrelated or correlated. To circumvent the limitations of a centralized approach for large-scale WSNs, we extend both algorithms to distributed ones where each sensor has to rely only on local information and has to decide whether it should contribute in the estimation task or not. Our detailed convergence proofs, quantified computational and communication costs, as well as our simulation results all confirm the applicability and efficiency of our newly introduced sensor selection paradigm.

Keywords: Mobile network localization, multi-source localization, sensor selection, sparse reconstruction, distributed optimization.

ACKNOWLEDGMENTS

This thesis marks the end of a journey! Four years full of memories have passed in a blink of an eye. I would like to express my gratitude to all who have supported me throughout this journey, and have taken parts in these unforgettable memories.

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Geert Leus. Thank you Geert for your trust in the first place to accept me as a PhD student. Also, thanks a lot for your wise guidance at the right moment; you are a brilliant scholar to learn from. I am also indebted to Geert for his support to go for internship and academic visits; he is a generous gentleman with a nice sense of humor. To be perfectly honest, Geert is the only reason why I decided to come to Delft, and I think it was a great decision.

I would like also to thank Prof. Alle-Jan van der Veen, the head of the Circuits and Systems (CAS) group, as well as all the past and current members of the group for creating such a nice environment for research, for wonderful discussions on variety of subjects during coffee breaks, and for our memorable times in group outings and conference excursions.

I do extend my appreciation to Prof. Xiaoli Ma at Georgia Institute of Technology (Georgia Tech) for hosting me for about four months in the center for signal and information processing (CSIP). It was a great pleasure for me to be in such a vibrant environment full of great minds. I am grateful to Toon van Waterschoot for arranging my short visit to Prof. Marc Moonen's group at Katholike Universiteit Leuven (KU Leuven). Toon, I enjoyed a lot our fruitful collaborations as well as our interesting discussions on music. It was an honor for me to stay in Marc's group; his scientific level and academic style are remarkable.

My appreciation goes to Prof. Bahman Abolhassani my past supervisor at Iran University of Science and Technology (IUST), for his patience and non-stop support, and for teaching me the first lessons on how to write an academic article, lessons which I have carried along till this very moment. I also thank my committee members for taking the time to read the draft of this thesis.

I was very fortunate to make quite a few friends in the Netherlands. First of all, I should thank Saleh Heidary for always being there to support me. Saleh, I am very happy that our paths have crossed in Delft and I could make such a good friend. It is always good to have your old friends around you, and coming to Delft was a lucky reunion with Mojtaba Shakeri, Amin Karamnejad and Mohammad-Hassan Javadzadeh. A special thanks goes to Andrea Simonetto, a nice colleague and a friend. I treasure every second of our intense discussions on optimization. I also enjoyed our chats on cool Italian gestures; always made me laugh out loud! *Andrea, vado pazzo per il tuo stile italiano!* The same goes to Jorge Martinez Castañeda and Rocio Arroyo-Valles for their help and support. *Además, muchas gracias mis amigos por enseñarme lo elemental de vuestro fantástico idioma, el espanõl!* I also thank my friends Pejman Pourvahedi, Hossein Khalilnejad, Sina Naddaf, Babak Raji, Amin Fatemi, Mehdi Mousivand, Dara Ghasimi, Vahid Arbabi, Shahab Pourtalebi, Amir Saeidi, Arash Noroozi, Bahman Yousefzadeh, Milad Sardarabadi and Sina Maleki for the memorable time we spent together.

I also take the opportunity to thank my friends at Georgia Tech (Saam Yasseri, Majid Sodagar, and Ahmad Beirami) who helped me a lot to settle there so quickly. Ahmad, I enjoyed our technical discussions and the unforgettable road trip to Washington, D.C. My gratitude goes to my dear friends at KU Leuven (Reza Sahraeian and Amin Hassani) for helping me a lot during my stay in Leuven.

I am grateful to my old friends, now residing in Iran or other countries. My gratitude goes to Ehsan Behrangi, my oldest and best friend from childhood (and a true genius!) who has never stopped supporting me, and giving me positive energy. Ehsan, I would not hesitate to call you brother. My sincere appreciation goes to my dear friends Hadi Meshgi, Payam Ghaebi-Panah, Mohammad Abdizadeh, Milad Khoshnegar, Kiarash Gharibdoust, Arash Amini and Ali Keymasi, as well as my family members Reza Darvishi, Ali Darvishi, and Meisam Jamali.

Finally, my foremost gratitude and recognition goes to my family for their unwavering support. Let me start with my sister Maryam, thanks a lot for showing me how to love and for always being there for me. I am also indebted to my brother-in-law, Omid, for being such a good friend and a true son for my family when I am so far away. My deepest appreciation goes to my parents, Mahrokh and Bahman, for their unconditional and insatiable dedication and love.

Delft, August 2014
Hadi Jamali-Rad

CONTENTS

PART

I

PROLOGUE

If you can't explain it simply, you don't understand it well enough.

ALBERT EINSTEIN

1 INTRODUCTION

This thesis is concerned with exploring sparse structures within some fundamental problems in wireless (sensor) networks such as network localization, multi-source localization, and sensor selection. Upon exploring such sparse structures, we incorporate this prior knowledge into our modeling and propose *sparsity-aware* solutions. We prove that looking at these problems from our newly introduced angle results in a substantial performance gain. We start this chapter by elaborating on the overall motivation of this thesis followed by introducing the concept of location awareness and distributed estimation in sensor networks. We then provide an outline of the presented work along with highlighting our main contributions.

1.1 Motivation

Nowadays world is blessed with an unprecedented wireless connectivity realized by a variety of wireless networks ranging from cellular connections to satellite broadcasting. This dramatic growth can be attributed to both technological advances as well as a tremendous demand. We envisage a future of ubiquitous wireless connectivity all around the globe. The unique nature of wireless networks (unpredictable wireless channels and complex network analysis) has made their design a challenging problem which has received an upsurge of attention over the past decades. In line with the development of wireless technologies, advances in miniaturization and integration of sensing and computation has made the emergence of wireless sensor networks (WSNs) possible.

WSNs consist of a large number of tiny sensor nodes (as shown in Fig. 1.1) with limited computational and communication capabilities. Nonetheless, when properly networked and programmed they can cooperate to perform advanced signal processing tasks with significant versatility and robustness. This potential has made them an attractive but at the same time cost-efficient technology for a wide variety of applications, such as remote sensing, environmental and wildlife monitoring, and asset tracking, to name a few [1]. WSNs are envisioned to be the building blocks of tomorrow's proactive computing world (as shown in Fig. 1.2).

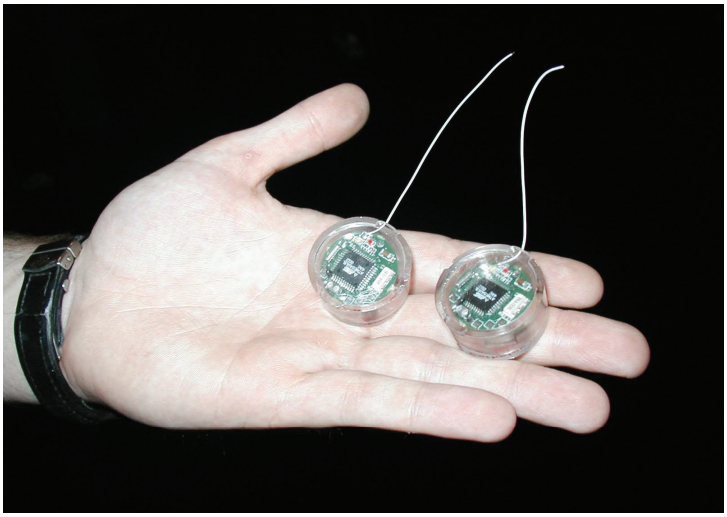


Figure 1.1: UC Berkeley sensor dots (Courtesy of UC Berkeley).

Figure 1.2: Applications of WSNs in our everyday life.

As research in wireless (sensor) networks proliferates, the problem of location awareness within such networks becomes significantly important, since it is a key feature enabling many other applications. Location awareness is for instance crucial for meaningful data inference when part of or the whole infrastructure of the network is mobile. Even though this mobility is a privilege raising the level of flexibility in the network, at the same time it introduces complications that have to be taken care of. Examples of such are designing proper dynamic localization, in-

formation retrieval, and data management algorithms as well as bandwidth-limited communication protocols and power-limited mobile nodes. Even in static but large scale WSNs the latter issue, i.e., designing resource-efficient and scalable (possibly distributed) paradigms has to be well investigated and then properly addressed. An example of such a scalable resource-efficient WSN management algorithm is the design of a distributed sensor selection algorithm to determine a minimum number of sensors to be activated within the network to accomplish a given task with a desired level of accuracy.

In this thesis, we look at the aforementioned challenges from a different angle. Particularly, we try to explore embedded sparse structures and propose sparsity-aware solutions.

1.2 Location Awareness in Wireless Networks

The paradigm of context-aware computing has received an upsurge of attention in the past few decades. Context-aware computing systems aim to adapt themselves to the variations in their surrounding environment autonomously, and also customize their behavior accordingly [2]. This paradigm puts a significant step towards the vision of ubiquitous computing [3]. Location-aware computing is an important subset of the context-aware computing paradigm. Nowadays, location awareness has become an indispensable feature of wireless networks. The overwhelming reason is that the buck of information collected by the components (for instance, sensor nodes) of a wireless network is in practice meaningless if one cannot attach them to location information. Examples of such scenarios are diverse among which we can point out environmental monitoring [4] and asset tracking [5]. Location information also facilitates the process of establishing a connection, routing and communication among adjacent (sensor) nodes, especially when dealing with a large scale network.

The process of acquiring location information is called *localization*. This process normally has two stages; the initial stage is a *measurement stage* in which the measurements are conducted, and the next stage is an *estimation stage* in which the locations are estimated based on the measurements [6, 7]. Common measurement types usually employed for localization purposes are received signal strength (RSS), time-of-arrival/flight (TOA/F), time-difference-of-arrival (TDOA), angle-of-arrival (AOA), or a combination of those in order to attain a better accuracy [6]. Measurements are normally corrupted due to noise, multipath, blockage, interference, and other environmental effects [8]. That is why the measured information will not be so accurate which leads to coarse location estimations. In practice,

TOA, AOA and TDOA measurements are more expensive to acquire and provide finer resolutions compared to RSS which usually comes for free as a built-in feature found in many wireless devices.

From a broader perspective, the problem of localization in wireless networks can be looked at from a few different angles. Examples are, for instance:

- ◇ *Range-based vs. range-free*: Range-based algorithms rely on range-related measurements such as the ones mentioned earlier (RSS, TDOA, TOA/F, and AOA) to estimate the location whereas the range-free algorithms [9, 10] work with the connectivity (or sometimes called proximity) information between the neighboring nodes. Generally speaking, range-free algorithms are rough in terms of accuracy compared to range-based algorithms. In this thesis, we consider range-based localization.

- ◇ *Centralized vs. distributed*: In a centralized algorithm, measurement information collected by nodes is sent to a central processing unit (CPU) or sometimes called fusion center (FC) to be analyzed and exploited for location estimation. On the other hand, in distributed localization algorithms, the nodes only make use of local information (measured themselves and/or provided by their neighboring nodes) to self-localize themselves. In comparison, distributed algorithms are more appropriate for large-scale networks. In this thesis, we mainly focus on centralized localization algorithms, even though as a special case, we also consider the case of *partially connected* networks where part of the measurements can be missing.

- ◇ *Indoor vs. outdoor*: Traditional outdoor localization systems, such as the global positioning system (GPS), are not necessarily a proper choice for network localization, especially when the nodes of the network are sensors with limited computational and communication capabilities. This issue calls for the design of proper resource-constrained localization algorithms. On top of that, traditional outdoor algorithms do not provide acceptable accuracy for indoor scenarios where a lot of reflection and multipath effects exist. For such challenging environments, ultrawide bandwidth (UWB) transmission technology is a promising alternative since it benefits from a superior signal penetration through obstacles and offers a finer delay resolution [11]. The complications of the wireless channel in indoor environments has also motivated another category of localization algorithms based on wireless fingerprinting [12, 13]. Fingerprinting consists of the the following two phases. In a training phase, a dictionary (fingerprinting map) is constructed by discretizing the area of interest into a mesh of grid points and by recording

the fingerprints from every single grid point. Next, in the real-time phase, the measurements are compared with the atoms of the dictionary (usually its columns) to determine where the source(s) is (are) located [13].

- ◇ *Static vs. mobile/dynamic*: The nodes to be localized can be static or mobile. Moreover, there exist scenarios in which the whole network is mobile. In this thesis, we consider both mobile and static localization algorithms.

Another fundamental angle from which we would like to distinguish between localization paradigms, is whether the nodes/components in the network can be uniquely identified based on the signals they transmit or not. This property plays a pivotal role in this thesis, because if the nodes cannot be distinguished based on the signals they transmit, many of the existing network algorithms will not operate anymore. This motivates us to classify the localization paradigms into "network" and "source" localization scenarios, as discussed in the following.

1.3 Network Localization

The first scenario we consider is network localization. Such a network is typically comprised of a few nodes with known location, sometimes called *anchors*, and some other nodes with unknown locations [14–17]. An essential assumption here is that all the nodes can be uniquely identified based on their transmitted signals. In special cases such as low-cost sensors deployed in harsh environments, it is possible that no anchors exist. Such a network localization paradigm is commonly addressed as *anchorless* [18]. In anchoreless scenarios, the ultimate goal is to find only the relative locations or the configuration of the network up to an ambiguity. We study both anchored and anchorless scenarios in this thesis.

We can divide the studies in network localization into two categories based on the notion of *cooperation*. From this viewpoint, the network localization algorithms are classified as:

- ◇ *Non-cooperative network localization*: This is the case when the unknown-location nodes do not cooperate with each other. An example could for instance be a wireless local area network (WLAN) setup where the access points (APs) act as the anchors and mobile stations (MSs) either try to self-localize themselves, or they will be localized at a FC in the backbone network. Typically, in such a case, long-range transmissions from MSs to APs are feasible.

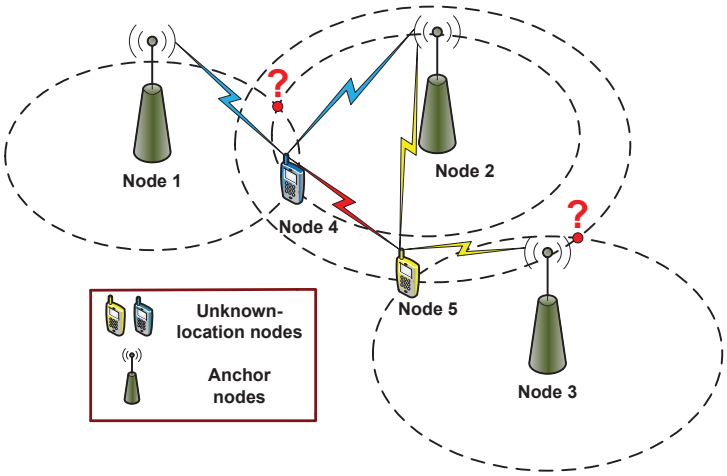


Figure 1.3: Network localization: Cooperative vs. non-cooperative.

- ◇ *Cooperative network localization* [6, 7, 16, 18–21]: Here, the cooperation among unknown-location nodes eliminates the need for them to be within the communication range of multiple anchors, and therefore, a high anchor density or long-range transmissions are no longer necessary. Moreover, the fact that the unknown-location nodes can receive information from nodes of the same type as well as from anchors within their communication range results in a superior performance in terms of accuracy and coverage for cooperative localization algorithms. On the other hand, this calls for a more elaborate and thus more expensive design of the nodes circuitry and their communication protocols.

Fig. 1.3 illustrates the advantage of cooperative network localization compared to non-cooperative in a simple setup. As can be seen, the unknown-location nodes 4 and 5 each can communicate only with two anchor nodes out of three, and thus only based on range estimations they cannot determine their locations. However, when they cooperate (the red link), they can uniquely determine their own locations and eliminate other ambiguous locations (red circles with "?" on top). In this thesis, we mainly focus on cooperative network localization.

1.4 Multi-Source Localization

The other major scenario that we consider in this thesis is the problem of multi-source localization. Notably, single-source localization, can simply be considered

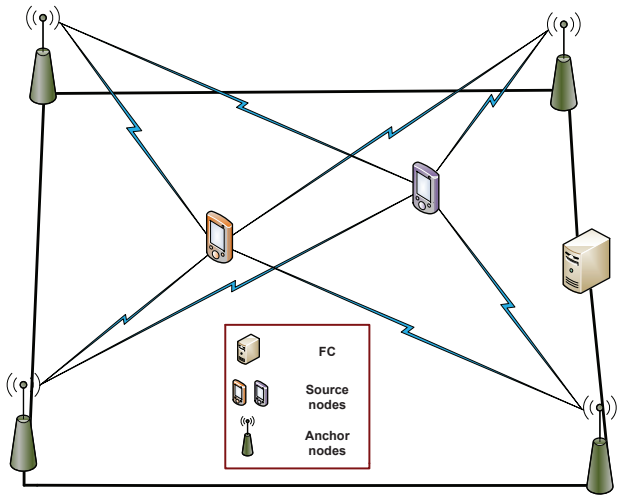


Figure 1.4: Multi-source localization scenario; case of heterogeneous sources.

as a case of network localization when a group of anchor nodes try to localize an unknown-location source (sometimes called emitter) node based on different measurement types available within a covered area [14, 22–25]. In this context, given the fact that source nodes are not sensor nodes anymore as they can be in Subsection 1.3, we prefer to divide the nodes into source(s) and sensors. Now, the principal question is what happens if we have multiple of such signal sources/emitters? The answer is actually up to the pivotal point we mentioned earlier. If the sources can be distinguished based on any unique signature (signal type they transmit, frequency band or time slot they occupy, etc.) the problem transforms back to the case of network localization, for instance by decomposing the problem into multiple single-source localization problems. On the other hand, if the sources cannot be distinguished based on the signal they transmit, for many practical signals such as electromagnetic and acoustic signals, the sensor nodes will receive a summation of signals transmitted by the sources. More importantly, they cannot decompose the received sum to the contribution of each source. This basically complicates the localization process by imposing an embedded assignment problem to determine which part of the received sum belongs to which source. A typical multi-source scenario is shown is Fig. 1.4 where the two source signals are of the same type, and thus are indistinguishable.

Similar to the case of network localization, the notion of cooperation (in a different sense from Subsection 1.3) splits the studies into the following two directions.

- ◇ *Cooperative sources* [26–29]: In this context, these are sources from which we have some information about the nature of the transmitted signals such as their statistics. Even though this information is not sufficient to decompose the sum of the received signals and assign them to their sources.

- ◇ *Non-cooperative sources* [30, 31]: The case of multiple non-cooperative sources is a formidable problem because in principle, nothing is known from the sources or transmitted signals. This necessitates coming up with solutions which are *blind* to the source signal information.

We study both cooperative and non-cooperative cases for multi-source localization.

1.5 Distributed Sensor Selection

WSNs are often large-scale self-organized networks with no pre-established infrastructure or a topology that can dynamically alter. In order to deduce meaningful and accurate information from such a network, raw data (signals) collected by different sensors should somehow be combined and processed, which is referred to as *data fusion* [32]. This can either happen in a centralized fashion by broadcasting all this data to a FC or it can be done in a distributed fashion. The centralized approach calls for a high communication bandwidth and transmission power, which is usually lacking due to limited capabilities of sensor nodes [1]. Moreover, the FC is potentially a single point of failure in a network, i.e., if the FC is compromised or jammed, the whole network fails to operate. A distributed approach eliminates the need for an FC. That is why often a distributed approach is preferred for large-scale WSNs owing to its scalability in term of communication and computational costs.

1.5.1 Distributed Estimation over WSNs

To be more specific, distributed processing means that instead of transmitting all the data collected by the sensors to an FC in order to make a decision or accomplish the final goal of the network, each sensor should rely only on local information received by itself and the sensors in its vicinity. This presents a formidable challenge to design appropriate distributed signal processing algorithms at local sensors to reduce data transmissions [32]. On the other hand, relying only on the information received by a single sensor (or a small group of them) might not necessarily lead to the overall precision required by the network. Thus, appropriate information sharing and collaborative processing algorithms should also be put in place to ensure a reliable inference [32, 33]. In a nutshell, distributed processing makes large-scale

sensor networking possible by striking a proper trade-off between performance and resource utilization.

It is worth highlighting that designing distributed signal processing algorithms is a challenging task which can sometimes lead to suboptimal solutions with inferior performance compared to the case of centralized algorithms. In general, given the fact that all the required information is present in one place (FC), the centralized algorithms are expected to show superior performance and hence, they can be used as performance metrics for the assessment of the distributed algorithms.

1.5.2 Sensor Selection

In this thesis, we particularly consider a *parameter estimation* problem over WSNs where the network is supposed to estimate a vector of parameters with a fixed length using measurements given by

$$y_i = f(\mathbf{x}) + \eta_i, \tag{1.1}$$

where subscript i indicates being associated with the i-th sensor, y_i and η_i are respectively the measurement and additive noise at the i-th sensor, $\mathbf{x}$ is the parameter vector of interest, and $f(.)$ can be a linear or nonlinear function representing the way the y_i's are related to $\mathbf{x}$. Notably, in typical parameter estimation problems, the parameter of interest is either fixed or varies slowly over time (like temperature, humidity, etc.) which permits applying iterative solutions [34].

Within this parameter estimation framework which can in turn be centralized or distributed, an important question is that if a parameter is supposed to be estimated within a medium-to-large-scale sensor network, do we need all the sensors to be activated to satisfy our desired estimation performance? Maybe it is enough to activate only some of the sensors to do the job. If yes, which sensors should be selected? This reminds us of the classical problem of sensor selection [35–37] which is about selecting k sensors out of m to satisfy a network estimation performance.

In this thesis, we formulate the problem in a rather general framework and solve the following closely related selection problem. We are interested in selecting the minimum number of sensors *a priori* based on our knowledge of $f(.)$ so that a specific performance metric related to the estimation of $\mathbf{x}$ is satisfied. From this perspective, our look at the problem is somehow closer to what is called robust sensing [38] or informative-sensor identification [39]. Moreover, given the aforementioned advantages of distributed implementation (cost and robustness), a major part of our study is devoted to developing a distributed sensor selection paradigm

wherein based on local information each sensor decides itself about its status of being active or inactive.

1.6 Thesis Outline and Contributions

The fundamental challenges we discussed earlier on location-awareness and resource efficiency in wireless (sensor) networks are the principal motives behind our contributions. In this thesis, our primary focus is on exploring embedded sparse structures within the network localization, source localization and distributed sensor selection problems. In other words, this thesis takes a step forward along the path to sparsity-aware wireless (sensor) networks. We illustrate that exploring such sparse structures, reformulating the problems at hand by incorporating the prior knowledge of sparsity, and solving them using proper sparsity-aware algorithms yields a significant performance gain over the existing algorithms which ignore this information.

This thesis consists of five main parts. In short, Part I contains this introduction in Chapter 1, and some preliminaries in Chapter 2. In the preliminaries, we briefly discuss multidimensional scaling (MDS), subspace perturbation expansion, compressive sensing, and the alternating direction method of multipliers (ADMM) in order to provide the reader with the basic mathematical and signal processing tools we employ in this thesis. The next three parts (Parts II-IV) present our main contributions which are concisely specified here and further elaborated on later in this section. More specifically, Part II, which includes Chapters 3-4, is devoted to our contributions to mobile network localization. Part III presents our contributions to sparsity-aware multi-source localization using TDOA (in Chapter 5) and RSS (in Chapter 6) for wireless channels, as well as for a different case study related to microseismic signals (in Chapter 7). In Part IV, which includes Chapters 8-9, we tackle the problem of sparsity-aware sensor selection in a distributed fashion. Finally, Part V contains Chapter 10 on conclusions and future research recommendations. In the following, we further elaborate on our main contributions in each part.

1.6.1 Contributions on Mobile Network Localization

The first problem we focus on is to find a solution for cooperative localization of a dynamic network of mobile sensor nodes with low computational complexity. This problem becomes very challenging for anchorless networks where there is no pre-existing infrastructure to rely on. Knowing the elegance of classical multidimensional scaling (MDS) for static anchorless localization, we have generalized it

to a dynamic MDS paradigm to handle a mobile network. In simple terms, classical MDS accepts a matrix of pairwise distances between all the nodes as input, and by the help of signal subspace analysis returns the configuration of those nodes.

1. Dynamic MDS

The idea is that under some conditions (detailed in Chapter 3) for small time intervals the next configuration of a network of nodes can be modeled as a perturbation of its previous configuration. Besides, subspace perturbation expansion (SPE) tells us how subspaces of a matrix alter by applying a perturbation. This triggered the idea of combining MDS and SPE to devise a dynamic MDS that can keep track of the configuration of a mobile network with low complexity. We also present a similar approach using a combination of MDS and orthogonal iterations to track the invariant subspace of our measurement matrix and thus to track the network of mobile nodes. The trick is to do only a single orthogonal iteration per step and to use the subspace estimated from the previous *time step* as the initial guess. This way there is no need for a large number of iterations and we avoid divergence.

We illustrate in Chapter 3 that compared to recently proposed competitors based on the extended and unscented Kalman filter (EKF and UKF), the proposed algorithms have a considerably lower computational complexity. Furthermore, model-independence, scalability as well as an acceptable accuracy make our proposed approach a good choice for practical mobile network localization.

MDS is known to be a good choice for a fully connected network of nodes. However, in practice, not all the nodes in a sensor network can communicate with each other. This motivates us to extend our contribution in this context to the case where some of the communication links are missing which is sometimes called a partially connected network. We have tackled this problem in two different but conceptually related ways.

2. Dynamic MDS for a partially connected network

The first idea, presented in Chapter 3, is to include an inner iteration (by repeating estimation-modification-estimation) per time step in our proposed dynamic MDS to account for the missing links. The other idea, presented in Chapter 4, explores the geometric relationships of the pairwise distances of the sensor nodes and in an *intelligent* local-to-global fashion reconstructs the missing links as good as possible or uses the previous estimates for the links that could not be reconstructed. This way we fill in the missing links in each step of our dynamic process and then use our previously proposed dynamic MDS idea.

We show that in terms of computational complexity as well as estimation performance our proposed algorithms for partially connected networks are excellent choices and they can handle situations where the network is moderately connected.

The earlier contributions for network localization are *model-independent* approaches. We also study the problem from a *model-based* perspective where statistical properties of measurement and process models are available. In the model-based context, most of the existing works employ Kalman filter (KF)-based approaches. However, since the distance measurements are by nature nonlinear, they have to use the EKF or other similar approaches based on computing partial derivatives of the measurements. This motivates us to try to look at the problem in a different way. Notably, this part is not presented in this thesis.

3. Linearize and use KF instead of EKF

The idea is that the knowledge of the location of the anchor nodes helps to linearize the nonlinear distance measurements with respect to (w.r.t.) the location of the unknown nodes. Next, based on this "linearized" measurement model, we can use the KF itself instead of a suboptimal EKF. The downside is that we have to estimate the corresponding unknown measurement noise covariance matrix using an iterative process, which comes at a price.

The simulation results illustrate that the proposed algorithm (only within a few iterations to account for the new covariance matrix estimation) attains the posterior Cramér-Rao bound (PCRB) of mobile location estimation and clearly outperforms related anchorless and anchored mobile localization algorithms.

Our contributions in this context are published in the following papers. The first two are respectively contained in Chapters 3-4, and the rest are omitted for the sake of brevity.

[J1] H. Jamali-Rad and G. Leus, "Dynamic Multidimensional Scaling for Low-Complexity Mobile Network Tracking," *IEEE Trans. on Sig. Proc. (TSP)*, vol. 60, no. 8, pp. 4485-4491, Aug. 2012.

[C4] H. Jamali-Rad, H. Ramezani, and G. Leus, "Cooperative Localization in Partially Connected Mobile Wireless Sensor Networks Using Geometric Link Reconstruction," in *Proc. of IEEE Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, pp. 2633- 2636, Japan, Mar. 2012.

[C3] H. Jamali-Rad, T. van Waterschoot, and G. Leus, "Cooperative Localization Using Efficient Kalman Filtering for Mobile Wireless Sensor Networks," in *Proc. of European Sig. Proc. Conf. (EUSIPCO)*, pp. 1984-1988, Spain, Aug.-Sep. 2011.

[C2] H. Jamali-Rad, T. van Waterschoot, and G. Leus, "Anchorless Cooperative Localization for Mobile Wireless Sensor Networks" in *Proc. of The Joint WIC/IEEE SP Symp. on Info. Theory and Sig. Proc. in the Benelux (WICSP)*, Belgium, May 2011.

[C1] H. Jamali-Rad, A. Amar, and G. Leus, "Cooperative Mobile Network Localization via Subspace Tracking," in *Proc. of IEEE Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, pp. 2612 - 2615, Czech Republic, May 2011.

Other contributions related to the context of network localization which are not presented in this thesis revolve around localization of a fixed node and tracking of a mobile node in an underwater medium with an isogradient sound speed profile. Knowing the nonlinear dependency of the traveled distance and the required time in such a medium, we try to analytically relate the acoustic wave TOFs between two nodes to their positions. Then, we respectively adopt Gauss-Newton and EKF for localization and tracking purposes.

Our results prove the efficiency of our proposed algorithms by showing that we can attain related performance bounds. Our contributions in this context are published in the following papers but they are omitted in this thesis for the sake of brevity.

[J2] H. Ramezani, **H. Jamali-Rad**, and G. Leus, "Target Localization and Tracking for an Isogradient Sound Speed Profile," *IEEE Trans. on Sig. Proc. (TSP)*, vol. 61, no. 6, Mar. 2013.

[C5] H. Ramezani, **H. Jamali-Rad**, and G. Leus, "Localization and Tracking of a Mobile Target for an Isogradient Sound Speed Profile," in *Proc. of IEEE Intl. Conf. on Communications (ICC)*, pp. 3654 - 3658, Canada, Jun. 2012.

1.6.2 Contributions on Multi-Source Localization

Studying cooperative network localization problems, triggers thinking about the case where the unknown-location nodes are simply sources from which not much information is available. As we discussed earlier, we define this as multi-source localization. Our study on this topic contains a large body of work and casts a big portion of this thesis. One of the major problems we tackle in this context in Chapter 5 is the problem of source localization using TDOA[1] measurements, which turns out to be a non-convex and complex problem due to its hyperbolic nature. The problem becomes highly involved for *multi-source* case where TDOAs should be assigned to their unknown respective sources.

[1] Our ideas here immediately apply to multi-source TOA Localization too.

4. Linearize by fingerprinting, explore the spatial sparsity

> Our idea is to first simplify this problem by linearizing it via a novel TDOA fingerprinting model. Now, at every receiver pair, we are left with an assignment problem to relate the TDOA values to their respective sources. The key idea is to sum the value of the TDOAs and construct a linear set of equations according to the fingerprinting model. We then explore the fact that the sources are sparse in the spatial domain (within the discretized area of interest) and solve the problem using an ℓ_1-norm minimization.

The above approach allows us to solve the problem within a sparse representation framework with a limited performance in terms of the number of identifiable sources. It also holds only for sources which are on the grid points (GPs) we have defined for fingerprinting. However, in real life, the sources are not always on-grid. These issues prompt us to think further and extend our proposed framework. The fact that the values of the TDOAs are known (but it is unclear to which source they belong), is an important observation and origin of the next idea.

5. Not only sum of TDOAs, but also sum of any nonlinear function of TDOAs!

> The idea is that we are not only able to use the values of the TDOAs in our linear sets of equations, but we can also apply any non-linear function to these values and create new linear sets of equations without taking new measurements!

We show that these new sets can be added to the initial set of equations and significantly improve our performance in terms of number of identifiable sources. We even show that under some conditions we can keep on creating new sets of equations until we convert the given underdetermined problem to an overdetermined one that could be solved using classical least squares (LS). We also tackle the problem of off-grid source localization as follows.

6. Find the closest GPs, then solve an LS

> The idea is to look at the problem as a case of grid mismatch where the effect of off-grid sources leads to a perturbation in the fingerprinting map. We then propose a two-step solution in which we first find the closest GPs using the sparse total least squares (STLS) and then having the closest GPs we solve the grid mismatch of each of the sources using a classical LS.

To the best of our knowledge, our work is the first solution to the multi-source TDOA localization problem from a sparsity-aware perspective. Our contributions in this context are published in the following papers. The first one in the list is

contained in Chapter 5, and the other one which is a precursor publication is omitted for the sake of brevity.

[J3] H. Jamali-Rad and G. Leus, "Sparsity-Aware Multi-Source TDOA Localization," *IEEE Trans. on Sig. Proc. (TSP)*, vol. 61, no. 19, Oct. 2013.

[C6] H. Jamali-Rad and G. Leus, "Sparsity-Aware TDOA Localization of Multiple Passive Sources" in *Proc. of IEEE Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, pp. 4021-4025, Canada, May 2013.

Alongside exploiting TDOA measurements, we also inspect the multi-source localization problem using RSS measurements with an emphasis on indoor multipath environments. This time, the complex indoor channel motivates using a fingerprinting approach resembling what we have proposed for TDOA measurements. However, there is an important difference here; RSS measurements from multiple sources automatically sum up at a given receiver. There exists some recent work in literature on a sparsity-aware solution to this problem. The question is whether the existing approaches are efficient in terms of the number of identifiable sources or not. Delving deeper into the problem proves otherwise; the existing sparsity-aware fingerprinting approaches only use the RSS measurements (autocorrelations) at different receivers separately and ignore the potential information present in the cross-correlations among the received signals.

7. Incorporate the cross-correlations, and different time lags

> Our idea is to reformulate this problem to exploit the information present in the cross-correlations by introducing a novel fingerprinting paradigm. Besides, we further enhance this newly proposed approach by incorporating the information present in the other time lags of the autocorrelation and cross-correlation functions rather than only considering the zeroth time lag.

An interesting by-product of the proposed approaches is that under some conditions we could convert the given underdetermined problem to an overdetermined one and efficiently solve it using classical LS. The idea of incorporating the cross-correlations yields a significant performance gain, but this gain comes at a price. In order to be able to make a fingerprint which contains the cross-correlations of the received signals one has to know about the statistical properties of the sources. This somehow prevents us from handling multiple non-cooperative sources. Bear in mind that one does not have to know the statistical properties of the emitted source signals to be able to compute their TDOA or RSS; it is enough to know only the type of the signal (electromagnetic, acoustic, etc.). Therefore, our proposed approach works well for the case of multiple sources with the same statistical prop-

erties, but fails to operate for hetrogenous sources with different statistical properties. This shortcoming urged us to figure out a way so that we can still gain from the received signal cross-correlations in a blind fashion w.r.t. the signal statistical properties.

8. A frequency-domain approach; a blind solution

> The idea is to approach the problem from the frequency domain, design a proper filter bank, explore the common sparsity support of the output of these filters and then propose a proper modified group least absolute shrinkage and selection operator (G-LASSO) estimator.

Our contributions in this context are published in the following papers. The first one is contained in Chapter 6, and the other two which are precursor publications are omitted for the sake of brevity.

[J4] H. Jamali-Rad, H. Ramezani, and G. Leus, "Sparsity-Aware Multi-Source RSS Localization," *Elsevier Sig. Proc.*, vol. 101, pp. 174-191, Aug. 2014.

[C8] H. Jamali-Rad, H. Ramezani and G. Leus, "Blind Sparsity-Aware Multi-Source Localization" in *Proc. of European Sig. Proc. Conf. (EUSIPCO)*, Morocco, 2013.

[C7] H. Jamali-Rad, H. Ramezani, and G. Leus, "Sparse Multi-Target Localization using Cooperative Access Points," in *Proc. of IEEE Sensor Array Multichannel Proc. Symp. (SAM)*, pp. 353 - 356, NJ, USA, Jun. 2012.

We are eager to explore other potential domains where our sparsity-aware multi-source ideas can be applied. Chapter 7 is the result of a short collaboration with Shell Global Solutions International B.V., where we tried to apply our ideas to a totally different context, i.e., seismic signals in an underground medium. It turns out that finding the location of microseismic fractures (our sources in this context) is of great interest in Geophysics because it can provide a better understanding of the reservoir behavior and can help to optimize the hydraulic fracturing process. Interestingly, creating a fingerprinting map and localizing multiple microseimic sources depends on the knowledge of the source time-function, which is lacking in practical applications. Note that this prerequisite originates from the natural properties of microseismic signals, and is not due to incorporating cross-correlations. However, as is clear, there is a connection to our earlier ideas.

9. Another frequency-domain approach

> The idea is again to analyze the problem in the frequency domain, explore the shared sparsity support and propose another modified G-LASSO estimator which simultaneously takes into account the group structure and the shared sparsity support of the signals in the frequency domain.

Our contributions in this context are published in the following paper.

[J5] H. Jamali-Rad, Z. Tang, X. Campman, A. Droujinine, and G. Leus, "Sparsity-Aware Multiple Microseismic Event Localization Blind to the Source Time-Function," to appear in *Geophysical Prospecting*.

1.6.3 Contributions on Sensor Selection

As we explained earlier in Subsection 1.5.2, we try to find the minimum number of sensors within a network to satisfy a certain estimation performance metric. Particularly, we do this for two reasons. First, this problem turns out to be even more interesting than the aforementioned traditional selection problem from a practical viewpoint. This is because from a cost efficiency perspective, we would like to activate as few sensors as possible rather than selecting k out m. Second, this new problem formulation allows us to explore the sparse structure embedded within this problem.

10. Explore the sparsity

> The idea is that in practice only a few sensors should be activated to satisfy the performance constraint. Therefore, the sensors to be selected are sparse compared to the total number of sensors in the network. This helps us to propose a sparsity-aware solution to the problem.

The problem becomes even more interesting in a distributed configuration when each sensor has to decide itself whether it should contribute to the estimation or not. This is also in line with the critical limitations of WSNs for which in many practical scenarios centralized solutions are useless. This motivates us to give the distributed problem some thought.

11. Distributed implementation

> The idea to distribute the problem is to look at the dual problem and try to find local costs that each sensor should optimize. We solve the resulting sub-problems using a combination of dual subgradient optimization and consensus averaging.

In the above, we adopt a major assumption that the noise experienced by different sensors is uncorrelated. Generally speaking, this is not the case, specially in dense networks where the sensors are usually closely spaced. In such a case, it is expected that neighboring sensors experience correlated noise. The problem becomes even more complicated in this case due to the coupling effects introduced by the noise

correlations. This triggers us to extend our proposed centralized and distributed algorithms to operate in correlated noise scenarios.

12. Handling correlated noise in a distributed scenario

Our first idea to handle correlated noise is by considering clusters of sensors with intra-cluster noise correlations and zero inter-cluster correlations. This helps us to simply extend our sensor-wise operations to cluster-wise ones. Our more elaborate approach is to incorporate ADMM inner-iterations to account for the coupling effects, which allows us to dropping the limiting assumption of having such clusters.

We theoretically prove the convergence of our proposed distributed algorithms as well as analytically quantify their complexity compared to the centralized algorithms.

Finally, our other contribution related to the context of sensor selection which is not presented in this thesis is about another look at the nature of the distributed optimization problem we solve for sensor selection. We study this problem, in a more general framework, as a consensus-based dual decomposition and provide detailed analysis and proofs on its performance.

Our contributions in this context are published in or submitted as the following papers. The first two are contained in Chapters 8 and 9, and the other two are omitted for the sake of brevity.

[J7] H. Jamali-Rad, A. Simonetto, X. Ma, and G. Leus, "Distributed Sparsity-Aware Sensor Selection," submitted to *IEEE Trans. on Sig. Proc. (TSP)*.

[J6, C9] H. Jamali-Rad, A. Simonetto, and G. Leus, "Sparsity-Aware Sensor Selection: Centralized and Distributed Algorithms," *IEEE Sig. Proc. Letters (SPL)*, vol. 21, no. 2, pp. 217-220, Feb. 2014. [Presented in the SPL track of ICASSP 2014]

[J8] A. Simonetto and **H. Jamali-Rad**, "Primal Recovery from Consensus-based Dual Decomposition for Distributed Convex Optimization," submitted to *Journal of Opt. Theory and App. (JOTA)*.

[C10] H. Jamali-Rad, A. Simonetto, X. Ma and G. Leus, "Sparsity-Aware Sensor Selection for Correlated Noise" in *Proc. of Intl. Conf. on Info. Fusion (Fusion 2014)*, Spain, Jul. 2014.

1.6.4 Full List of Contributions

To summarize this introductory chapter, this Ph.D. work has resulted in the publication or submission of 8 journal papers as listed below. The work has also been disseminated at pertinent conferences where the 10 articles listed below have been presented. As we discussed earlier, for the sake of adhering to our main ideas as well as to keep the contents concise and clear, we only present 5 journal papers and 2 conference papers in the next parts of this thesis. Notably, IF stands for the journal impact factor.

Results 1 (Journal publications)

1. A. Simonetto and **H. Jamali-Rad**, "Primal Recovery from Consensus-based Dual Decomposition for Distributed Convex Optimization," submitted to *Journal of Opt. Theory and App. (JOTA)*. [IF: 1.406]
2. **H. Jamali-Rad**, A. Simonetto, X. Ma, and G. Leus, "Distributed Sparsity-Aware Sensor Selection," submitted to *IEEE Trans. on Sig. Proc. (TSP)*. [IF: 2.813]
3. **H. Jamali-Rad**, Z. Tang, X. Campman, A. Droujinine, and G. Leus, "Sparsity-Aware Multiple Microseismic Event Localization Blind to the Source Time-Function," to appear in *Geophysical Prospecting*. [IF: 1.506]
4. **H. Jamali-Rad**, A. Simonetto, and G. Leus, "Sparsity-Aware Sensor Selection: Centralized and Distributed Algorithms," *IEEE Sig. Proc. Letters (SPL)*, vol. 21, no. 2, pp. 217-220, Feb. 2014. [IF: 1.674]
5. **H. Jamali-Rad**, H. Ramezani, and G. Leus, "Sparsity-Aware Multi-Source RSS Localization," *Elsevier Sig. Proc.*, vol. 101, pp. 174-191, Aug. 2014. [IF: 1.745]
6. **H. Jamali-Rad** and G. Leus, "Sparsity-Aware Multi-Source TDOA Localization," *IEEE Trans. on Sig. Proc. (TSP)*, vol. 61, no. 19, Oct. 2013. [IF: 2.813]
7. H. Ramezani, **H. Jamali-Rad**, and G. Leus, "Target Localization and Tracking for an Isogradient Sound Speed Profile," *IEEE Trans. on Sig. Proc. (TSP)*, vol. 61, no. 6, Mar. 2013. [IF: 2.813]
8. **H. Jamali-Rad** and G. Leus, "Dynamic Multidimensional Scaling for Low-Complexity Mobile Network Tracking," *IEEE Trans. on Sig. Proc. (TSP)*, vol. 60, no. 8, pp. 4485-4491, Aug. 2012. [IF: 2.813]

Results 2 (Conference publications)

1. **H. Jamali-Rad**, A. Simonetto, X. Ma and G. Leus, "Sparsity-Aware Sensor Selection for Correlated Noise" in *Proc. of International Conf. on Info. Fusion (Fusion 2014)*, Spain, Jul. 2014. [Special Session]
2. **H. Jamali-Rad**, A. Simonetto, and G. Leus, "Sparsity-Aware Sensor Selection: Centralized and Distributed Algorithms," *IEEE Sig. Proc. Letters (SPL)*, presented in the SPL track of IEEE Conf. on Acoustics, Speech and Sig. Proc. (ICASSP), Italy, May 2014.
3. **H. Jamali-Rad**, H. Ramezani and G. Leus, "Blind Sparsity-Aware Multi-Source Localization" in *Proc. of European Sig. Proc. Conf. (EUSIPCO)*, Morocco, 2013.
4. **H. Jamali-Rad** and G. Leus, "Sparsity-Aware TDOA Localization of Multiple Passive Sources" in *Proc. of IEEE Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, pp. 4021-4025, Canada, May 2013.
5. **H. Jamali-Rad**, H. Ramezani, and G. Leus, "Sparse Multi-Target Localization using Cooperative Access Points," in *Proc. of IEEE Sensor Array Multichannel Proc. Symp. (SAM)*, pp. 353 - 356, NJ, USA, Jun. 2012.
6. H. Ramezani, **H. Jamali-Rad**, and G. Leus, "Localization and Tracking of a Mobile Target for an Isogradient Sound Speed Profile," in *Proc. of IEEE Intl. Conf. on Communications (ICC)*, pp. 3654 - 3658, Canada, Jun. 2012.
7. **H. Jamali-Rad**, H. Ramezani, and G. Leus, "Cooperative Localization in Partially Connected Mobile Wireless Sensor Networks Using Geometric Link Reconstruction," in *Proc. of IEEE Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2633-2636, Japan, 2012.
8. **H. Jamali-Rad**, T. van Waterschoot, and G. Leus, "Cooperative Localization Using Efficient Kalman Filtering for Mobile Wireless Sensor Networks," in *Proc. of European Sig. Proc. Conf. (EUSIPCO)*, pp. 1984-1988, Spain, Aug.-Sep. 2011.
9. **H. Jamali-Rad**, T. van Waterschoot, and G. Leus, "Anchorless Cooperative Localization for Mobile Wireless Sensor Networks" in *Proc. of the Joint WIC/IEEE SP Symp. on Info. Theory and Sig. Proc. in the Benelux (WICSP)*, Belgium, May. 2011.
10. **H. Jamali-Rad**, A. Amar, and G. Leus, "Cooperative Mobile Network Localization via Subspace Tracking," in *Proc. of IEEE Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, pp. 2612 - 2615, Czech Republic, May 2011.

2

PRELIMINARIES

For the sake of a self-contained thesis, in this chapter, we briefly revisit some of the main mathematical methods we use throughout this thesis. The interested readers are referred to the corresponding references for more elaborate details.

2.1 Multidimensional Scaling

The idea of multidimensional scaling (MDS) was initially proposed in psychometrics [40] as a means of visualizing the level of similarity (or dissimilarity) of individual cases of a dataset. In other words, the goal of MDS is to find a representation of n points in a certain dimension so that their pairwise distances as good as possible fit a measured set of dissimilarities between these points [41]. MDS has found a wide variety of applications in different domains such as sociology, political sciences [42], machine learning [43], and signal processing [44], which is of special concern to us.

In technical terms, MDS is normally referred to as an approach to solve the aforementioned *dimensionality reduction* problem as we look for a representation in a lower dimension. There exist several methods in literature to perform this procedure among which we can note the following two prominent ones.

2.1.1 Classical MDS

If all the measured pairwise distances are noiseless, classical MDS is capable of recovering the correct configuration of points (up to a translation and orthogonal transformation) as is explained in the following. Let us consider that our n points $\mathbf{x}_i \in \mathbb{R}^d$ are stacked in $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$, where d is the number of dimensions ($d < n$), also called the embedding dimension [45]. Let us also define a centering operator $\mathbf{\Gamma}$ as

$$\mathbf{\Gamma}_n = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T, \tag{2.1}$$

where $\mathbf{I}_n$ denotes the $n \times n$ identity matrix and $\mathbf{1}_n$ represents the $n \times 1$ vector of all ones. The following lemma explains why $\mathbf{\Gamma}$ is called a centering operator.

Lemma 2.1 (*Centering operator*)

Given $\mathbf{X}$, $\mathbf{X}_c = \mathbf{X}\,\boldsymbol{\Gamma}_n$, *denotes a related set of points centered at the origin.*

Proof. The geometric center of $\mathbf{X}$ is given by

$$\mathbf{x}_c = \frac{1}{n}\mathbf{X}\,\mathbf{1}_n.$$

Thus,

$$\begin{aligned}\mathbf{X}\,\boldsymbol{\Gamma}_n &= \mathbf{X}\,(\mathbf{I}_n - \frac{1}{n}\,\mathbf{1}_n\,\mathbf{1}_n^T)\\ &= \mathbf{X} - \frac{1}{n}\,(\mathbf{X}\,\mathbf{1}_n)\,\mathbf{1}_n^T\\ &= \mathbf{X} - \mathbf{x}_c\,\mathbf{1}_n^T = \mathbf{X}_c,\end{aligned}$$

and the claim follows. ■

Next, we define an Euclidean distance matrix (EDM) $[\mathbf{D}]_{i,j} = d_{ij}^2$, $\forall i,\ j = 1, \cdots, n$, where $d_{ij}^2 = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$. It is straightforward to verify that

$$\mathbf{D} = \text{diag}(\mathbf{X}^T\mathbf{X})\mathbf{1}_n^T - 2\mathbf{X}^T\mathbf{X} + \mathbf{1}_n\text{diag}(\mathbf{X}^T\mathbf{X})^T. \tag{2.2}$$

Let us also define

$$D(\mathbf{X}) = \text{diag}(\mathbf{X}^T\mathbf{X})\mathbf{1}_n^T - 2\mathbf{X}^T\mathbf{X} + \mathbf{1}_n\text{diag}(\mathbf{X}^T\mathbf{X})^T,$$

where $\text{diag}(\mathbf{A})$ returns a vector containing the diagonal elements of $\mathbf{A}$, and where we introduce $D(\mathbf{X})$ as the function that returns a noiseless EDM from the set of points $\mathbf{X}$. From (2.1), we know that $\boldsymbol{\Gamma}_n\mathbf{1}_n = \mathbf{1}_n^T\boldsymbol{\Gamma}_n = \mathbf{0}_n$ with $\mathbf{0}_n$ denoting the $n \times 1$ all-zero vector. Therefore, it is easy to confirm that

$$-\frac{1}{2}\boldsymbol{\Gamma}_n\mathbf{D}\boldsymbol{\Gamma}_n = \boldsymbol{\Gamma}_n\mathbf{X}^T\mathbf{X}\boldsymbol{\Gamma}_n,$$

where $\mathbf{B} = -\frac{1}{2}\boldsymbol{\Gamma}_n\mathbf{D}\boldsymbol{\Gamma}_n$ is a symmetric positive semidefinite (PSD) matrix. Besides, $\mathbf{B}$ and $\mathbf{D}$ have some rank properties [45] (presented in the following lemma) which play an important role in the derivation of the classical MDS.

Theorem 2.1 (*Rank of* $\mathbf{D}$ *and* $\mathbf{B}$)

Given $\text{rank}(\mathbf{X}) = d$, $\text{rank}(\mathbf{D}) \leqslant d + 2$, *and* $\text{rank}(\mathbf{B}) = d$.

Proof. Obviously, $\text{rank}(\mathbf{X}) = d$ if all the $\mathbf{x}_i$'s do not lie on a $(d-1)$-dimensional hyperplane at the same time. From (2.2), defining $\boldsymbol{\psi} = \text{diag}(\mathbf{X}^T\mathbf{X})$, we have

$$\begin{aligned} \text{rank}(\mathbf{D}) &\leqslant \text{rank}(\boldsymbol{\psi}\mathbf{1}_n^T) + \text{rank}(2\mathbf{X}^T\mathbf{X}) + \text{rank}(\mathbf{1}_n\boldsymbol{\psi}^T) \\ &\leqslant 1 + d + 1 = d + 2, \end{aligned}$$

where we have used $\text{rank}(\mathbf{A}+\mathbf{B}) \leqslant \text{rank}(\mathbf{A}) + \text{rank}(\mathbf{B})$. Now, we can write

$$\begin{aligned} \text{rank}(\mathbf{B}) &= \text{rank}(\boldsymbol{\Gamma}_n\mathbf{X}^T\mathbf{X}\boldsymbol{\Gamma}_n) = \text{rank}(\mathbf{X}\boldsymbol{\Gamma}_n) \\ &= \text{rank}(\mathbf{X}) = d, \end{aligned}$$

where we have used $\text{rank}(\mathbf{X}^T\mathbf{X}) = \text{rank}(\mathbf{X})$. ■

Finally, given $\mathbf{B}$ one can recover $\mathbf{X}$ (and thus the points), up to a translation and orthogonal transformation, as the solution to the following optimization problem

$$\min_{\tilde{\mathbf{X}}} \ \|\mathbf{B} - \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\|_F,$$

with $\tilde{\mathbf{X}} = \boldsymbol{\Psi}\mathbf{X}\boldsymbol{\Gamma}_n$, and $\boldsymbol{\Psi}$ denoting an arbitrary orthogonal transformation, where the minimum is searched over all possible $d \times n$ rank-d matrices [41]. Given the singular value decomposition (SVD) of our rank-d and symmetric PSD $\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T$, we have

$$\tilde{\mathbf{X}} = \boldsymbol{\Sigma}_d^{1/2}\mathbf{U}_d^T,$$

where $\boldsymbol{\Sigma}_d$ denotes a $d \times d$ diagonal matrix containing the d largest singular values, and $\mathbf{U}_d$ contains the corresponding orthonormal eigenvectors. Note that even though classical MDS returns exact solutions in the noiseless case, there is no guarantee to return an optimal solution in noisy scenarios.

2.1.2 Stress Function Minimization

The MDS problem has also been looked at from an optimization perspective; however, enforced by the nature of the problem the proposed optimization problems are nonlinear and non-convex. An example is to minimize a weighted version of distance errors (instead of squared distance errors in classical MDS) the so-called "raw Stress" as in [46]

$$\min_{\mathbf{X}} \ \sum_{i,j} w_{ij} \left(\sqrt{[\mathbf{D}]_{i,j}} - \sqrt{[D(\mathbf{X})]_{i,j}}\right)^2, \tag{2.3}$$

where the weights w_{ij} are supposed to quantify the accuracy of the measured values in $\mathbf{D}$. The fact that (2.3) is also non-differentiable leads to formidable solutions such as iterative majorization [47] and steepest decent methods [48]. An alternative in the same family of optimization problems is the so-called "S-Stress" as in [49]

$$\min_{\mathbf{X}} \sum_{i,j} w_{ij} \left([\mathbf{D}]_{i,j} - [D(\mathbf{X})]_{i,j}\right)^2, \tag{2.4}$$

which is differentiable all over its domain. On the other hand, a disadvantage of the S-Stress function is that it favors larger distances over smaller ones [49]. The S-Stress is solved in a distributed fashion in [41] to develop an distributed weighted MDS (dwMDS). In this thesis, in Part II, we mainly focus on the classical MDS due to its elegance and the simplicity of its solution.

2.2 Subspace Perturbation Expansion

Subspace perturbation expansion explains how much subspace perturbation is induced by additive noise in data. This method can be used to derive optimally weighted subspace fitting algorithms for different estimation problems [50, 51]. Here is the main message. Let us assume that $\mathbf{B}$ is an $m \times n$ matrix with rank $p < \min(m, n)$. The SVD of $\mathbf{B}$ can be given by

$$\mathbf{B} = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix} \begin{bmatrix}\mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix} \begin{bmatrix}\mathbf{V}_1^H \\ \mathbf{V}_2^H\end{bmatrix},$$

where $(.)^H$ stands for Hermitian, and $\mathbf{U}_1$ and $\mathbf{U}_2$ respectively represent the $m \times p$ and $m \times (m-p)$ matrices whose columns are an orthonormal basis for the column space and null space of $\mathbf{B}$. Clearly, $\mathbf{\Sigma}_1$ contains the singular values and $\mathbf{V}_1$ and $\mathbf{V}_2$ can also be defined similar to $\mathbf{U}_1$ and $\mathbf{U}_2$. We are basically interested in $\text{col}(\mathbf{U}_1)$ and $\text{col}(\mathbf{U}_2)$ where $\text{col}(.)$ stands for the column space of a matrix. Now, let $\mathbf{B}$ be perturbed as $\tilde{\mathbf{B}} = \mathbf{B} + \Delta\mathbf{B}$. The SVD of $\tilde{\mathbf{B}}$ can then be given by

$$\tilde{\mathbf{B}} = \begin{bmatrix}\tilde{\mathbf{U}}_1 & \tilde{\mathbf{U}}_2\end{bmatrix} \begin{bmatrix}\tilde{\mathbf{\Sigma}}_1 & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{\Sigma}}_2\end{bmatrix} \begin{bmatrix}\tilde{\mathbf{V}}_1^H \\ \tilde{\mathbf{V}}_2^H\end{bmatrix}.$$

It is shown in [50] that we can find orthonormal bases for $\text{col}(\tilde{\mathbf{U}}_1)$ and $\text{col}(\tilde{\mathbf{U}}_2)$, i.e., a basis for the column space and null space of the perturbed matrix $\tilde{\mathbf{B}}$, respectively given by

$$\text{col}(\tilde{\mathbf{U}}_1) = (\mathbf{U}_1 + \mathbf{U}_2\mathbf{P})(\mathbf{I} + \mathbf{P}^H\mathbf{P})^{-\frac{1}{2}}, \tag{2.5}$$

$$\text{col}(\tilde{\mathbf{U}}_2) = (-\mathbf{U}_1\mathbf{P}^T + \mathbf{U}_2)(\mathbf{I} + \mathbf{P}\mathbf{P}^H)^{-\frac{1}{2}}, \tag{2.6}$$

where $\mathbf{P}$ is a coefficient matrix. It turns out that $\mathbf{P}$ can be written as a series of terms $\mathbf{P} = \mathbf{0} + \mathbf{P}^{(1)} + \mathbf{P}^{(2)} + \cdots$, where $\mathbf{P}^{(i)}$ refers to a matrix product containing $\Delta\mathbf{B}^i$. For instance, the computations in [50] show that resorting to only first-order terms leads to $\mathbf{P} = \mathbf{P}^{(1)} = \mathbf{U}_2\Delta\mathbf{B}\mathbf{V}_1\mathbf{\Sigma}_1^{-1}$. Higher-order approximations of $\mathbf{P}$ can be found in [50]. Now, the important question is, are there any restrictions on the amount of perturbation reflected in $\Delta\mathbf{B}$ for (2.5)-(2.5) to hold? The answer is yes, and it is explained in the following remark.

Remark 2.1 (*How small should the perturbations be?*)

> *The derivations in* (2.5)-(2.6) *are based on the assumption that* $\Delta\mathbf{B}$ *is small enough so that the invariant subspace of the perturbed matrix* $\tilde{\mathbf{B}}$ *(i.e.,* $col(\tilde{\mathbf{U}}_1)$*) does not contain any vector that is orthogonal to the invariant subspace of the unperturbed matrix* $\mathbf{B}$ *(i.e.,* $col(\mathbf{U}_1)$*).*

As we explained in Chapter 1, we exploit the concept of subspace perturbation expansion in Part II to derive a dynamic MDS.

2.3 Power Method

The power method is used to compute the dominant eigenvector of a diagonalizable matrix. A natural extension, which is of interest in this thesis, is the computation of an invariant subspace of a diagonalizable matrix using orthogonal (power) iterations. The orthogonal iterations start from an initial guess of the desired subspace $\mathbf{U}^{(0)}$ and iterates as follows

$$\mathbf{U}^{(i+1)} = \mathbf{B}\mathbf{U}^{(i)}, \tag{2.7}$$

where $\mathbf{U}$ is the desired subspace of $\mathbf{B}$ and the superscript (i) denotes the i-th iteration. Under some conditions, (2.7) is shown to asymptotically converge to the true subspace if in each iteration $\mathbf{U}^{(i+1)}$ is orthonormalized using any possible method such as a Gram-Schmidt process [52]. It is notable that the number of required iterations as well as a smooth convergence of the orthogonal iterations are both dependent upon the choice of the initial guess $\mathbf{U}^{(0)}$. Therefore, an improper initial guess not only leads to a large number of iterations, but it might also result in the divergence of the algorithm.

2.4 Sparse Reconstruction and Compressive Sensing

In this subsection, we briefly introduce the celebrated concept of *sparse reconstruction* and *compressive sensing (CS)*. This is of particular interest in this thesis

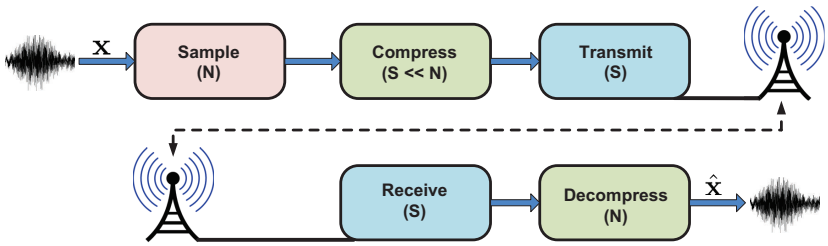


Figure 2.1: Compression/decompression process. The number of samples is shown in each block, where as can be seen the signal is N/S times compressed.

because the main flavor of the Parts III-IV revolves to a great extent around the concept of sparsity.

Let us start by stating that in nowadays world we are drowning in a huge amount of data measured by many different types of sensors. This huge amount of information also called "big data" calls for either a vast amount of storage, which is lacking or expensive in principle, or efficient ways to handle such a big buck of data. Delving deeper into this issue from a signal processing perspective reveals that part of this originates from the fact that many of the signals being measured have an extremely high frequency content, for example high quality images. Therefore, following the classical *Nyquist-Shannon* theorem, these signals should be sampled at a very high rate (to be more precise, at least two times the highest frequency) in order to be reconstructed with proper quality from the samples. This significantly increases the number of sensors sampling at high rates resulting in a "deluge of data" [53].
A traditional wisdom to deal with this situation is to compress the buck of data and then store/transmit it, and decompress it when it is required to be processed (see Fig. 2.1). However, this still mandates taking all those samples, which is expensive and might not be used after all. A promising alternative, which has received an upsurge of attention recently, is to think of a new generation of data acquisition systems with *compressive* sensors which try to measure only the required information (thus producing a much smaller amount of data) rather than measuring a massive amount of data and then compressing it [54]. The key idea behind this innovative trend is to recognize that many practical signals actually live in a very low-dimensional space. In other words, many natural signals have a *low-dimensional model* which can be due to their sparsity or low-rank structure. The idea of CS is to exploit this low-dimensional structure in the design of signal processing algorithms; the main message is to sample *smarter* not *faster* [55]!

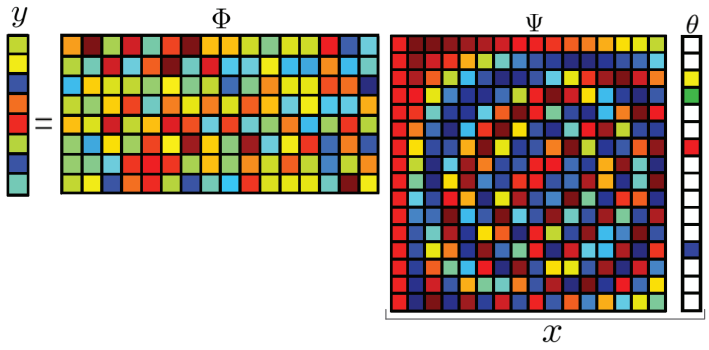


Figure 2.2: An underdetermined measurement system. Image is taken from [54] and is modified.

In order to make it look like what we will be dealing with in the next parts of this thesis, we model the sampling problem in a more general form as a linear set of equations like

$$\mathbf{y} = \mathbf{\Phi x},$$

where $\mathbf{x}$ of size $N \times 1$ contains the original signal (for instance, a vector containing the pixels of an image), $\mathbf{\Phi}$ of size $M \times N$ is the measurement matrix or sometimes called sampling matrix, and $\mathbf{y}$ of size $M \times 1$ is the vector containing the measurements. Now, it is well-known that if $M > N$ and $\mathbf{\Phi}$ has full column rank, recovering $\mathbf{x}$ is trivial through the classical LS as

$$\hat{\mathbf{x}}_{\text{LS}} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{y}.$$

On the other hand, if $M < N$ and $\mathbf{\Phi}$ is full row rank (similar to Fig. 2.2), i.e., an underdetermined linear set of equations, the problem can have infinite solutions. A simple example is to take any vector from the null-space of $\mathbf{\Phi}$ and add it to $\mathbf{x}$, the sum will return exactly the same $\mathbf{y}$. Note that this is clearly a case of dimensionality reduction as our measurements inhabit a space with a dimension that is smaller than the dimension of the space our natural data lives in. The larger N/M, the worse the situation, because we lose information by this "under-sampling".

What happens in a CS framework is that we have much less measurements than the number of unknowns $M \ll N$ and still we would like to recover $\mathbf{x}$. The key enabler to realize this is the concept of sparsity. But what is sparsity? Generally speaking, any $\mathbf{x}$ can be written as

$$\mathbf{x} = \sum_{j=1}^{N} \theta_j \boldsymbol{\psi}_j,$$

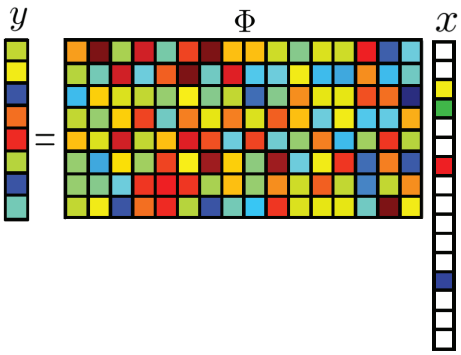


Figure 2.3: Sparsifying basis. Image is taken from [54] and is modified.

which is the expansion of $\mathbf{x}$ using a basis defined by the columns of $\mathbf{\Psi}$. Fig. 2.2 illustrates a specific case where only a few columns of $\mathbf{\Psi}$ contribute to the construction of $\mathbf{x}$. This means $\mathbf{x}$ is sparse within the basis defined by $\mathbf{\Psi}$, and thus $\mathbf{\Psi}$ is sometimes referred to as the *sparsifying basis*. To be more specific, when the representation of a vector has only K significant elements (and the remaining $N - K \gg K$ elements are almost negligible), we call it a K-sparse vector. In plain terms, the theory of CS says that if $\mathbf{x}$ is sparse in some given domain, and $\mathbf{\Phi}$ (or better to say $\mathbf{\Phi\Psi}$) satisfies specific properties, even though the problem is highly underdetermined, we can recover $\mathbf{x}$ with a very high quality. For the sake of simplicity of our notations, let us consider that we deal with

$$\mathbf{y} = \mathbf{\Phi x},$$

where $\mathbf{x}$ is sparse in the basis $\mathbf{\Psi} = \mathbf{I}_N$, and $\mathbf{\Phi}$ is an $M \times N$ matrix with $M \ll N$ as shown in Fig. 2.3. Now, the point is that we are not interested in recovering any general $\mathbf{x}$, but an $\mathbf{x}$ with only a few (let us say K) non-zero elements. This means we know that only K columns of $\mathbf{\Phi}$ contribute to the construction of $\mathbf{y}$. As an intuition why such a problem can be solved, assume that we know which columns of $\mathbf{\Phi}$ contribute to the construction of $\mathbf{y}$, or equivalently we know which elements in $\mathbf{x}$ are significant. Then, after removing the irrelevant columns in $\mathbf{\Phi}$ and the irrelevant elements in $\mathbf{x}$, we can basically solve a new problem which is not underdetermined anymore. The difficulty is that, in practice, we do not know the indices of the contributing columns.

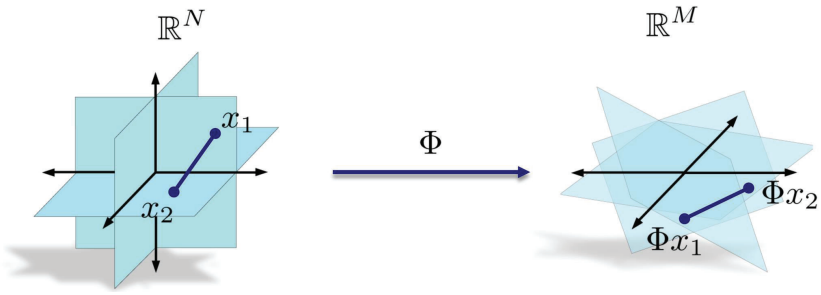


Figure 2.4: Distance preservation in projection. Image is taken from [56].

2.4.1 Restricted Isometry Property

The important question is what are the conditions that $\mathbf{\Phi}$ should satisfy to ensure a proper recovery of $\mathbf{x}$? The answer has been studied in great detail in literature [57, 58]. In the following, we try to specify those conditions in simple terms. A proper $\mathbf{\Phi}$ should be such that any randomly chosen K columns should be full column rank (Rank-K). On top of that, we also want those K columns to be as close as possible to orthogonal to each other. This is in principle what is called the restricted isometry property (RIP). To clarify things better, let us look at the RIP from a geometrical angle as is shown in Fig. 2.4. Suppose we have a set of vectors (like $\mathbf{x}_1$ and $\mathbf{x}_2$) in $\mathbb{R}^N$ each of which is sparse, and thus, lives in a much lower dimensional subspace $\mathbb{R}^K$. Here, $\mathbf{\Phi}$ acts as a projection operator which forces our points to fit into a lower dimensional space in $\mathbb{R}^M$. The RIP says that, as long as our vectors are sparse, $\mathbf{\Phi}$ is a proper matrix if when applied, it (approximately) preserves the distances between the vectors as

$$\|\mathbf{x}_1 - \mathbf{x}_2\|_2 \approx \|\mathbf{\Phi}\mathbf{x}_1 - \mathbf{\Phi}\mathbf{x}_2\|_2.$$

Note that this distance preservation means nothing but *information preservation* because this way we make sure our information components will not be confused after projection. More elaborate definitions of the RIP are based on the aforementioned concept of distance preservation. Two such common definitions are as follows. A popular definition is given in [54], where for $K = 1, 2, \cdots$, the RIP constant δ_K of a matrix $\mathbf{\Phi}$ (with normalized columns) is the smallest number that satisfies

$$-\delta_K \leqslant \frac{\|\mathbf{\Phi}\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} - 1 \leqslant \delta_K,$$

for all K-sparse $\mathbf{x} \in \mathbb{R}^N$. Roughly speaking, as long as $0 < \delta_K < 1$, the RIP holds. Evidently, this is an NP-hard combinatorial problem. A computationally less demanding definition is given in [59], where δ_K is defined as the maximum distance from 1 of all the eigenvalues of the $\binom{N}{K}$ submatrices, $\mathbf{\Phi}_\Lambda^H \mathbf{\Phi}_\Lambda$, derived from $\mathbf{\Phi}$, where Λ is a set of indices with cardinality K which selects those columns of $\mathbf{\Phi}$ indexed by Λ. Hence, for each K, the RIP constant is given by

$$\delta_K = \max\left(|\lambda_{\max}(\mathbf{\Phi}_\Lambda^H \mathbf{\Phi}_\Lambda) - 1|,\ |\lambda_{\min}(\mathbf{\Phi}_\Lambda^H \mathbf{\Phi}_\Lambda) - 1|\right).$$

2.4.2 Common Choices for the Measurement Matrix

An extensive body of research in literature shows that even though verifying the RIP for a generic matrix is a tough problem, there are big classes of matrices for which the RIP (approximately) holds. Three widely-used classes are given below.

- ◇ *Gaussian matrices*: An interesting example is to fill the elements of $\mathbf{\Phi}$ with independent identically distributed (i.i.d.) samples drawn from a Gaussian distribution as $[\mathbf{\Phi}]_{i,j} \sim \mathcal{N}(0, 1/M)$. In this case, if $M = O(K \log(N/K))$, $\mathbf{\Phi}$ satisfies the RIP with a very high probability.

- ◇ *Bernoulli matrices*: A Bernoulli matrix is comprised of independent and equiprobable elements taken from $\{+1/\sqrt{M}, -1/\sqrt{M}\}$. Similar to the case of Gaussian matrices, if $M = O(K \log(N/K))$, $\mathbf{\Phi}$ satisfies the RIP with a very high probability.

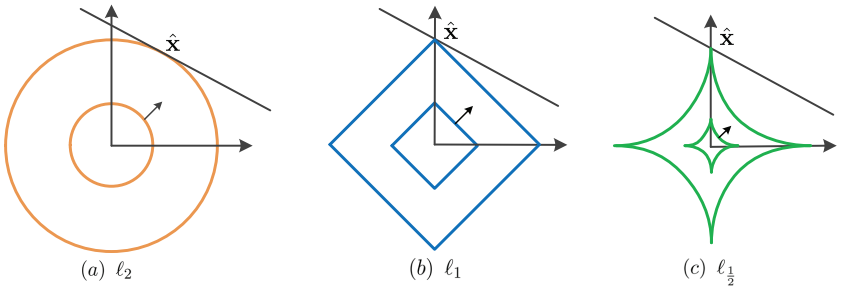
- ◇ *Fourier matrices*: A Fourier matrix is constructed by randomly selecting M rows from an $N \times N$ Fourier matrix, and normalizing the columns of the resulting matrix. It is proved in [60] that if $M = O\left(K(\log(N))^6\right)$, the RIP holds with a high probability. This result has been further improved to $M = O\left(K(\log(N))^4\right)$ in [61].

2.4.3 Sparse Recovery

Before we move on, bear in mind that the ℓ_p norm of a given $N \times 1$ vector $\mathbf{x}$ is defined as

$$\|\mathbf{x}\|_p = \begin{cases} \left(\sum_{i=1}^N |x_i|^p\right)^{\frac{1}{p}} & 0 < p < \infty \\ \max(|x_1|, \cdots, |x_N|), & p = \infty \end{cases}.$$

A special case is the ℓ_0 norm $\|\mathbf{x}\|_0$ which counts the number of non-zero elements of $\mathbf{x}$.



(a) ℓ_2 (b) ℓ_1 (c) $\ell_{\frac{1}{2}}$

Figure 2.5: Estimation with ℓ_p norms for $p = 2$, 1, and 0.5

Knowing that $\mathbf{\Phi}$ is properly designed, and $\mathbf{x}$ is sparse, the questions is how to recover $\mathbf{x}$ from $\mathbf{y}$? As we discussed earlier, without any prior knowledge on $\mathbf{x}$ there are infinitely many solutions for the problem at hand. One way to deal with it is to somehow go for a favorite solution within the space of possible solutions. A natural choice is LS; or in other words, to model the problem as

$$\begin{aligned} \hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \quad & \|\mathbf{x}\|_2 \\ \text{s.t.} \quad & \mathbf{y} = \mathbf{\Phi}\mathbf{x}, \end{aligned} \tag{2.8}$$

which is basically seeking a minimum energy $\mathbf{x}$. However, the ℓ_2 norm is not an optimal option. This is illustrated in Fig. 2.5-(a) within a hypothetical 2-dimensional setup. The black line represents the set of all feasible solutions. Solving (2.8) means blowing the ℓ_2 norm circle until it touches the black line. Depending on the angle of the black line, the solution $\hat{\mathbf{x}}$ does not necessarily lie on one of the coordinate axes, and thus yields a non-sparse solution. Now, the question is what is a proper norm if ℓ_2 is not. The answer clearly is the ℓ_0 norm which promotes sparsity the best as

$$\begin{aligned} \hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \quad & \|\mathbf{x}\|_0 \\ \text{s.t.} \quad & \mathbf{y} = \mathbf{\Phi}\mathbf{x}. \end{aligned} \tag{2.9}$$

However, (2.9) is known to be combinatorial (more specifically NP-complete) and non-convex.

The great idea is to convexify this non-convex problem and to go for the ℓ_1 norm. Then, we solve

$$\begin{aligned} \hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \quad & \|\mathbf{x}\|_1 \\ \text{s.t.} \quad & \mathbf{y} = \mathbf{\Phi}\mathbf{x}. \end{aligned} \tag{2.10}$$

The ground-breaking results in [57, 58, 62] illustrate that if RIP is satisfied, (2.10) and (2.9) return exactly the same results. Furthermore, (2.10) can be solved in polynomial time because it is essentially a linear programming (LP) problem. As is clear from Fig. 2.5-(b) the ℓ_1 norm yields a diamond which is more pointy than a ball with a higher probability of touching the feasible set on one of the the coordinate axes. Following the same line, the situation gets even better with ℓ_p norms defined with $0 < p < 1$, as is also shown in Fig. 2.5-(c) for $p = 1/2$. On the other hand, solving an ℓ_p-norm optimization enforces an extra computational cost. Our focus for sparse reconstruction in this thesis is based on employing ℓ_1 norms and solving (2.10). Note that the aforementioned problems are given for noiseless samples/measurements. In case of noisy measurements such as $\mathbf{y} = \mathbf{\Phi x} + \mathbf{e}$, where $\mathbf{e}$ denotes the noise vector, instead of (2.10), we solve

$$\begin{aligned} \hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \quad & \|\mathbf{x}\|_1 \\ \text{s.t.} \quad & \|\mathbf{y} - \mathbf{\Phi x}\| \leqslant \epsilon. \end{aligned} \tag{2.11}$$

2.4.4 Recovery Algorithms

A vast variety of algorithms has been developed in literature to solve the sparse recovery problem. In general, these algorithms fall broadly into two main categories: *greedy pursuit* algorithms and *convex relaxation* algorithms.

Generally speaking, a greedy pursuit method refers to an algorithm which chooses the best immediate or local optimum at each stage and it is eventually expected to find the global optimum. The greedy pursuit algorithms operate by selecting atoms iteratively (finding the indices of the non-zero elements in the sparse vector), and subtracting the contribution of each selected atom from the signal residual. This selection/removal process is repeated until a stopping criterion is met. The stopping criterion is either to meet a target sparsity level, or to ensure that the magnitude of the residual gets smaller than a pre-determined threshold. Notable examples of greedy pursuit approaches include matching pursuit (MP) [63], orthogonal matching pursuit (OMP) [64], or a family of subspace pursuit algorithms including CoSaMP [65].

The convex relaxation algorithms are based on relaxing the non-convex ℓ_0 norm in (2.9) with a convex objective ℓ_1 norm as in (2.10) which can be solved efficiently with existing linear (or convex) solvers. The relaxation is also commonly known as basis pursuit (BP) [66]. In noisy conditions, especially when there is no knowledge about the noise, (2.11) can be reformulated as an unconstrained optimization

problem given by

$$\hat{\mathbf{x}}_{\text{LASSO}} = \underset{\mathbf{x}}{\text{argmin}} \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{2.12}$$

where λ is a regularization (fidelity) term which trades off sparsity for accuracy. A proper value for λ can be found using a method called cross-validation (CVD) [67]. Commonly used optimization programs to solve (2.12) are the least absolute shrinkage and selection operator (LASSO) [68] or basis pursuit denoising (BPDN) [66].

The greedy pursuit algorithms are usually harder to analyze in terms of performance compared to the mathematically elegant convex relaxation algorithms. On the other hand, the greedy pursuit algorithms have proven computationally efficient and easier to implement while providing almost similar performance compared to the convex relaxation based algorithms. In this thesis, we mainly focus on convex optimization algorithms, especially LASSO. Besides, we also face with cases where the sparse vector of interest is comprised of p non-overlapping groups $\mathbf{x} = [\mathbf{x}_1^T, \cdots, \mathbf{x}_p^T]^T$. Accordingly, we have $\mathbf{\Phi} = [\mathbf{\Phi}_1, \cdots, \mathbf{\Phi}_p]$. We expect sparsity at the group level meaning that the entire group will be either zero or non-zero. Due to the embedded group structure, the solver for this problem is called group LASSO (G-LASSO) [69] and it solves

$$\hat{\mathbf{x}}_{\text{G-LASSO}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \sum_{i=1}^{p} \mathbf{\Phi}_i\mathbf{x}_i\|_2^2 + \lambda \sum_{i=1}^{p} \|\mathbf{x}_i\|_2.$$

2.5 Distributed Optimization via ADMM

A big class of problems in nowadays world deal with an extremely large amount of high dimensional data or so-called “big data”. Such problems arise in different domains including machine learning, statistics, and dynamic optimization in large-scale networks. This sort of problems prompt designing systems and algorithms which are scalable enough to handle huge data sets in a parallel or decentralized fashion [70]. Many of these problems can be modeled as convex optimization problems, and thus, they can be approached using well-known distributed optimization techniques such as alternating direction method of multipliers (ADMM). Before detailing ADMM, we would first like to briefly go over two preliminary concepts on which ADMM is based, namely, dual decomposition and method of multipliers.

2.5.1 Dual Decomposition

Consider an equality-constrained optimization problem

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \end{aligned} \tag{2.13}$$

with $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, and $f : \mathbb{R}^n \to \mathbb{R}$. A specific case of such an optimization problem occurs when the objective function is separable w.r.t. the variables as

$$f(\mathbf{x}) = \sum_{i=1}^{N} f_i(\mathbf{x}_i),$$

where $\mathbf{x} = [\mathbf{x}_1^T, \cdots, \mathbf{x}_N^T]^T$, and the vectors $\mathbf{x}_i \in \mathbb{R}^{n_i}$ are subvectors of $\mathbf{x}$. A particular case occurs when the $\mathbf{x}_i$'s are single elements. Therefore, we can write

$$\mathbf{A}\mathbf{x} = \sum_{i=1}^{N} \mathbf{A}_i \mathbf{x}_i,$$

by a proper partitioning of $\mathbf{A}$ as $\mathbf{A} = [\mathbf{A}_1, \cdots, \mathbf{A}_N]$. This means that the Lagrangian of the problem can also be split as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{N} \mathcal{L}_i(\mathbf{x}_i, \boldsymbol{\lambda}) = \sum_{i=1}^{N} \left(f_i(\mathbf{x}_i) + \boldsymbol{\lambda}^T \mathbf{A}_i \mathbf{x}_i - \frac{1}{N} \boldsymbol{\lambda}^T \mathbf{b} \right),$$

where $\boldsymbol{\lambda}$ is a properly-sized dual variable. Given the above separable structure, and following a dual ascent optimization recursion, a solution to the above problem is

$$\mathbf{x}_i^{(k+1)} = \underset{\mathbf{x}_i}{\operatorname{argmin}} \ \mathcal{L}_i(\mathbf{x}_i, \boldsymbol{\lambda}^{(k)}), \ i = 1, \cdots, N, \tag{2.14}$$

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \alpha^{(k)} (\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{b}), \tag{2.15}$$

where superscript (k) indicates the k-th iteration, (2.14) is run in parallel for all i, and $\alpha^{(k)}$ is the step size. As is clear from (2.14)-(2.15), a gather-scatter paradigm is observed in our primal-dual update solution, where first in parallel in (2.14) the subvectors $\mathbf{x}_i$ are optimized based on the current value of the dual variable. Next, in (2.15) these subvectors are gathered to evaluate the residual which is used to update the dual variable. Notice that we are readily doing a distributed optimization. Dual decomposition (DD) is known to be slow in terms of convergence and working only under a strict set of conditions making it a very fragile algorithm [70, 71].

2.5.2 Method of Multipliers

In order to cope with the performance issues of DD (being slow and fragile), usually an augmented Lagrangian is employed. This is done by adding a (non-negative) quadratic penalty term to the normal Lagrangian of (2.13) as

$$\mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

The solution which is called the method of multipliers (MM) is similar to the one of DD and is given by

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \underset{\mathbf{x}}{\operatorname{argmin}}\, \mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}^{(k)}), \\ \boldsymbol{\lambda}^{(k+1)} &= \boldsymbol{\lambda}^{(k)} + \rho(\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{b}), \end{aligned}$$

with the specific choice of step size $\alpha^{(k)} = \rho$ which leads to both primal and dual feasibility. Notably, MM converges under much more general conditions compared to dual ascent including even the case where $f(.)$ is not even strictly convex [70]. On the other hand, adding the quadratic term destroys the separability of the problem, i.e., we cannot decompose the problem anymore. Now, the important question is how can we combine the robustness of MM and separability offered by DD? A popular answer which has drawn lots of attention is ADMM.

2.5.3 ADMM

The ADMM can solve problems of the following form

$$\begin{aligned} \underset{\mathbf{x},\mathbf{y}}{\text{minimize}} \quad & f(\mathbf{x}) + g(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{c}, \end{aligned} \tag{2.16}$$

with $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{p\times n}$, $\mathbf{B} \in \mathbb{R}^{p\times m}$, and $\mathbf{c} \in \mathbb{R}^p$. The only main assumptions are that $f(.)$ and $g(.)$ are convex. Let us construct the augmented Lagrangian of (2.16) as

$$\mathcal{L}_\rho(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) = f(\mathbf{x}) + g(\mathbf{y}) + \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c}\|_2^2. \tag{2.17}$$

The solution to (2.17) which resembles a lot the one of MM can be given by

$$\mathbf{x}^{(k+1)} = \underset{\mathbf{x}}{\operatorname{argmin}}\, \mathcal{L}_\rho(\mathbf{x}, \mathbf{y}^{(k)}, \boldsymbol{\lambda}^{(k)}), \tag{2.18}$$

$$\mathbf{y}^{(k+1)} = \underset{\mathbf{y}}{\operatorname{argmin}}\, \mathcal{L}_\rho(\mathbf{x}^{(k+1)}, \mathbf{y}, \boldsymbol{\lambda}^{(k)}), \tag{2.19}$$

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \rho(\mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{B}\mathbf{y}^{(k+1)} - \mathbf{c}). \tag{2.20}$$

Note that (2.16) can clearly be seen as a small example of DD problem with $N = 2$, $\mathbf{A} = \mathbf{A}_1$, $\mathbf{B} = \mathbf{A}_2$, $\mathbf{c} = \mathbf{b}$, $f(.) = f_1(.)$, and $g(.) = f_2(.)$. Also, notice that with ADMM, in contrast to MM, we never optimize over both $\mathbf{x}$ and $\mathbf{y}$ at the same time but instead we have a single pass of Gauss-Seidel pass as given by (2.18)-(2.19). There is also a slightly modified version of ADMM in which we basically combine the linear and quadratic terms, the so-called scaled ADMM [70]. We omit it here for the sake of space limitation.

The assumptions on the convergence of ADMM are quite general and they are also extensively studied in literature. The important point to highlight is that ADMM can be very slow to converge if a high accuracy in required. However, in many applications, such as the one we consider in Part IV, only a modest level of accuracy is needed which can be obtained quite fast with only a few iterations.

2.5.4 Consensus with ADMM

A class of optimization problems that can be solved in a distributed fashion using ADMM is a *consensus* problem given by

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) = \sum_{i=1}^{N} f_i(\mathbf{x}),$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ are convex. This is a practical problem that appears in many contexts such as distributed wireless networks. The problem can be rewritten by defining a *dummy variable* $\mathbf{z}$ as

$$\begin{aligned} \underset{\{\mathbf{x}\},\mathbf{z}}{\text{minimize}} \quad & \sum_{i=1}^{N} f_i(\mathbf{x}_i) \\ \text{s.t.} \quad & \mathbf{x}_i - \mathbf{z} = \mathbf{0},\ i = 1, \cdots, n. \end{aligned} \tag{2.21}$$

This is sometimes referred to as global consensus optimization [70] as all the local $\mathbf{x}_i$'s should be equal to a global $\mathbf{z}$. Notice that (2.21) can be seen as a special case of (2.16) with $g(.) = 0$, $\mathbf{A} = \mathbf{I}$, $\mathbf{B} = -\mathbf{I}$, $\mathbf{y} = \mathbf{z}$, and $\mathbf{c} = \mathbf{0}$. In order to solve (2.21) using ADMM, we construct the augmented Lagrangian associated with the

problem as

$$\mathcal{L}_\rho(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) = \sum_{i=1}^{N} \left(f_i(\mathbf{x}_i) + \boldsymbol{\lambda}_i^T(\mathbf{x}_i - \mathbf{z}) + \frac{\rho}{2}\|\mathbf{x}_i - \mathbf{z}\|_2^2 \right).$$

The resulting ADMM after simplifications to eliminate the updates of the dummy variable $\mathbf{z}$ can be given by

$$\begin{aligned}
\mathbf{x}_i^{(k+1)} &= \underset{\mathbf{x}_i}{\operatorname{argmin}} \left(f_i(\mathbf{x}_i) + (\boldsymbol{\lambda}_i^{(k)})^T(\mathbf{x}_i - \bar{\mathbf{x}}^{(k)}) + \frac{\rho}{2}\|\mathbf{x}_i - \bar{\mathbf{x}}^{(k)}\|_2^2 \right), \\
\boldsymbol{\lambda}^{(k+1)} &= \boldsymbol{\lambda}^{(k)} + \rho \left(\mathbf{x}_i^{(k+1)} - \bar{\mathbf{x}}^{(k+1)} \right),
\end{aligned}$$

where $\bar{\mathbf{x}}^{(k)} = 1/N \sum_{i=1}^{N} \mathbf{x}_i^{(k)}$. In Part IV, we use the ADMM to solve a similar consensus problem for distributed sensor selection.

PART

II

MOBILE NETWORK LOCALIZATION

There is no royal road to Geometry.

EUCLID OF ALEXANDRIA

3

Dynamic Multidimensional Scaling for Low-Complexity Mobile Network Tracking

Abstract

Cooperative localization of mobile sensor networks is a fundamental problem which becomes challenging for anchorless networks where there is no pre-existing infrastructure to rely on. Two cooperative mobile network tracking algorithms based on novel dynamic multidimensional scaling (MDS) ideas are proposed. The algorithms are also extended to operate in partially connected networks. Compared with recently proposed algorithms based on the extended and unscented Kalman filter (EKF and UKF), the proposed algorithms have a considerably lower computational complexity. Furthermore, model-independence, scalability as well as an acceptable accuracy make our proposed algorithms a good choice for practical mobile network tracking.

3.1 Introduction

Cost and energy prohibitive global positioning systems (GPS) motivate researchers to focus on estimating the location of sensor nodes using their pairwise distances in a *cooperative* context [6]. Studies on cooperative network localization can be divided into two main categories, i.e., anchored and anchor-less localization. Anchored localization algorithms rely on distance measurements between the unknown-location nodes and the anchor nodes, whereas anchorless ones can work without such information and determine the *relative* location of the sensor nodes from pairwise distance measurements. Such a relative location map could for instance be useful to determine the distribution of the nodes, but other applications might require an additional relative or absolute frame of reference. One popular anchorless localization algorithm for a static network is classical multidimensional scaling (MDS) [72] or its distributed version [41].

Surprisingly, the problem of *cooperative network* localization for *mobile* sensor networks has not been efficiently solved yet. There are a lot of studies in the literature on single and multiple target tracking using the extended and unscented Kalman filter (EKF and UKF) as well as particle filters (PFs) [73]; however, they are mainly non-cooperative classical target tracking approaches. For *anchored* localization, studies in [14–17, 74] investigate the problem of localizing a mobile target or network using distance measurements in an MDS-based context. In [15], for instance, a Jacobian-like mobile network tracking algorithm is proposed by exploiting the Nyström approximation. However, this approach is non-cooperative.

On the other hand, in [7] an *anchorless* localization scheme for mobile network localization based on the theory of factor graphs is proposed in which each node requires knowledge about its own movement model as a probability distribution, which is not so simple to acquire in a real application. In [19, 20], cooperative network localization algorithms based on the EKF and the UKF are developed which incorporate the locations of the nodes as well as their velocities in a state-space model. Although velocity measurements of the nodes aid cooperative network localization, it requires the use of costly Doppler sensors, and hence, we avoid using it here. Inspired by the elegance of MDS localization, we propose to use two novel subspace tracking algorithms (Section 3.2) to track the variations in the signal eigenvectors and corresponding eigenvalues of the time-varying double-centered distance matrix. We show that this leads to a *dynamic MDS* paradigm which enables us to track the relative locations of a mobile network using only pairwise distance measurements. The absolute locations of the mobile nodes can then be recovered by the help of an absolute frame of reference provided by a few anchor nodes. In order to circumvent the limitations of the classical MDS, we then also propose an extension for partially connected mobile networks (Section 3.4). A detailed computational complexity analysis as well as the posterior Cramér-Rao bound (PCRB) derivation (Section 3.3) together with extensive simulation results (Section 3.5) illustrate that the proposed algorithms are scalable, acceptably accurate and have a much lower computational complexity compared to algorithms based on the EKF [20] and the UKF [19].

3.2 Dynamic Multidimensional Scaling

In this section, we formulate the problem of cooperative network localization and develop the dynamic MDS idea.

3.2.1 Problem Formulation

We consider a network of N mobile wireless sensor nodes, living in a D-dimensional space ($D < N$). Let $\mathbf{x}_{i,k}$ be the actual coordinate vector of the i-th sensor node at the k-th snapshot of the mobile network, or equivalently, let $\mathbf{X}_k = [\mathbf{x}_{1,k}, \ldots, \mathbf{x}_{N,k}]$ be the matrix of coordinates. Let us consider an environment with line-of-sight (LOS) conditions between the nodes and let us assume that time of flight (ToF) and/or received signal strength (RSS) information is already converted into noisy distance measurements as $r_{i,j,k} = d_{i,j,k} + v_{i,j,k}$ where $d_{i,j,k} = \|\mathbf{x}_{i,k} - \mathbf{x}_{j,k}\|$ is the noise-free Euclidean distance and $v_{i,j,k} \sim \mathcal{N}(0, \sigma^2_{v,i,j,k})$ is the additive white noise both at the k-th snapshot. The problem considered herein can now be stated as follows. Given the pairwise noisy distance measurements $r_{i,j,k}$ at each snapshot of the mobile network, determine the location of the mobile nodes and keep their track (up to a translation and orthogonal transformation). In case of a network with fixed nodes, the squared noisy distance measurements $r^2_{i,j,k}$ between the nodes can be collected in a distance matrix $\mathbf{D}_k$, i.e., $[\mathbf{D}_k]_{i,j} = r^2_{i,j,k}$, after which the double-centered distance matrix can be calculated as $\mathbf{B}_k = -1/2\mathbf{\Gamma}\mathbf{D}_k\mathbf{\Gamma}$ using the centering operator $\mathbf{\Gamma} = \mathbf{I}_N - \mathbf{1}_N\mathbf{1}_N^T/N$, where $\mathbf{I}_N$ denotes an $N \times N$ identity matrix and $\mathbf{1}_N$ represents an $N \times 1$ vector of all ones. For the k-th snapshot of the mobile network, the well-known MDS approach [14, 41, 72] then finds the locations as the solution to

$$\min_{\tilde{\mathbf{X}}} \|\mathbf{B}_k - \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\|_F^2,$$

where the minimum is taken over all $D \times N$ matrices $\tilde{\mathbf{X}}$ and $\|.\|_F$ denotes the Frobenius norm. The solution can be found by means of the eigenvalue decomposition (EVD) of $\mathbf{B}_k$ which can be expressed as

$$\mathbf{B}_k = \begin{bmatrix}\mathbf{U}_{1,k} & \mathbf{U}_{2,k}\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma}_{1,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{2,k}\end{bmatrix}\begin{bmatrix}\mathbf{U}_{1,k}^T \\ \mathbf{U}_{2,k}^T\end{bmatrix}, \tag{3.1}$$

where $\mathbf{U}_{1,k}$ and $\mathbf{U}_{2,k}$ respectively represent the $N \times D$ and $N \times (N-D)$ matrices containing the orthonormal eigenvectors corresponding to the signal and noise subspace of $\mathbf{B}_k$, and $\mathbf{\Sigma}_{1,k}$ and $\mathbf{\Sigma}_{2,k}$ respectively contain the eigenvalues corresponding to the signal and noise subspace. The MDS estimate of the location matrix up to a translation and orthogonal transformation can then be expressed as

$$\tilde{\mathbf{X}}_k = \mathbf{\Sigma}_{1,k}^{\frac{1}{2}}\mathbf{U}_{1,k}^T. \tag{3.2}$$

In the noiseless case, $\tilde{\mathbf{X}}_k = \mathbf{\Psi}\mathbf{X}_k\mathbf{\Gamma}$ where $\mathbf{\Psi}$ is an arbitrary orthogonal transformation and $\mathbf{\Gamma}$ translates the nodes such that their center of gravity is at the origin. Although the above procedure can be carried out for every snapshot of the mobile network, the complexity of computing the EVD in (3.1) can be quite intensive for large N [52]; especially when the nodes have to be monitored continuously. The idea behind the proposed dynamic MDS materialized by two subspace tracking algorithms is that in order to calculate the location of the nodes using (3.2), we only need to update the D signal eigenvectors in $\mathbf{U}_{1,k}$ and their corresponding eigenvalues in $\mathbf{\Sigma}_{1,k}$ [75, 76]. This can be done by more efficient iterative approaches as follows.

3.2.2 Perturbation Expansion-Based Subspace Tracking

In this subsection, we will present the perturbation expansion-based subspace tracking (PEST) algorithm. The idea is that in a mobile network the new location of a node can be considered as a perturbation of its previous location. Correspondingly, the double-centered distance matrix $\mathbf{B}_k$ can also be modeled as a perturbed version of $\mathbf{B}_{k-1}$ ($\mathbf{B}_k = \mathbf{B}_{k-1} + \Delta\mathbf{B}_k$). Now, if the movement of the nodes satisfies the property that the invariant subspace (here, the signal subspace) of the next (perturbed) double-centered distance matrix $\mathbf{B}_k$ does not contain any vectors that are orthogonal to the invariant subspace of the current $\mathbf{B}_{k-1}$, then the two bases respectively spanning the signal and noise subspace of the next double-centered distance matrix follow the expressions [50]

$$\tilde{\mathbf{U}}^u_{1,k} = \tilde{\mathbf{U}}_{1,k-1} + \tilde{\mathbf{U}}_{2,k-1}\mathbf{P}_k, \tag{3.3}$$

$$\tilde{\mathbf{U}}^u_{2,k} = -\tilde{\mathbf{U}}_{1,k-1}\mathbf{P}_k^T + \tilde{\mathbf{U}}_{2,k-1}, \tag{3.4}$$

where $\mathbf{P}_k$ is a coefficient matrix, $\tilde{\mathbf{U}}_{i,k}$ represents an orthonormal basis spanning the same subspace as the matrix of eigenvectors $\mathbf{U}_{i,k}$, and $\tilde{\mathbf{U}}^u_{i,k}$ is an unorthonormalized version of $\tilde{\mathbf{U}}_{i,k}$. Observe that in (3.3) and (3.4), different from the expressions in [50], we do not necessarily have the matrices of eigenvectors $\mathbf{U}_{i,k}$ on the right-hand side. In order to keep the computational complexity as low as possible, we will resort to a first-order approximation to compute $\mathbf{P}_k$. However, since we will continuously use first-order approximations, we cannot assume that $\tilde{\mathbf{U}}_{1,k-1}$ and $\tilde{\mathbf{U}}_{2,k-1}$ in (3.3) and (3.4) are orthonormal bases exactly spanning respectively the signal and noise subspaces of $\mathbf{B}_{k-1}$. And thus, the first-order approximation of $\mathbf{P}_k$ in [50] does not hold anymore, and we need to derive a new $\mathbf{P}_k$. The value of $\mathbf{P}_k$ should satisfy the necessary and sufficient condition for $\tilde{\mathbf{U}}^u_{1,k}$ and $\tilde{\mathbf{U}}^u_{2,k}$ to be bases

for the perturbed signal and noise subspaces. Thus, we need

$$\tilde{\mathbf{U}}_{2,k}^{u^T}\mathbf{B}_k\tilde{\mathbf{U}}_{1,k}^{u} = \mathbf{0}. \tag{3.5}$$

We can expand (3.5) by substituting (3.3) and (3.4) as follows

$$\begin{aligned}(-\tilde{\mathbf{U}}_{1,k-1}\mathbf{P}_k^T + \tilde{\mathbf{U}}_{2,k-1})^T\mathbf{B}_k(\tilde{\mathbf{U}}_{1,k-1} + \tilde{\mathbf{U}}_{2,k-1}\mathbf{P}_k) = \\ -\mathbf{P}_k\tilde{\mathbf{U}}_{1,k-1}^T\mathbf{B}_k\tilde{\mathbf{U}}_{1,k-1} - \mathbf{P}_k\tilde{\mathbf{U}}_{1,k-1}^T\mathbf{B}_k\tilde{\mathbf{U}}_{2,k-1}\mathbf{P}_k \\ +\tilde{\mathbf{U}}_{2,k-1}^T\mathbf{B}_k\tilde{\mathbf{U}}_{1,k-1} + \tilde{\mathbf{U}}_{2,k-1}^T\mathbf{B}_k\tilde{\mathbf{U}}_{2,k-1}\mathbf{P}_k = \mathbf{0}.\end{aligned} \tag{3.6}$$

Now by using $\mathbf{B}_k = \mathbf{B}_{k-1} + \Delta\mathbf{B}_k$ we can rewrite (3.6) as

$$\begin{aligned}-\underbrace{\mathbf{P}_k\tilde{\mathbf{U}}_{1,k-1}^T\mathbf{B}_k\tilde{\mathbf{U}}_{2,k-1}\mathbf{P}_k}_{\text{2nd order}} - \underbrace{\mathbf{P}_k\tilde{\mathbf{U}}_{1,k-1}^T\Delta\mathbf{B}_k\tilde{\mathbf{U}}_{1,k-1}}_{\text{2nd order}} +\tilde{\mathbf{U}}_{2,k-1}^T\Delta\mathbf{B}_k\tilde{\mathbf{U}}_{1,k-1} \\ + \underbrace{\tilde{\mathbf{U}}_{2,k-1}^T\Delta\mathbf{B}_k\tilde{\mathbf{U}}_{2,k-1}\mathbf{P}_k}_{\text{2nd order}} -\mathbf{P}_k\tilde{\mathbf{U}}_{1,k-1}^T\mathbf{B}_{k-1}\tilde{\mathbf{U}}_{1,k-1} \\ +\tilde{\mathbf{U}}_{2,k-1}^T\mathbf{B}_{k-1}\tilde{\mathbf{U}}_{1,k-1} + \tilde{\mathbf{U}}_{2,k-1}^T\mathbf{B}_{k-1}\tilde{\mathbf{U}}_{2,k-1}\mathbf{P}_k = \mathbf{0}.\end{aligned} \tag{3.7}$$

Note that for small perturbations, $\mathbf{P}_k$ in (3.7) will be close to a zero matrix. Thus, by neglecting the second-order terms, we obtain

$$\begin{aligned}-\mathbf{P}_k\tilde{\mathbf{U}}_{1,k-1}^T\mathbf{B}_{k-1}\tilde{\mathbf{U}}_{1,k-1} + \tilde{\mathbf{U}}_{2,k-1}^T\Delta\mathbf{B}_k\tilde{\mathbf{U}}_{1,k-1}+ \\ \underbrace{\tilde{\mathbf{U}}_{2,k-1}^T\mathbf{B}_{k-1}\tilde{\mathbf{U}}_{1,k-1}}_{\neq\mathbf{0}} + \underbrace{\tilde{\mathbf{U}}_{2,k-1}^T\mathbf{B}_{k-1}\tilde{\mathbf{U}}_{2,k-1}}_{\neq\mathbf{0}}\mathbf{P}_k = \mathbf{0}.\end{aligned} \tag{3.8}$$

Different from the derivations in [50], the third and fourth terms in (3.8) are close but not equal to zero due to the successive first-order approximations as explained earlier. It is notable that (3.8) is linear in the elements of $\mathbf{P}_k$ and can easily be solved w.r.t $\mathbf{P}_k$. However, this requires a $DN \times DN$ matrix inverse calculation which is undesirable due to its high complexity. Therefore, we confine our approximation of $\mathbf{P}_k$ to the first three terms in (3.8). By defining

$$\tilde{\mathbf{\Sigma}}_{1,k-1} = \tilde{\mathbf{U}}_{1,k-1}^T\mathbf{B}_{k-1}\tilde{\mathbf{U}}_{1,k-1}, \tag{3.9}$$

this results in

$$\mathbf{P}_k = \tilde{\mathbf{U}}_{2,k-1}^T\mathbf{B}_k\tilde{\mathbf{U}}_{1,k-1}\tilde{\mathbf{\Sigma}}_{1,k-1}^{-1}. \tag{3.10}$$

To avoid updating $\tilde{\mathbf{U}}^u_{2,k}$ in (3.3), we use the property that

$$\tilde{\mathbf{U}}_{1,k-1}\tilde{\mathbf{U}}^T_{1,k-1} + \tilde{\mathbf{U}}_{2,k-1}\tilde{\mathbf{U}}^T_{2,k-1} = \mathbf{I}_N.$$

Together with (3.10), this allows us to rewrite (3.3) as

$$\tilde{\mathbf{U}}^u_{1,k} = \tilde{\mathbf{U}}_{1,k-1} + (\mathbf{I} - \tilde{\mathbf{U}}_{1,k-1}\tilde{\mathbf{U}}^T_{1,k-1})\mathbf{B}_k\tilde{\mathbf{U}}_{1,k-1}\tilde{\mathbf{\Sigma}}^{-1}_{1,k-1}. \tag{3.11}$$

Now, to be able to use the above formula in an iterative fashion we can normalize it using any orthonormalization process like Gram-Schmidt (GS) factorization. We call the orthonormalized result $\tilde{\mathbf{U}}_{1,k}$. As can be seen from the above derivations, $\tilde{\mathbf{U}}_{1,k}$ is an approximation of the orthonormal basis which spans the same subspace as its corresponding signal eigenvectors in $\mathbf{U}_{1,k}$. However, to be able to calculate the relative locations using (3.2), we have to find the matrix of eigenvectors $\mathbf{U}_{1,k}$. To this aim, we look for a transformation matrix $\mathbf{A}_k$ to map $\tilde{\mathbf{U}}_{1,k}$ to $\mathbf{U}_{1,k}$ as follows

$$\tilde{\mathbf{U}}_{1,k} = \mathbf{U}_{1,k}\mathbf{A}_k. \tag{3.12}$$

Note that since $\tilde{\mathbf{U}}_{1,k}$ and $\mathbf{U}_{1,k}$ are isometries, $\mathbf{A}_k$ will be a unitary matrix according to the definition in (3.12). To be able to estimate the locations using (3.2), we also need to calculate $\mathbf{\Sigma}_{1,k}$, which depends on the value of $\mathbf{U}_{1,k}$ and $\mathbf{B}_k$ as $\mathbf{\Sigma}_{1,k} = \mathbf{U}^T_{1,k}\mathbf{B}_k\mathbf{U}_{1,k}$. From (3.9), and using (3.12), we finally obtain

$$\tilde{\mathbf{\Sigma}}_{1,k} = (\mathbf{U}_{1,k}\mathbf{A}_k)^T\mathbf{B}_k(\mathbf{U}_{1,k}\mathbf{A}_k) = \mathbf{A}^T_k\mathbf{\Sigma}_{1,k}\mathbf{A}_k. \tag{3.13}$$

From (3.13), $\mathbf{A}_k$ and $\mathbf{\Sigma}_{1,k}$ can be calculated by the EVD of $\tilde{\mathbf{\Sigma}}_{1,k}$. Note that, our main goal for using perturbation expansion was to avoid computationally intensive EVD calculations, while here we require it again. However, the point is that $\tilde{\mathbf{\Sigma}}_{1,k}$ is a $D \times D$ matrix (the number of dimensions D is in practice at most 3), which is very small in size compared to the $N \times N$ double-centered distance matrix $\mathbf{B}_k$ for large scale sensor networks. The overall PEST algorithm is summarized in Algorithm 3.1. Increasing the measurement interval decreases the computational cost but introduces larger perturbations, which leads to a degraded result. To heal this degradation, we can divide $\Delta\mathbf{B}_k$ in P proportional parts and run the PEST algorithm P times in each snapshot by successively applying these partial perturbations as shown by the following measurement update equation

$$\mathbf{B}_{k,p} = \mathbf{B}_{k-1} + p\frac{\Delta\mathbf{B}_k}{P}, \quad p = 1, \cdots, P.$$

We call this modified algorithm the Modified PEST.

Algorithm 3.1 PEST/PIST

Initialization: Start with an initial location guess

For $k = 1$ to K (movement steps)
- Calculate $\tilde{\mathbf{U}}^u_{1,k}$ using (3.11) (for PEST) or (3.14) (for PIST)
- GS orthonormalization $\tilde{\mathbf{U}}_{1,k} = \text{GS}(\tilde{\mathbf{U}}^u_{1,k})$
- Calculate $\tilde{\mathbf{\Sigma}}_{1,k}$, $\mathbf{A}_k$ and $\mathbf{\Sigma}_{1,k}$ using (3.9) and (3.13)
- Calculate $\mathbf{U}_{1,k}$ using (3.12)
- Location estimation using (3.2)

End

3.2.3 Power Iteration-Based Subspace Tracking

Power iterations can also be used to efficiently calculate an invariant subspace of a diagonalizable matrix (like $\mathbf{B}_k$) [52]. Power iterations are normally used in an iterative manner till an acceptable accuracy is reached. Depending on the initial guess, the number of iterations can be large, which in turn leads to a high computational complexity. Additionally, an inappropriate choice of the initial guess can sometimes lead to instability and divergence problems [52]. To avoid both problems (complexity and divergence) in mobile network localization, we propose to do just one iteration in each snapshot of the mobile network and use the previous estimate of the orthonormal basis as the initial guess for the next estimate. This leads to a scheme that tracks the desired invariant subspace in a similar fashion as PEST, and we call it power iteration-based subspace tracking (PIST). Here, instead of using (3.11) as for the PEST, an unorthonormalized version of $\tilde{\mathbf{U}}_{1,k}$ can be calculated using

$$\tilde{\mathbf{U}}^u_{1,k} = \mathbf{B}_k \tilde{\mathbf{U}}_{1,k-1}. \tag{3.14}$$

Note that the resulting $\tilde{\mathbf{U}}_{1,k}$ after orthonormalization will again be an orthonormal basis spanning the desired signal subspace. Thus, the same EVD calculations as in (3.13) for PEST are required to obtain the matrix of eigenvectors. The overall PIST algorithm is summarized in Algorithm 3.1. We emphasize that the proposed algorithms are *anchorless* in the sense that the relative position of the mobile nodes (also called network configuration in this context) can continuously be calculated without requiring any anchor nodes. However, determining the absolute location of the nodes (removing the unknown translation and orthogonal transformation) requires a coordinate system consisting of at least $D + 1$ anchor nodes with known locations. Hence, if recovering the absolute locations is also of interest, e.g., for comparison purposes, then a possible additional step can be implemented for every snapshot of the mobile network.

Algo.	**Multiplication**	**GS**	**SQRT**	**Matrix inverse**	**EVD**	**Total FLOPS**
PEST	$4N^2D + 3ND^2 + ND$	$1(N \times D)$	2 (Scalar)	1 $(D \times D)$	1 $(D \times D)$	$4DN^2 + (5D^2 + D)N + 2D^3 + 6D^2 + 24$
PIST	$2N^2D + 2ND^2 + ND$	1 $(N \times D)$	2 (Scalar)	-	1 $(D \times D)$	$4DN^2 + (5D^2 + D)N + D^3 + 24$
EKF [20]	$(D/2)N^5 + (5D^2/2 - D)N^4 + (12D^3 - 5D^2/2 + 2D)N^3 + (4D^2 - D/2 + 2)N^2 - 2N$	-	$DN(N-1)/2$ (Scalar)	2 $(2DN \times 2DN)$	-	$(D/2)N^5 + (5D^2/2 - D)N^4 + (28D^3 - 5D^2/2 + 2D)N^3 + (52D^2 + 11D/2 + 2)N^2 + (-6D - 2)N$
UKF [19]	$(7D/2)N^5 + (6D^2 - 3D + 1/2)N^4 + (16D^3/3 - 8D^2 + 25D/2 - 1)N^3 + (D^2 - 9D + 2)N^2 + (D - 3/2)N$	-	$2DN^3 - (2D - 1/2)N^2 - 1/2N$ (Scalar) and 1 $(2DN \times 2DN)$	1 $(N(N-1)/2 \times N(N-1)/2)$	-	$(1/8)N^6 + (7D/2 - 3/8)N^5 + (6D^2 - 3D + 19/8)N^4 + (16D^3/3 - 8D^2 + 73D/2 - 33/8)N^3 + (D^2 - 33D + 19/2)N^2 + (D - 3/2)N$

Table 3.1: Computational Complexity

3.3 Analysis of the Proposed Algorithms

In the following, we analyze the computational complexity and attainable accuracy of the algorithms under consideration.

3.3.1 Computational Complexity

We define the computational complexity as the number of operations required to estimate the locations for a single snapshot. For the sake of simplicity, we do not count the number of additions and subtractions, due to their negligible complexity in comparison with the other operations. Also, we consider the same complexity for multiplications and divisions, and hence, we present the sum of them as the number of floating point operations (FLOPS). The results of the complexity calculations for the PEST, the PIST, the EKF, and the UKF algorithms are summarized in Table 3.1. The last column in the table presents the maximum number of FLOPS. To calculate this value, we assume that Gauss-Jordan elimination requiring $N^3 + 6N^2$ multiplications is used to calculate the inverse of an $N \times N$ matrix. Further, we assume that Newton's method is used to calculate a scalar square root (SQRT) which requires 12 multiplications and a Cholesky decomposition is used to calculate a matrix square root which requires $2N^3/3$ multiplications for an $N \times N$ matrix [17]. Moreover, a GS orthonormalization process is considered which requires $2ND^2$ multiplications for an $N \times D$ matrix [77]. For a $D \times D$ matrix EVD computation, we consider a maximum number of D^3 multiplications [77]. As can be seen in the table, both PEST and PIST have a quadratic complexity in N while it is of order 5 and 6 in N for the EKF and UKF, respectively. As can be seen, the considerably lower computational complexity is a significant gain for the proposed algorithms, especially for large networks (large N). Finally, the Jacobian-like algorithm proposed in [15] although being non-cooperative approximately leads to a complexity order of 3 in N which is still one order of magnitude larger than our complexity. An advantage of this low complexity is that the central unit of our algorithm can simply be one of the nodes of the network.

3.3.2 Tracking Accuracy

To derive the tracking accuracy, let us assume that the nodes move according to the following state-space model

$$\mathbf{s}_k = \mathbf{\Phi}_k \mathbf{s}_{k-1} + \mathbf{w}_k + \mathbf{m}_{k-1}, \tag{3.15}$$

$$\mathbf{r}_k = \mathbf{h}(\mathbf{s}_k) + \mathbf{v}_k, \tag{3.16}$$

where

$$\mathbf{s}_k = [\mathbf{x}_{1,k}^T, \ldots, \mathbf{x}_{N,k}^T, \dot{\mathbf{x}}_{1,k}^T, \ldots, \dot{\mathbf{x}}_{N,k}^T]^T,$$

is a column vector of length $2DN$ containing the locations and velocities at the k-th snapshot, and

$$\mathbf{r}_k = [r_{1,2,k}, r_{1,3,k}, \ldots, r_{N-1,N,k}]^T,$$

is the column vector of pairwise distance measurements of length $N(N-1)/2$ at the k-th snapshot. Next, $\mathbf{h}(.)$ is a deterministic observation function which relates the locations of the nodes (inside $\mathbf{s}_k$) to their corresponding pairwise distances and $\mathbf{m}_{k-1}$ is an optional control input at the $(k-1)$-th snapshot [78]. Further, $\mathbf{w}_k$ and $\mathbf{v}_k$ are vectors with zero mean Gaussian entries with standard deviation (std) $\sigma_{w,k}$ and $\sigma_{v,i,j,k}$, respectively. For the sake of clarity, we denote the elements of the state vector as $\mathbf{s}_k = [\mathbf{s}_{l,k}^T, \mathbf{s}_{v,k}^T]^T$, where $\mathbf{s}_{l,k}$ of length DN represents the vectorized version of the locations and $\mathbf{s}_{v,k}$ of length DN represents the vectorized version of the corresponding velocities. The lower bound on the mean squared error (MSE) of estimation for any discrete-time nonlinear filtering problem can be computed via the posterior Cramér-Rao bound (PCRB) [79]. For our problem, i.e., estimating the locations in the state vector $\mathbf{s}_k$ using all the previous and current measurements $\mathbf{r}_0, \cdots, \mathbf{r}_k$, the lower bound on the MSE covariance matrix (matrix of the state error) of any unbiased estimator is given by

$$\mathbb{E}\{[\hat{\mathbf{s}}_k - \mathbf{s}_k][\hat{\mathbf{s}}_k - \mathbf{s}_k]^T\} \geqslant \mathbf{J}_k^{-1}, \tag{3.17}$$

where $\mathbb{E}(.)$ stands for statistical expectation and $\hat{\mathbf{s}}_k$ is the state estimate. The recursive PCRB derived in [79] for updating the posterior Fisher information matrix ($\mathbf{J}_k$) for our model expressed by (3.15)-(3.16) boils down to

$$\mathbf{J}_k = \left(\mathbf{Q}_k + \mathbf{\Phi}_k \mathbf{J}_{k-1}^{-1} \mathbf{\Phi}_k^T\right)^{-1} + [\nabla_{\mathbf{s}_k} \mathbf{h}(\mathbf{s}_k)]^T \mathbf{R}_k^{-1} [\nabla_{\mathbf{s}_k} \mathbf{h}(\mathbf{s}_k)], \tag{3.18}$$

where $\mathbf{Q}_k$ and $\mathbf{R}_k$ respectively represent the exact covariance matrices of the process (movement) and measurement noise $\mathbf{w}_k$ and $\mathbf{v}_k$, and the gradient $\nabla_{\mathbf{s}_k} \mathbf{h}(\mathbf{s}_k)$ should be calculated using the true locations. Since we basically estimate the locations of the nodes and not their velocities, the PCRB of our location estimates is given by

$$\text{PCRB}_k = \sum_{i=1}^{DN} [\mathbf{J}_k^{-1}]_{i,i}, \tag{3.19}$$

which we average over different Monte Carlo (MC) realizations of the movement process. It is worth mentioning that the MSE of our location estimates will corre-

spond to the errors on the absolute locations and not on those up to a translation and orthogonal transformation. As mentioned earlier, the absolute locations can be recovered by considering l anchor nodes with known locations. Now, if we compute (3.18) for the location estimates of our anchorless network, $\nabla_{\mathbf{s}_k}\mathbf{h}(\mathbf{s}_k)$ and correspondingly $[\nabla_{\mathbf{s}_k}\mathbf{h}(\mathbf{s}_k)]^T\mathbf{R}_k^{-1}[\nabla_{\mathbf{s}_k}\mathbf{h}(\mathbf{s}_k)]$ will be rank deficient with a rank of at most $DM - D - 1$ (due to the unknown translation and orthogonal transformation in every snapshot). To resolve this problem, we try to obtain a bound by reformulating (3.18) for a network with l anchor nodes, and modify the process and measurement models as

$$\bar{\mathbf{s}}_k = \bar{\mathbf{\Phi}}_k\bar{\mathbf{s}}_{k-1} + \bar{\mathbf{w}}_k + \bar{\mathbf{m}}_{k-1}, \tag{3.20}$$

$$\bar{\mathbf{r}}_k = \mathbf{h}(\bar{\mathbf{s}}_k) + \bar{\mathbf{v}}_k, \tag{3.21}$$

where $\bar{\mathbf{s}}_k$, $\bar{\mathbf{w}}_k$ and $\bar{\mathbf{m}}_{k-1}$ are $2D(N-l)\times 1$ vectors calculated by removing the elements corresponding to the locations and velocities of the anchors from $\mathbf{s}_k$, $\mathbf{w}_k$ and $\mathbf{m}_{k-1}$, respectively. Therefore, $\bar{\mathbf{\Phi}}_k$ will be a $2D(N-l)\times 2D(N-l)$ matrix relating the previous modified state vector $\bar{\mathbf{s}}_{k-1}$ to the next one $\bar{\mathbf{s}}_k$. For the modified measurement model, $\bar{\mathbf{r}}_k$ is an $(N(N-1)/2 - |\mathbf{\Omega}|)\times 1$ vector similar to $\mathbf{r}_k$ but the noisy distance measurements ($r_{i,j,k}$) between the l anchors are removed ($|.|$ denotes the cardinality). The indices of the removed distance measurements are contained in

$$\mathbf{\Omega} = \{(i-1)N - (i(i-1)/2) + 1, \cdots, \\ (i-1)N - (i(i-1)/2) + l - i \mid i = 1, 2, \cdots, l-1\}.$$

The modified sequence of the posterior FIM can then be obtained as

$$\bar{\mathbf{J}}_k = \left(\bar{\mathbf{Q}}_k + \bar{\mathbf{\Phi}}_k\bar{\mathbf{J}}_{k-1}^{-1}\bar{\mathbf{\Phi}}_k^T\right)^{-1} + [\nabla_{\bar{\mathbf{s}}_k}\mathbf{h}(\bar{\mathbf{s}}_k)]^T\bar{\mathbf{R}}_k^{-1}[\nabla_{\bar{\mathbf{s}}_k}\mathbf{h}(\bar{\mathbf{s}}_k)],$$

where $\bar{\mathbf{Q}}_k$ is the $2D(N-l)\times 2D(N-l)$ process noise covariance matrix corresponding to $\bar{\mathbf{w}}_k$. The modified measurement noise covariance matrix $\bar{\mathbf{R}}_k$ will be a $(N(N-1)/2 - |\mathbf{\Omega}|)\times(N(N-1)/2 - |\mathbf{\Omega}|)$ diagonal matrix similar to $\mathbf{R}_k$ but corresponding to $\bar{\mathbf{v}}_k$. Further, $\nabla_{\bar{\mathbf{s}}_k}\mathbf{h}(\bar{\mathbf{s}}_k)$ is a $(N(N-1)/2 - |\mathbf{\Omega}|)\times 2D(N-l)$ matrix similar to $\nabla_{\mathbf{s}_k}\mathbf{h}(\mathbf{s}_k)$ but it is calculated by taking partial derivatives from the remaining $N(N-1)/2 - |\mathbf{\Omega}|$ distance measurements with respect to the $2D(N-l)$ elements in the modified state vector $\bar{\mathbf{s}}_k$.

3.4 Extension to Partially Connected Networks

The derivations of the proposed algorithms in Section 3.2 are based on the assumption that all the pairwise distance measurements are available. However, this assumption is not valid for many practical mobile scenarios where the nodes only have a limited communication range. Therefore, we also consider a simple finite-range model where the distances can be measured only if they are below a certain communication range r_0, otherwise they cannot be measured and they are considered *missing links*. To tackle this problem, there has been a lot of research in the literature to reconstruct the squared distance matrix $\mathbf{D}_k$ or correspondingly its double-centered version $\mathbf{B}_k$ by exploiting their specific properties like rank and inertia [80, 81]. However, we are interested in a low-complexity algorithm which also fits to our proposed dynamic MDS model. To this aim, we propose to include an additional inner iterative procedure (iterating P times in each snapshot) to account for the missing links. In each snapshot, we first construct $\hat{\mathbf{D}}_k$ from the measured noisy $\mathbf{D}_k$ as

$$[\hat{\mathbf{D}}_k]_{i,j} = \begin{cases} [\mathbf{D}_k]_{i,j}, & (i,j) \text{ measured}, \\ [\hat{\mathbf{D}}_{k-1}]_{i,j}, & (i,j) \text{ missing } \& \ [\hat{\mathbf{D}}_{k-1}]_{i,j} > r_0^2 \\ r_0^2, & (i,j) \text{ missing } \& \ [\hat{\mathbf{D}}_{k-1}]_{i,j} \leqslant r_0^2 \end{cases} \tag{3.22}$$

where the link between nodes i and j is denoted by (i,j). As is clear from (3.22), we fill the missing links with their corresponding previously recovered distance estimates, if their value is larger than r_0; otherwise we just fill the missing links with r_0 since we know that they should be larger than r_0. We then use the modified squared distance matrix $\hat{\mathbf{D}}_k$ to calculate $\hat{\mathbf{B}}_k$ which we feed to the PEST or the PIST to calculate the signal eigenvectors and corresponding eigenvalues. Then the relative locations of the nodes are used to recalculate a new set of pairwise distances and to construct a temporary squared distance matrix $\mathbf{E}_k$ similar to $\hat{\mathbf{D}}_k$. Then, we modify $\hat{\mathbf{D}}_k$ by updating the distances corresponding to the missing links from the recently calculated $\mathbf{E}_k$ as

$$[\hat{\mathbf{D}}_k]_{i,j} = \begin{cases} [\hat{\mathbf{D}}_k]_{i,j}, & (i,j) \text{ measured} \\ [\hat{\mathbf{D}}_{k-1}]_{i,j} + \rho \left([\mathbf{E}_k]_{i,j} - [\hat{\mathbf{D}}_{k-1}]_{i,j} \right), & (i,j) \text{ missing} \end{cases} \tag{3.23}$$

where $\rho \in (0,1]$ is a smoothing gain. This gain avoids divergence of the algorithm for cases where the signal subspace is affected due to a large number of missing links. Now, a new $\hat{\mathbf{B}}_k$ can be calculated from the recently updated $\hat{\mathbf{D}}_k$ which can be used for the next (inner) iteration in the same snapshot. The final $\hat{\mathbf{D}}_k$ from the inner loop will be transferred to the next snapshot. The modified iterative algo-

Algorithm 3.2 Extension to Partially Connected Networks

Initialization: Start with an initial location guess
For $k = 1$ to K (movement steps)
  Construct $\hat{\mathbf{D}}_k$ from $\mathbf{D}_k$ using (3.22)
  Calculate $\hat{\mathbf{B}}_k$ from $\hat{\mathbf{D}}_k$
    For $p = 1$ to P
    Use PEST/PIST to estimate locations from $\hat{\mathbf{B}}_k$
    Calculate new pairwise distances and construct $\mathbf{E}_k$
    Update $\hat{\mathbf{D}}_k$ using (3.23)
    Calculate a new $\hat{\mathbf{B}}_k$ for the next (inner) iteration
  End
End

rithm for partially connected networks is shown in Algorithm 3.2. Note that these P inner iterations scale the computational complexity of the algorithms by at most a factor P. Since in practice $P \leqslant 10$ this does not increase the order of complexity of the modified algorithms for networks with $N > 10$. It is noteworthy that different from ranging, communication between each node and the central unit can be accomplished by multi-hop communications.

3.5 Simulation Results

We consider a network of N mobile sensors, living in a two-dimensional space ($D = 2$). The mobile nodes are considered to be moving inside a bounded area of 100×100 squared meters determined by its vertices located at $(0, 0)$m, $(0, 100)$m, $(100, 0)$m and $(100, 100)$m. Note that our proposed algorithms are blind to the movement model, but for the sake of comparison we consider a modified random walk process where $\mathbf{\Phi} = \mathbf{I}_{4N} + \mathbf{F}T_s$, with T_s the measurement interval and $\mathbf{F} = [\mathbf{0}_{2N\times 2N},\ \mathbf{I}_{2N\times 2N}; \mathbf{0}_{2N\times 2N},\ \mathbf{0}_{2N\times 2N}]$. We set $\mathbf{w}_k = [\mathbf{0}^T, \breve{\mathbf{w}}_k^T]^T$, where we assume that $\breve{\mathbf{w}}_k$ is a vector with i.i.d. zero-mean Gaussian entries with std σ_w. This movement model does not guarantee that the mobile nodes stay inside the bounded area. To make this happen without greatly violating the predefined movement model in favor of the model-based algorithms (the EKF and the UKF), we propose to slightly change the movement pattern so that each time a node gets closer than a threshold ($d_0 = 5$m) to the borders of the covered area, we gradually decrease the velocity of that particular node with a centripetal force. The center of the area is $\mathbf{c} = (50, 50)$m. Let us define the $2N \times 2N$ diagonal matrix

$\mathbf{G}_{k-1} = \text{diag}(\mathbf{g}_{k-1})$, where $\mathbf{g}_{k-1}$ is given by

$$[\mathbf{g}_{k-1}]_i = \begin{cases} 0, & [\mathbf{c}]_1 - |[\mathbf{s}_{l,k-1}]_{2i-1} - [\mathbf{c}]_1| < d_0 \\ \left[\frac{[\mathbf{c}]_1 - |[\mathbf{s}_{l,k-1}]_{2i-1} - [\mathbf{c}]_1|}{[\mathbf{c}]_1}\right]^{\frac{1}{\alpha}}, & [\mathbf{c}]_1 - |[\mathbf{s}_{l,k-1}]_{2i-1} - [\mathbf{c}]_1| \geqslant d_0 \\ 0, & [\mathbf{c}]_2 - |[\mathbf{s}_{l,k-1}]_{2i} - [\mathbf{c}]_2| < d_0 \\ \left[\frac{[\mathbf{c}]_2 - |[\mathbf{s}_{l,k-1}]_{2i} - [\mathbf{c}]_2|}{[\mathbf{c}]_2}\right]^{\frac{1}{\alpha}}, & [\mathbf{c}]_2 - |[\mathbf{s}_{l,k-1}]_{2i} - [\mathbf{c}]_2| \geqslant d_0 \end{cases}$$

with $i = 1, 2, \cdots, N$. This equation investigates whether or not the nodes are closer to the borders than the threshold. Now, the velocity of the nodes in the next step will be computed as

$$\mathbf{s}_{v,k} = \mathbf{G}_{k-1}\mathbf{s}_{v,k-1} + \mathbf{G}_{k-1}\breve{\mathbf{w}}_k + \underbrace{\sigma_w[\mathbf{I}_{2N} - \mathbf{G}_{k-1}]\left(\frac{[\mathbf{c}]_1\mathbf{1}_{2N} - \mathbf{s}_{l,k-1}}{\|[\mathbf{c}]_1\mathbf{1}_{2N} - \mathbf{s}_{l,k-1}\|}\right)}_{\breve{\mathbf{m}}_{k-1}}, \tag{3.24}$$

where the third term $\breve{\mathbf{m}}_{k-1}$ is the $2N \times 1$ non-zero vector in the optional control input ($\mathbf{m}_{k-1} = [\mathbf{0}^T \breve{\mathbf{m}}_{k-1}^T]^T$) which imposes a centripetal force directed towards the center of the area $\mathbf{c}$. Note that the elements of the $\mathbf{G}_{k-1}$ matrix are 0 for the nodes that have passed the threshold and therefore only the third term pulls them back into the area. For those nodes that have not passed the threshold, the elements of $\mathbf{G}_{k-1}$ are close to 1 since α is chosen to be a large integer $10 < \alpha < 20$, and therefore (3.24) acts very close to the classical random walk ($\mathbf{s}_{v,k} = \mathbf{s}_{v,k-1} + \breve{\mathbf{w}}_k$) for those nodes. Inspired by the CRB for range estimation in additive white Gaussian noise, following [16, 74], for a realistic free-space model we introduce a constant $\gamma = d_{i,j,k}^2/\sigma_{v,i,j,k}^2$ which punishes the longer distances with larger measurement errors. For a quantitative comparison, we consider the positioning root mean squared error (PRMSE) of the algorithms at the k-th snapshot, which is defined by

$$\text{PRMSE}_k = \sqrt{\frac{\sum_{m=1}^{M}\sum_{n=1}^{N} e_{n,m,k}^2}{M}}, \tag{3.25}$$

where $e_{n,m,k}$ represents the distance between the real location of the n-th node and its estimated location at the m-th MC trial of the k-th snapshot. All simulations are averaged over $M = 100$ independent MC runs where in each run the nodes move in random directions starting from random initial locations. For the sake of comparison, we also simulate the cooperative network localization method of [20] based on the EKF and also the algorithm in [19] based on the UKF modified to our setup. Fig. 3.1 illustrates a realization of the mobile network ($N = 3$) were for

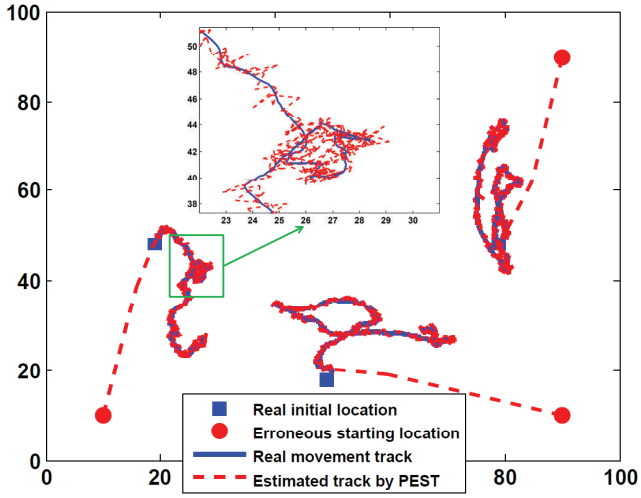


Figure 3.1: Tracking of a single realization from erroneous initial locations

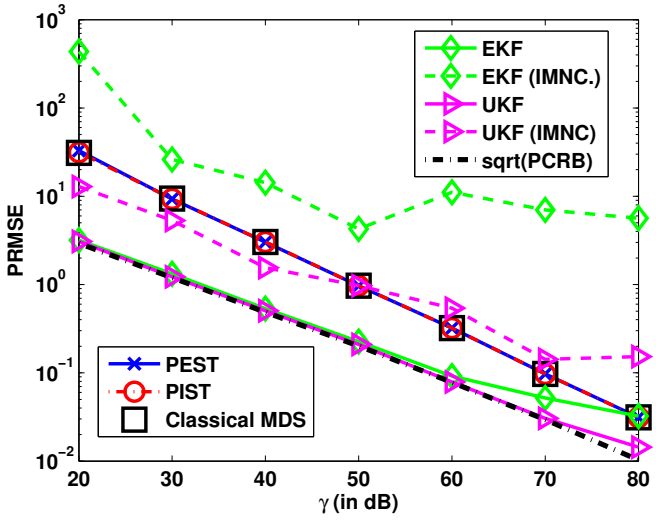


Figure 3.2: PRMSE performance for $T_s = 0.1$s

the sake of clarity only the PEST is plotted (we show in the following simulations that both algorithms have very close performances). For all simulations, to be able to plot and/or evaluate the results based on the absolute locations, we resolve the unknown translation and orthogonal transformation of our location estimates by considering $l = 3$ anchor nodes. In general, for all the simulations, we initialize the

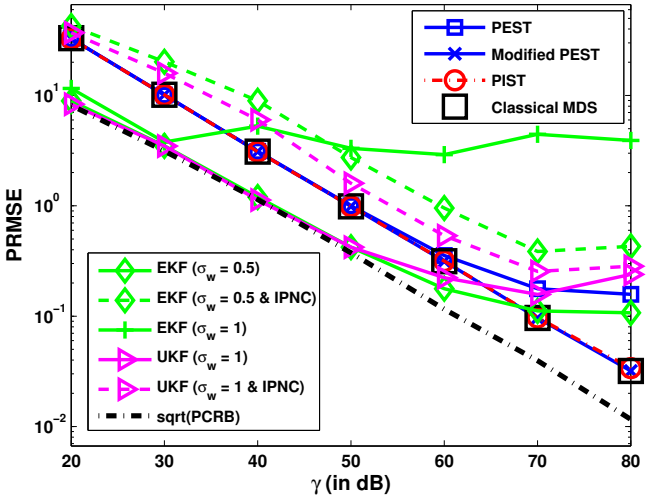


Figure 3.3: PRMSE performance for $T_s = 1$s

algorithms with random erroneous initializations. Here, for the sake of visibility, we initialize the algorithm close to the borders of the covered area, which is far from the real initial locations. As is clear, convergence is a matter of a few steps. During our simulations we observed that random initializations lead to divergence of the EKF in many of its runs, while the UKF and (even better) our proposed algorithms are robust against erroneous initializations.

Fig. 3.2 shows the PRMSE performance of the algorithms vs. γ for $N = 10$, $T_s = 0.1$s, $\sigma_w = 0.1$ and at the snapshot $k = 250$ where all the algorithms have converged. We also plot the performance of classical MDS and the derived PCRB as the performance bounds of the algorithms. From the figure, the PEST and the PIST perform very close to each other and attain the classical MDS performance while they are much more computationally efficient. The EKF performs better than the proposed algorithms in terms of accuracy, and the UKF is even better than the EKF (closer to the PCRB) but they both come at the price of a much higher complexity and depend on the information about the process and measurement models. That is why if we feed both the EKF and UKF with imperfect measurement noise covariance (IMNC) information (here, $\mathbf{R}_k$ with 40% error), the EKF diverges drastically while the UKF degrades and performs worse than the proposed algorithms for $\gamma > 50$dB. Beyond the computational efficiency, this is another advantage of our proposed *model-independent* algorithms over model-based ones (the EKF and the UKF). Fig. 3.3 depicts the same results as Fig. 3.2 ($N = 10$) but for $T_s = 1$s

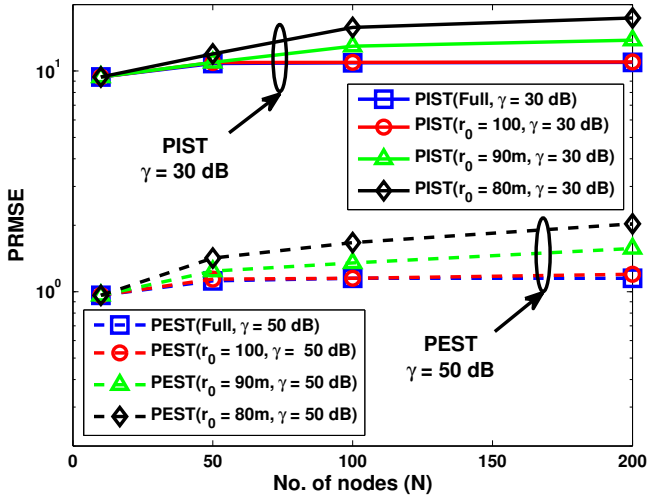


Figure 3.4: Partial connectivity and scalability

and $\sigma_w = 0.5$ and 1. Increasing σ_w boosts the effect of increasing T_s. From the figure, by increasing T_s and σ_w the EKF diverges drastically even with perfect model information while in a similar situation the UKF is just degraded for $\gamma > 50$dB. The PIST performs superb and the PEST is a little bit degraded for $\gamma > 60$dB, which can be healed by using the Modified PEST as explained in Subsection 3.2.2. Again we investigate the model-dependency of the EKF and the UKF by feeding them with imperfect process noise covariance (IPNC) information, e.g., a scaled σ_w is adopted here. The results are interesting since both algorithms degrade significantly and perform worse than the proposed algorithms for all γ. Notably, the UKF is much more robust against an increase of T_s, and the proposed algorithms are even more *robust* than the UKF and this makes them cost-efficient algorithms for practical scenarios.

Finally, Fig. 3.4 investigates two important issues, i.e., scalability and tackling partial connectivity. For the sake of clarity, we plot the performance of the PIST for $\gamma = 30$dB and the one of the PEST for $\gamma = 50$dB both for $T_s = 0.1$s. From the figure, the performance of the algorithms in fully connected networks remains almost the same with increasing the size of the network up to $N = 200$ (i.e., *scalability*). For partially connected networks, we decrease r_0 from the maximum distance in the network $r_{\max} = 100\sqrt{2} \approx 141$m to $r_0 = 100$m, 90m and 80m. As can be seen, the performance of the algorithms in partially connected networks (for $r_0 < 100$m) gradually deviates from that of the fully connected network. In our simulations,

we observe that $r_0 = 100$m and $r_0 = 80$m approximately correspond to respectively 30% and 50% misconnectivity in the network which is considerable. Note that decreasing r_0 further leads to many possible configurations which are not rigid anymore and thus in principle there will be no solution for the reconstruction problem. This might lead to large estimation errors by our algorithm in cases where the signal subspace is badly damaged due to the large number of missing links.

3.6 Conclusions

We have proposed two cooperative mobile network tracking algorithms based on a novel dynamic MDS. We have also extended the proposed algorithms to operate in more realistic partially connected networks. The proposed algorithms are model-independent. It has been shown that the proposed algorithms are characterized by a low computational complexity, an acceptable accuracy, and robustness against the measurement interval of the network, which makes them a superb choice for practical implementations. As a future work, we will explore a distributed implementation of the proposed algorithms.

4

Cooperative Localization In Partially Connected Mobile Wireless Sensor Networks Using Geometric Link Reconstruction

Abstract

We extend one of our recently proposed anchorless mobile network localization algorithms (called PEST) to operate in a partially connected network. To this aim, we propose a geometric missing link reconstruction algorithm for noisy scenarios and repeat the proposed algorithm in a local-to-global fashion to reconstruct a complete distance matrix. This reconstructed matrix is then used in the PEST to localize the mobile network. We compare the computational complexity of the new link reconstruction algorithm with existing related algorithms and show that our proposed algorithm has the lowest complexity, and hence, is the best extension of the low complexity PEST. Simulation results further illustrate that the proposed link reconstruction algorithm leads to the lowest reconstruction error as well as the most accurate network localization performance.

4.1 Introduction

Numerous applications of wireless sensor networks (WSNs) cannot rely on a pre-existing and fixed infrastructure. In such scenarios, there are typically no anchor nodes (with known locations) and determining the relative location of the sensor nodes is the ultimate goal. The problem of localization in anchorless networks becomes more challenging when the nodes of the network are mobile. In [18] an anchorless localization scheme for mobile networks is proposed wherein each node requires knowledge about its own movement model as a probability distribution in order to do predictions, which is not so simple to acquire and additionally increases the computational complexity significantly. In [20], a method based on extended Kalman filtering is developed which incorporates the locations of the nodes as well as their velocities in a state-space model. But, this algorithm also

has a high complexity. In [75], we proposed two anchorless network localization algorithms using novel subspace tracking ideas to adapt the classical multidimensional scaling (MDS) [82] for mobile WSNs. The proposed model-independent algorithms (PEST and PIST) have a considerably lower complexity than existing algorithms as well as an acceptable accuracy. Surprisingly, the problem of partial connectivity in not well investigated in a mobile WSN.

In this chapter, we propose to use a local-to-global missing link reconstruction to end up with a reconstructed network distance measurement matrix which can be fed to the PEST algorithm for localization. To this aim, we modify an existing link reconstruction algorithm [81], modify the Nyström algorithm [82] for link reconstruction, and also propose a novel geometric missing link reconstruction algorithm and modify it by proposing a selection criterion for noisy measurements. The rest of this chapter is organized as follows. In Section 4.2, we present the network model and state the problem under consideration. Section 4.3 tackles the problem of partial connectivity. Section 4.4 compares the computational complexity of the missing link reconstruction algorithms under consideration. Section 4.5 provides simulation results for evaluation of missing link reconstruction as well as mobile localization in a partially connected WSN. Finally, concluding remarks are presented in Section 4.6.

4.2 Network Model and Problem Statement

We consider a network of N mobile wireless sensor nodes, living inside a bounded 2-dimensional space. Our network model is based on pairwise distance measurements and these distance measurements themselves can be calculated by means of time of flight (ToF) measurements. Hence, we assume that the ToF information is already converted into noisy distance measurements as

$$r_{i,j,k} = d_{i,j,k} + v_{i,j,k}, \tag{4.1}$$

where $d_{i,j,k} = \|\mathbf{x}_{i,k} - \mathbf{x}_{j,k}\|$ is the noise-free Euclidean distance, $v_{i,j,k} \sim \mathcal{N}(0, \sigma^2_{v,i,j,k})$ is the uncorrelated additive noise and $\mathbf{x}_{i,k}$ is the actual coordinate vector of the i-th sensor node, all for the k-th snapshot of a mobile scenario. For a free space propagation model, we consider a constant

$$\gamma = d^2_{i,j,k}/\sigma^2_{v,i,j,k}, \tag{4.2}$$

which punishes the longer distances with larger measurement errors. Meanwhile, we consider a simple finite-range model where the distances can be measured only

if they are below a certain communication range r_0, otherwise they cannot be measured and we call them *missing links*. A wide variety of movement models can be considered for the mobile nodes since in [75] we explain that the proposed algorithms, one of which is also considered here, are blind to the movement model. The problem considered herein can be stated as follows. Having a fully connected network, the squared noisy distance measurements $r^2_{i,j,k}$ between the nodes can be collected in a distance matrix $\mathbf{D}_k$, i.e., $[\mathbf{D}_k]_{i,j} = r^2_{i,j,k}$, after which the double-centered distance matrix can be calculated as $\mathbf{B}_k = -1/2\mathbf{H}_N\mathbf{D}_k\mathbf{H}_N$ using the centering operator $\mathbf{H}_N = \mathbf{I}_N - \mathbf{1}_N\mathbf{1}_N^T/N$, where $\mathbf{I}_N$ denotes an $N \times N$ identity matrix and $\mathbf{1}_N$ represents an $N \times 1$ vector of all ones. Then, $\mathbf{B}_k$ can be used in the PEST to track the locations of the nodes in an iterative manner [75]. However, unlike [75], we here consider a partially connected network. To be able to modify our previously proposed PEST algorithm to operate in partially connected networks, we propose to recover the missing links in a local-to-global fashion and then use the PEST. As we use the PEST, the network localization will be anchorless.

4.3 Tackling Partial Connectivity

We first consider the problem of missing link reconstruction, which is then used in a local-to-global fashion to reconstruct $\mathbf{D}_k$.

4.3.1 Missing Link Reconstruction

In [83], a distributed algorithm for anchorless localization based on building a relative coordinate system is explained. For every node of the network a relative coordinate system is considered which is used to localize the neighboring nodes. We will here exploit this idea to reconstruct missing links in our mobile network. We propose to build a local coordinate system only around 5 nodes including 3 interconnected nodes (N_1 to N_3) and 2 other nodes (N_4 and N_5) which are both connected to the first three and the link between the last two nodes is missing as shown in Fig. 4.1. Let us start with the noiseless case. We choose one of the first three nodes as N_1 and place it on the origin of the local coordinate system $[0,\ 0]^T$. Since we know $d_{1,2}$, we can set the coordinates of N_2 to $[d_{1,2},\ 0]^T$. Now, by calculating $\cos(\alpha_3)$ using

$$\cos(\alpha_3) = \frac{d^2_{1,2} + d^2_{1,3} - d^2_{2,3}}{2d_{1,2}d_{1,3}}, \tag{4.3}$$

the location of N_3 will then be $[d_{1,3}\cos(\alpha_3),\ d_{1,3}\sqrt{1-\cos(\alpha_3)^2}]^T$ or $[d_{1,3}\cos(\alpha_3),\ -d_{1,3}\sqrt{1-\cos(\alpha_3)^2}]^T$, but we set it to the former. In order to acquire a rigid

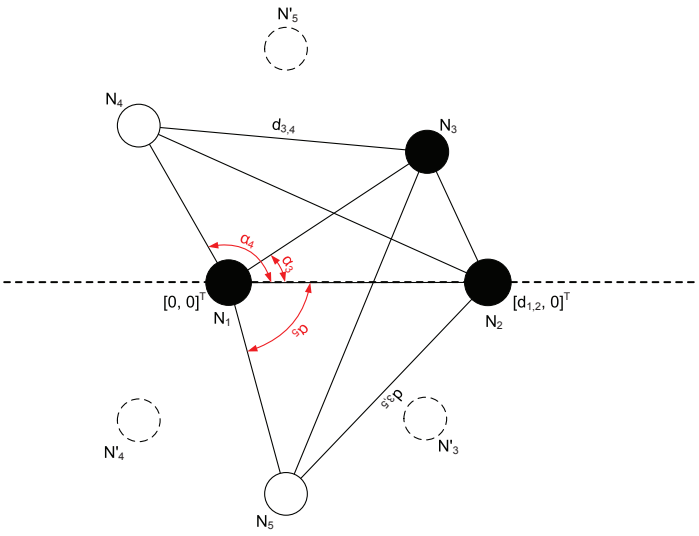


Figure 4.1: Geometric link reconstruction (GLR)

configuration (up to a translation and orthogonal transformation) we calculate the two possible locations for N_4 (also N_5) similar to N_3 and decide between the two possible locations by comparing the distances $d(N_4, N_3)$ and $d(N_4', N_3)$ with the available measured $d_{3,4}$ and choose the one which is equal to it. For a noisy scenario, however, we will have to choose the location which yields a closer distance compared to the noisy measured $r_{3,4}$. The same explanations hold for N_5. Now, having the relative location of N_4 and N_5 in the considered coordinate system we can calculate their missing distance. We call this algorithm geometric link reconstruction (GLR). Note that considering the above explanations, this 5-node setup is the simplest configuration of nodes with unknown locations (fits in anchorless network localization) by means of which we can recover one missing link.

For the case of noisy measurements, however, we expect that the accuracy of our relative location estimates for N_4 and N_5 will depend on the choice of the base-line nodes N_1 and N_2. For the sake of simplicity, let us assume that N_2 is already perfectly located using the available information. Further, the location estimation error in both N_4 and N_4' is similar with respect to the base-line since N_3 is only used to choose N_4 or N_4'. Therefore, the Cramér-Rao bound (CRB) of our location estimate will depend on the measurement vector $\mathbf{r} = [r_{1,4},\ r_{2,4}]^T$, where $r_{i,4} = \sqrt{(x_4 - x_i)^2 + (y_4 - y_i)^2}$. Under the above assumptions, the CRB of the N_4 location estimate for general Gaussian noise can be derived using the Fisher

information matrix (FIM) as explained in [78]

$$\mathbf{I}(N_4) = \begin{bmatrix} (\frac{\partial \mathbf{r}}{\partial x_4})^T \mathbf{C}^{-1} (\frac{\partial \mathbf{r}}{\partial x_4}) & (\frac{\partial \mathbf{r}}{\partial x_4})^T \mathbf{C}^{-1} (\frac{\partial \mathbf{r}}{\partial y_4}) \\ (\frac{\partial \mathbf{r}}{\partial y_4})^T \mathbf{C}^{-1} (\frac{\partial \mathbf{r}}{\partial x_4}) & (\frac{\partial \mathbf{r}}{\partial y_4})^T \mathbf{C}^{-1} (\frac{\partial \mathbf{r}}{\partial y_4}) \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \mathrm{tr}[\mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial x_4} \mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial x_4}] & \mathrm{tr}[\mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial x_4} \mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial y_4}] \\ \mathrm{tr}[\mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial y_4} \mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial x_4}] & \mathrm{tr}[\mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial y_4} \mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial y_4}] \end{bmatrix}. \tag{4.4}$$

For distance-dependent measurement noise (as defined by (4.1) and (4.2)), the covariance matrix of the measurements $\mathbf{C}$ will be

$$\mathbf{C} = \mathbb{E}\{(\mathbf{r} - \mathbb{E}\{\mathbf{r}\})(\mathbf{r} - \mathbb{E}\{\mathbf{r}\})^T\} = \begin{bmatrix} \frac{d_{1,4}^2}{\gamma} & 0 \\ 0 & \frac{d_{2,4}^2}{\gamma} \end{bmatrix}. \tag{4.5}$$

Our derivations show that the second term of (4.4) is independent of γ and is negligible compared to the first term for large values of γ. Thus, the FIM can be approximated by the first term of (4.4) as

$$\mathbf{I}(N_4) \approx \gamma \begin{bmatrix} (\frac{x_4^2}{d_{1,4}^4}) + \frac{(x_4 - d_{1,2})^2}{d_{2,4}^4} & y_4(\frac{x_4}{d_{1,4}^4} + \frac{x_4 - d_{1,2}}{d_{2,4}^4}) \\ y_4(\frac{x_4}{d_{1,4}^4} + \frac{x_4 - d_{1,2}}{d_{2,4}^4}) & y_4^2(\frac{1}{d_{1,4}^4} + \frac{1}{d_{2,4}^4}) \end{bmatrix}. \tag{4.6}$$

Now, by considering the configuration shown in Fig. 4.1, the CRB after elaborate simplifications can be given by

$$\mathrm{CRB}_{N_4} \approx \frac{(d_{1,4}^2 + d_{2,4}^2)(d_{1,4}^2 d_{2,4}^2)}{4\gamma A_{(N_1,N_2,N_4)}^2}, \tag{4.7}$$

where $A_{(N_1,N_2,N_4)}$ indicates the area of the triangle with vertices N_1, N_2 and N_4. The same calculations can be carried out for the case of distance-independent measurement noise with $v_{i,j} \sim \mathcal{N}(0, \sigma_v^2)$. For that case, $\mathbf{C} = \sigma_v^2 \mathbf{I}_2$ and the second term of (4.4) will be zero, and therefore, the CRB expression boils down to

$$\mathrm{CRB}_{N_4} = \frac{\sigma_v^2 d_{1,4}^2 d_{2,4}^2}{2A_{(N_1,N_2,N_4)}^2}. \tag{4.8}$$

These CRB expressions provide a selection criterion (SC) for choosing the baseline nodes N_1 and N_2. Considering the aforementioned assumption that N_2 is perfectly located, the location estimates of N_4 and N_5 can be considered statistically

independent which results in

$$\text{SC} = \text{CRB}_{\text{total}} = \text{CRB}_{N_4} + \text{CRB}_{N_5}. \tag{4.9}$$

The pair of nodes that provides the minimum SC in (4.9) will be chosen as N_1 and N_2. We call this modified algorithm for noisy scenarios, the modified GLR (MGLR). One interesting solution proposed in [81] called linear algebraic reconstruction (LAR) proves that if we have a similar 5-node setup as explained for the GLR, the missing distance can be recovered by considering the singularity of the Schur complement of $\mathbf{D}^{(5)}$ (noisy distance matrix for N_1 to N_5 with missing link set to zero) with respect to $\mathbf{D}^{(3)}$ (noisy distance matrix for N_1 to N_3) as defined by

$$\mathbf{D}^{(5)} = \begin{bmatrix} \mathbf{D}^{(3)} & \mathbf{E} \\ \mathbf{E}^T & \mathbf{0}_{2\times 2} \end{bmatrix}. \tag{4.10}$$

This will give us a second-order polynomial with two roots corresponding to the missing distance. The root which constructs a rank-2 $\mathbf{B}^{(5)}$ matrix corresponding to the reconstructed complete $\hat{\mathbf{D}}^{(5)}$, will be chosen. Although the algorithm is exact for noiseless scenarios, in a noisy scenario, none of the two roots will construct a $\mathbf{B}^{(5)}$ matrix with rank two. A simple modification that comes to mind is to construct both $\mathbf{B}^{(5)}$ matrices and choose the one which is closer to a rank-2 matrix. To this aim, we can define a rank selection metric $\rho = \sum_{i=1}^{2} |\lambda_i| / \sum_{i=3}^{5} |\lambda_i|$ (where $\{\lambda_i\}$ denote the eigenvalues of $\mathbf{B}^{(5)}$) and choose the root which yields a larger ρ. We call this algorithm the modified LAR (MLAR). The other possible solution is to simplify the Nyström algorithm (on behalf of all Nyström-based algorithms explained in [82]) for the case of the explained 5-node setup with one missing link. To do this, we first calculate the relative coordinates of N_1 to N_3, denoted by a 2×3 matrix $\mathbf{Y}$, by doing a double-centering on $\mathbf{D}^{(3)}$ and then computing an EVD on $\mathbf{B}^{(3)}$ as

$$\mathbf{B}^{(3)} = -\tfrac{1}{2}\mathbf{H}_3\mathbf{D}^{(3)}\mathbf{H}_3, \;\; \mathbf{B}^{(3)} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T, \;\; \mathbf{Y} = \mathbf{\Sigma}_s^{\frac{1}{2}}\mathbf{U}_s^T,$$

where $\mathbf{H}_n$ stands for an $n \times n$ centering operator and subscript s indicates the submatrices corresponding to the eigenvectors with the 2 largest positive eigenvalues. Next, we also bring the center of gravity of the group containing N_4 and N_5 to the origin and exploit the known distances between N_4 and N_5 and the other nodes to recover the coordinates of N_4 and N_5, denoted by a 2×2 matrix $\mathbf{Z}$, as in [82]

$$\mathbf{F} \;\; = \;\; -\tfrac{1}{2}\mathbf{H}_3\mathbf{E}\mathbf{H}_2, \;\; \mathbf{Z} \;\; = \;\; \mathbf{Y}^{-T}\mathbf{F}.$$

Finally, the missing distance can be recovered from the dummy locations we calculated in $\mathbf{Z}$ for the nodes 4 and 5.

4.3.2 Distance Matrix Reconstruction and Network Localization

To be able to reconstruct the network distance matrix completely, and subsequently use it in the PEST, we propose to repeat the missing link reconstruction for all the missing links in a local-to-global fashion. Therefore, in every snapshot of the mobile network, we first discover the missing links, then for every pair of nodes with a missing link we try to find three other nodes meeting the requirements explained in Subsection 4.3.1. Obviously, in sparsely connected networks, there may be two nodes for which we cannot find the three neighboring nodes as explained earlier (irrecoverable missing links). To alleviate this problem, we should always recover the missing links which are recoverable in a first round and in the next round there is a good chance that some of the irrecoverable missing links can be recovered due to previously recovered missing links. We repeat this procedure as long as we can recover some missing links. Notably, as we recover the missing links the probability that we can find more than one group of three nodes meeting the required conditions increases. In those cases, we choose one of these groups which meets the following criterion

$$\arg \min_{g,l} \ \mathrm{SC}_{g,l} \quad g = 1, 2, \cdots, G; \ l = 1, 2, 3, \tag{4.11}$$

where G denotes the number of possible 3-node neighboring groups and l indicates the index of the chosen edge determined by N_1 and N_2. At the end, if there are still a few missing links not recovered, for mobile networks with slow dynamics, we can always exploit the previously recorded distance measurements (or recovered distance estimates) and use them instead of the shortest path estimate, which hopefully can give us better estimates. This can be further refined by filling the missing distances with r_0 if the previously recorded distance measurement (or recovered distance estimate) for that link is less than r_0 as

$$[\hat{\mathbf{D}}_k]_{i,j} = \begin{cases} [\hat{\mathbf{D}}_{k-1}]_{i,j} & \text{if } (i,j) \text{ is irrecoverable } \& \ [\hat{\mathbf{D}}_{k-1}]_{i,j} > r_0^2, \\ r_0^2 & \text{if } (i,j) \text{ is irrecoverable } \& \ [\hat{\mathbf{D}}_{k-1}]_{i,j} \leqslant r_0^2. \end{cases} \tag{4.12}$$

By exploiting this property of mobile networks, we depart from the existing literature that may leave some nodes not localized [83]. The reconstructed distance matrix at the k-th snapshot ($\hat{\mathbf{D}}_k$) will be fed to the PEST to recover the locations of the mobile nodes. The whole process of localization in a partially connected mobile network is shown in Algorithm 4.1.

Algorithm 4.1 Localization in partially connected networks

Initialization: Start with an initial location guess
For $k = 1$ to K (movement steps)
  Step I: Reconstruction
  $\hat{\mathbf{D}}_k \leftarrow \mathbf{D}_k$
  While no. of recoverable missing links > 0
    Look for groups of three appropriate nodes in $\hat{\mathbf{D}}_k$
    Choose one appropriate group and N_1 and N_2 using (4.11)
    Recover the missing using MGLR and fill $\hat{\mathbf{D}}_k$
  End
  Complete irrecoverable missing links using (4.12)
  Step II: Localization
  Use $\hat{\mathbf{D}}_k$ in PEST to recover the locations
End

Table 4.1: Reconstruction computational complexity

Algorithm	Mult.	SQRT	Matrix inverse	EVD	Tot. FLOPS
MLAR	37	3	1 (3×3)	2 (5×5)	404
Nyström	122	2	-	1 (3×3)	173
GLR	37	5	-	-	97
MGLR	67	5	-	-	127

4.4 Reconstruction Computational Complexity

We define the reconstruction computational complexity as the number of operations required to reconstruct one missing link. For the sake of simplicity, we do not count the number of additions and subtractions due to the negligible complexity in comparison with the other operations. Also, we consider the same complexity for multiplications and divisions, and hence, we present the sum of them as the number of floating point operations (FLOPS). The results of the computational complexity for the MLAR, the Nyström, the GLR and the MGLR algorithms are summarized in Table 4.1. To calculate the total number of FLOPS required, we assume the same methods and complexities as explained in [75] for matrix inverse, scalar square root (SQRT) and EVD computation. As can be seen from the last column of the table, the GLR and the MGLR algorithms have the lowest complexities among all the algorithms under consideration and this makes them preferable for practical implementations, especially for sparsely connected networks with a lot of missing links. Note that this amount of complexity times the number of missing links in a given network yields the total complexity overload imposed by the network distance matrix reconstruction process. It is noteworthy that in the GLR (and MGLR), after

fixing the locations of N_1 to N_3 in the relative coordinate system, we could also use them to find the locations of N_4 and N_5 using classical trilateration; however, it requires much higher complexity and thus we prefer the proposed MGLR.

4.5 Simulation Results

We start by illustrating the effect of the proposed MGLR algorithm on a 5-node link reconstruction setup. The nodes are randomly deployed in an area of 100×100 square meters and the link between N_4 and N_5 is always missing. The result is shown in Fig. 4.2 where we plot the root mean squared error (RMSE) of missing link reconstruction versus γ. The results are averaged over 50000 Monte Carlo (MC) trials for 50 random configurations of nodes and 1000 realizations of the noise. The results reveal that GLR performs better than the MLAR and the Nyström. Moreover, the MGLR which exploits the proposed SC outperforms all the other algorithms. Remember that the MGLR has a much lower complexity compared to the MLAR and the Nyström, as well. In the next simulations, we present the results of exploiting the MLAR, the Nyström and the MGLR in distance matrix reconstruction for anchorless localization of a mobile network as explained in Subsection 4.3.2 and briefly illustrated in Algorithm 4.1. To this aim, we consider a network of $N = 10$ mobile sensors living inside a 2-dimensional bounded area of 100×100 square meters. Further, to be able to evaluate and plot the results based on the absolute locations, we resolve the unknown translation and orthogonal transformation of our obtained location estimates for all the algorithms by considering 3 anchor nodes. As explained earlier, the distance measurements are impaired by additive distance dependent noise. Note that, for instance, according to (4.2) at $\gamma = 30$dB we can have a maximum $\sigma_{v,i,j,k} = 100\sqrt{2}/\sqrt{1000} \approx 4.5$m of error on distance measurements. The detail of the movement model is perfectly similar to the explanations in [20, 75] with process noise standard deviation $\sigma_w = 0.1$ and measurement time interval $T_s = 0.1$s.

For a quantitative comparison, we define the positioning root mean squared error (PRMSE) of the algorithms at the k-th snapshot as

$$\text{PRMSE} = \sqrt{\frac{\sum_{m=1}^{M}\sum_{n=1}^{N} e_{n,m,k}^2}{M}}, \tag{4.13}$$

where $e_{n,m,k}$ represents the distance between the real location of the n-th node and its estimated location at the m-th MC trial of the k-th snapshot. All simulations are

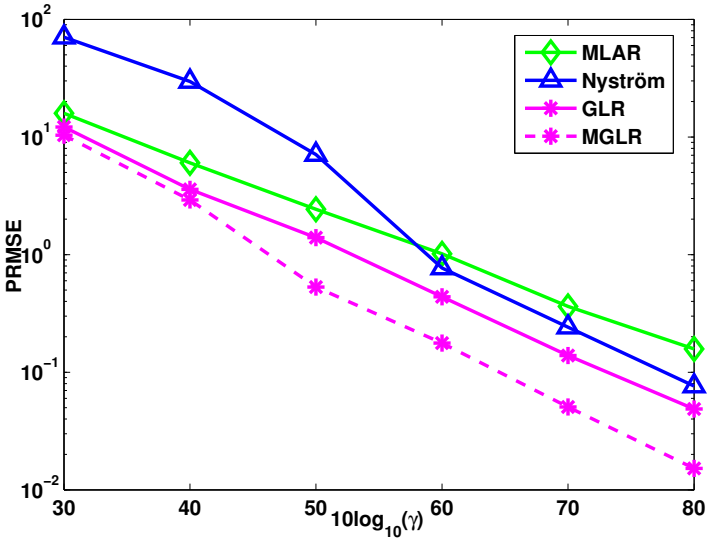


Figure 4.2: Missing link reconstruction error

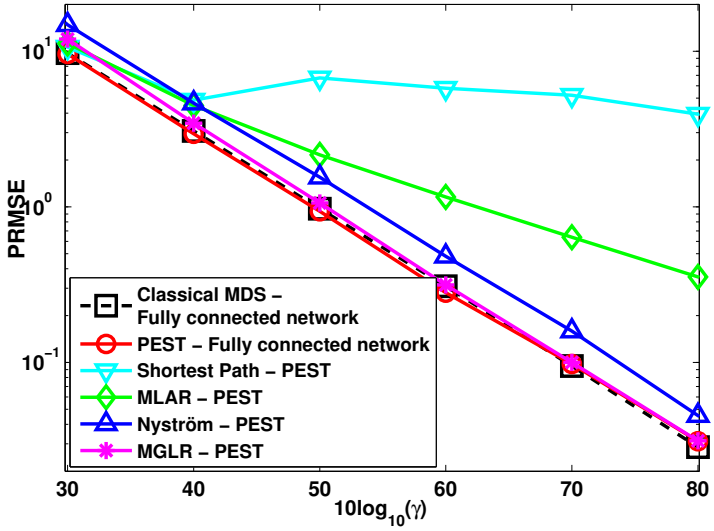


Figure 4.3: Results for a partially connected network with $r_0 = 110$m

averaged over $M = 100$ independent MC runs where in each run the nodes move toward random directions starting from random initial locations.

Fig. 4.3 depicts the performance of Algorithm 4.1 using the MLAR, the Nyström and the MGLR for a partially connected WSN with $r_0 = 110$m (approximately

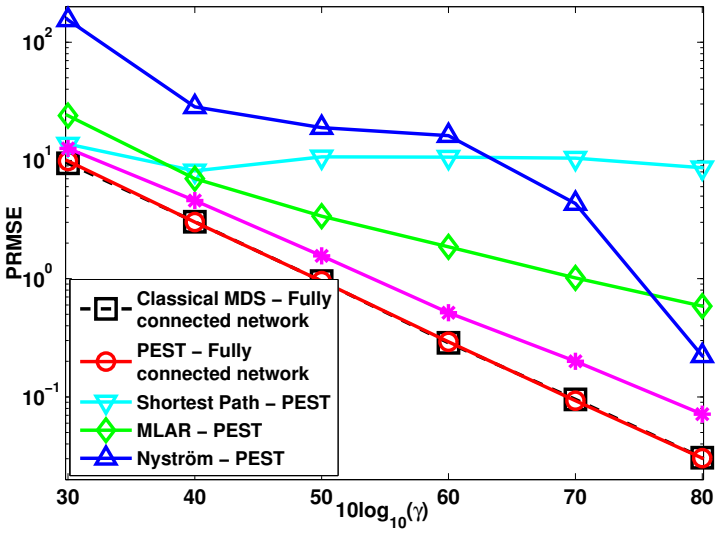


Figure 4.4: Results for a partially connected network with $r_0 = 100$m

up to 10 missing links). We plot the performance of the classical MDS over the same fully connected network as the lower bound of PRMSE and the PEST over the fully connected network as a base-line algorithm for the sake of comparison [75, 82]. Besides, we also plot the results of using the shortest path algorithm to estimate the missing links in combination with the PEST. The results illustrate that the PEST attains the achievable bound determined by the classical MDS for the fully connected network. The proposed MGLR algorithm performs the best and is very close to the performance of a fully connected network, which means it is capable of reconstructing up to 10 missing links. Note that the shortest path fails to recover the missing links as it does not show any improvement by increasing γ and also the MLAR performs much worse than the MGLR and the Nyström. Remember that considering the lowest complexity of the MGLR as well as its best accuracy, it is the preferable choice for a partially connected network. Fig. 4.4 shows the same scenario as in Fig. 4.3 except for $r_0 = 100$m (approximately up to 14 missing links). It is interesting that while the Nyström shows signs of instability and the MLAR still does not perform well, the MGLR gives the best performance even for a network with $14/\binom{10}{2} > 30\%$ missconnectivity.

4.6 Conclusions

We have proposed a geometric link reconstruction algorithm for noisy scenarios. The proposed algorithm is then used in a local-to-global fashion to reconstruct the

complete network distance matrix and localize the mobile network. It has been shown that the proposed algorithm has a low computational complexity and outperforms comparable existing approaches in terms of link reconstruction and network localization accuracy in noisy scenarios.

PART

III

SPARSITY-AWARE MULTI-SOURCE LOCALIZATION

No great discovery was ever made without a bold guess.

SIR ISAAC NEWTON

5

SPARSITY-AWARE MULTI-SOURCE TDOA LOCALIZATION

Abstract

The problem of source localization from time-difference-of-arrival (TDOA) measurements is in general a non-convex and complex problem due to its hyperbolic nature. This problem becomes even more complicated for the case of *multi-source* localization where TDOAs should be assigned to their respective sources. We simplify this problem to an ℓ_1-norm minimization by introducing a novel TDOA fingerprinting and grid design model for a multi-source scenario. Moreover, we propose an *innovative trick* to enhance the performance of our proposed fingerprinting model in terms of the number of identifiable sources. An interesting by-product of this enhanced model is that under some conditions we can convert the given underdetermined problem to an overdetermined one that could be solved using classical least squares (LS). Finally, we also tackle the problem of off-grid source localization as a case of grid mismatch. Our extensive simulation results illustrate a good performance for the introduced TDOA fingerprinting paradigm as well as a significant detection gain for the enhanced model.

5.1 Introduction

Determining the position of multiple sources in a two-dimensional or three-dimensional (2-D or 3-D) space is a fundamental problem which has received an upsurge of attention recently [84]. Many different approaches have been proposed in literature to recover the source locations based on time-of-arrival (ToA), time-difference-of-arrival (TDOA) or received-signal-strength (RSS) measurements between the source nodes (SNs) and some fixed receivers or access points (APs). A traditional wisdom in RSS-based localization tries to extract distance information from the RSS measurements. However, this approach fails to provide accurate location estimates due to the complexity and unpredictability of the wireless channel. This has motivated another category of RSS-based positioning, the so-called location

fingerprinting, which discretizes the physical 2-D or 3-D space into grid points (GPs) and creates a map representing the space by assigning to every GP a set of location-dependent RSS parameters, one for every AP. The location of the source is then estimated by comparing real-time measurements with the fingerprinting map at APs, for instance using K-nearest neighbors (KNN) [12] or Bayesian classification (BC) [13].

A closer look at the grid-based fingerprinting localization problem reveals that the source location is unique in the spatial domain, and can thus be represented by a 1-sparse vector. This motivated the use of compressive sampling (CS) [85] to recover the location of the source using only a few measurements by solving an ℓ_1-norm minimization problem. This idea (for RSS measurements) illustrated promising results for the first time in [86, 87] as well as in the subsequent works [26, 29, 88]. Existing RSS-based sparse localization algorithms only make use of the signal/RSS readings at different receivers (or APs) separately. However, there is potential information in the cross-correlations of these received signals at different APs which has not been exploited in the aforementioned works. In [89], we have proposed to reformulate the sparse localization problem so that we can make use of the cross-correlations of the signal readings at different APs, which leads to a considerable improvement in terms of the number of identifiable sources. Notably, all the aforementioned studies consider on-grid target(s) or source(s).

On the other hand, the problem of TDOA-based localization for a single (multiple) source(s) has been investigated from different perspectives in literature, for instance in the speech and acoustic domain [30, 31, 90–92]. In speech processing, algorithms often rely on the speech non-stationarity (TDOAs can be assigned to different sources using this assumption) which does not hold in our context. That is why some of these studies consider disjoint sources such as [90] and in many others linear array receivers are assumed and thus the problem basically boils down to direction of arrival (DOA) estimation [92]. In a big line of research, the conversion of phase to TDOA leads to aliasing effects at high frequencies for large receiver spacings [30, 92]. In [30], for instance, a blind source separation (BSS) signal model is considered and a beamforming procedure is proposed to produce an acoustic map of the covered area. To obtain such a map, distance information (between source(s) and receivers) is required which becomes computationally demanding for a near-field assumption. In [91], a fingerprinting-like approach is proposed and the area is discretized into a set of GPs for which an acoustic map function is defined. Through a proper processing of the acoustic map and de-emphasizing the effect of the dominant source, they illustrate a good performance in localizing two sources, but in some situations their performance drops if the number of tar-

gets is larger than three. Interestingly, none of the aforementioned studies exploits CS or sparse reconstruction ideas and surprisingly, not much work can be found on TDOA-based source localization within a sparse representation framework. In [25], a single-source TDOA-based localization is proposed wherein the sparsity of the multipath channel is exploited for time delay estimation but we are basically interested in spatial source sparsity, i.e., we want to exploit the fact that the sources are sparse in the 2-D or 3-D space. On the other hand, in [93], the spatial source sparsity is exploited to simplify the hyperbolic source localization problem into an ℓ_1-norm minimization. However, the algorithm in [93] treats different sources separately, i.e., it is in principle a single-source localization approach. Besides, the problem of off-grid source localization is not really tackled in [93]. A conference pre-cursor of this work is presented in [94].

The contribution of this chapter is four-fold. Firstly, we formulate the problem of sparsity-based multi-source localization by defining a novel TDOA fingerprinting paradigm to simplify the complexity and non-convexity of the multi-source TDOA localization problem. The proposed paradigm solves the problem of the TDOA assignment and multi-source localization in a joint fashion. Second, we present an appropriate grid design for our fingerprinting model. Further, we propose a *novel trick* to enhance our proposed fingerprinting paradigm in terms of the number of identifiable sources, which leads to a significant detection gain. And finally, we extend our ideas by tackling the problem of off-grid source localization. To this aim, we propose two algorithms inspired by the grid mismatch concept as well as the sparse total least squares (STLS) method proposed in [95]. It is worth pointing out that the proposed algorithms can be applied in outdoor environments where location-based services are of interest. Therefore, there is no limitation to employ the proposed ideas in wireless local area networks (WLANs) or wireless sensor networks (WSNs) operating in a centralized fashion. A notation summary of the symbols used in the following sections is given in Tabel 5.1.

The rest of this chapter is organized as follows. In Section 5.2, the TDOA network model as well as our measurement model are explained. Section 5.3 introduces our novel sparse multi-source TDOA localization idea as well as the proposed grid design. Section 5.4 presents an innovative approach to enhance the performance of our proposed multi-source algorithm. The problem of off-grid source localization is investigated in Section 5.5. Extensive simulations in Section 5.6 corroborate our analytical claims in several scenarios. Finally, this chapter is wrapped up in Section 5.7 with brief concluding remarks.

Symbol	Description
$\mathbf{s}_k(t)$	k-th SN's signal
$x_i(t),\ n_i(t)$	Received signal and noise at the i-th AP
$r_i(\Delta)$	Cross-correlation w.r.t $(\text{AP}_1,\ \text{AP}_i)$ pair
$\Delta_i^{(k)}$	k-th TDOA peak in $r_i(\Delta)$
$\mathbf{y}^{(k)}$	Measurement vector containing $\Delta_i^{(k)}$'s
$\Delta_{i,k}$	TDOA of the k-th SN in $r_i(\Delta)$
$\mathbf{y}_k$	Measurement vector containing $\Delta_{i,k}$'s
$\Delta_{i,n}^{\text{g}}$	TDOA of n-th GP w.r.t $(\text{AP}_1,\ \text{AP}_i)$ pair
$\mathbf{y}_n^{\text{g}}$	Measurement vector containing $\Delta_{i,n}^{\text{g}}$'s
$\text{hyp}_{i,k}$	Hyperbola of k-th SN w.r.t. $(\text{AP}_1,\ \text{AP}_i)$ pair
$\text{hyp}_{i,n}^{g}$	Hyperbola of n-th GP w.r.t. $(\text{AP}_1,\ \text{AP}_i)$ pair

Table 5.1: Description of the symbols

5.2 TDOA Network Model

Consider that we have M APs distributed over a 2-D or 3-D area which is discretized into N GPs. Note that the APs can be located anywhere, not necessarily on the GPs. We consider K SNs which are randomly located either on any of these GPs ("*on-grid*") or possibly "*off-grid*". We assume that the APs are connected to each other in a wireless or wired fashion so that they can cooperate by exchanging their signal readings. Now, if the k-th source broadcasts a time domain signal $s_k(t)$, the received signal at the i-th AP can be expressed by

$$x_i(t) = \sum_{k=1}^{K} h_{i,k} s_k(t - \tau_{i,k}) + n_i(t), \tag{5.1}$$

where in general $h_{i,k}$ is the channel coefficient and $\tau_{i,k}$ is the time delay from the k-th source to the i-th AP and $n_i(t)$ represents additive white noise. Here, for the sake of simplicity, we have considered a single-tap flat fading channel. We only consider a single-path scenario here, since it might be more suited to an outdoor environment and since it simplifies the setting in order to have a better focus on the core idea of this chapter.

In this work, we choose a set of $M-1$ TDOA measurements (the so-called non-redundant set) by always considering the first AP as the reference. Since we consider a passive source localization scenario, taking cross-correlations of the received signals is the optimal approach for extracting the TDOAs [96]. The signals

$s_k(t)$ and $n_i(t)$ are assumed to be ergodic, mutually uncorrelated white sequences, i.e.,

$$\int_t s_k(t)s_{k'}(t-\Delta)dt = \begin{cases} 0, & k \neq k' \\ \delta(\Delta), & k = k' \end{cases}, \tag{5.2a}$$

$$\int_t n_i(t)n_j(t-\Delta)dt = \begin{cases} 0, & i \neq j \\ \delta(\Delta), & i = j \end{cases}, \tag{5.2b}$$

$$\int_t s_k(t)n_i(t)dt = 0. \tag{5.2c}$$

where $\delta(.)$ stands for the unit impulse function. Therefore, by considering (5.2), the cross-correlation between the received signal at the i-th AP and the reference AP is given by

$$\begin{aligned} r_i(\Delta) &= \int_t \Big(\sum_{k=1}^{K} h_{i,k}s_k(t-\tau_{i,k}) + n_i(t) \Big) \times \\ &\qquad \Big(\sum_{k'=1}^{K} h_{1,k'}s_{k'}(t-\Delta-\tau_{1,k'}) + n_1(t-\Delta) \Big) dt \\ &= \sum_{k=1}^{K}\sum_{k'=1}^{K} \int_t \Big(h_{i,k}s_k(t-\tau_{i,k}) + n_i(t) \Big) \times \\ &\qquad \Big(h_{1,k'}s_{k'}(t-\Delta-\tau_{1,k'}) + n_1(t-\Delta) \Big) dt \\ &= \sum_{k=1}^{K} \int_t \Big(h_{i,k}s_k(t-\tau_{i,k}) + n_i(t) \Big) \times \\ &\qquad \Big(h_{1,k}s_k(t-\Delta-\tau_{1,k}) + n_1(t-\Delta) \Big) dt \\ &= \sum_{k=1}^{K} h_{i,k}h_{1,k}\delta(\Delta-\Delta_{i,k}), \end{aligned} \tag{5.3}$$

where $\Delta_{i,k} = \tau_{1,k} - \tau_{i,k}$ is the TDOA of the k-th source w.r.t. the AP pair $(\text{AP}_1,\ \text{AP}_i)$. As is shown by (5.3), for a single-tap channel as considered here, the K dominant peaks of $r_i(\Delta)$ return the TDOA values $\{\Delta_{i,k}\}_k$ related to the K sources. Note that in this work we assume that K is known even though target counting algorithms (such as a modified version of [29]) can be applied to estimate K in advance.

The main problem with (5.3) is that even though we can estimate the set of TDOAs

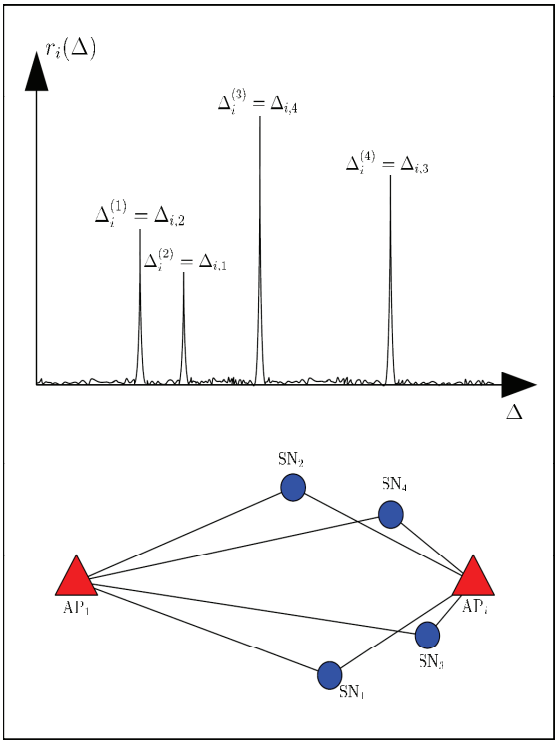


Figure 5.1: Artificial setup for assignment problem; definition of $\Delta_i^{(k)}$ and $\Delta_{i,k}$. Note that SN_2 produces the smallest TDOA while SN_3 produces the largest one.

$\{\Delta_{i,k}\}_k$, we do not know the source indices of the TDOAs. This leads to an assignment problem to relate the TDOAs to the sources. To make it more clear, as shown in Fig. 5.1, we define the $\Delta_i^{(k)}$'s which denote the TDOAs in an increasing order ($\Delta_i^{(1)} \leqslant \cdots \leqslant \Delta_i^{(K)}$). These $\Delta_i^{(k)}$'s can be measured for $i = 2, \cdots, M$ and they are stacked in the measurement vectors $\mathbf{y}^{(k)} = [\Delta_2^{(k)}, \cdots, \Delta_M^{(k)}]^T$. Note the difference with the $\Delta_{i,k}$'s, which denote the TDOA values ordered according to the source indices leading to the vectors $\mathbf{y}_k = [\Delta_{2,k}, \cdots, \Delta_{M,k}]^T$. It is worth mentioning that while the $\mathbf{y}^{(k)}$ vectors are perfectly known, the $\mathbf{y}_k$ vectors are not. Now, the problem considered herein can be stated as follows. How can we assign the TDOAs to the different sources and simultaneously localize them? We would like to emphasize that we tackle the problem of *passive* multi-source localization where we have no knowledge about the signals transmitted by the sources except for the common assumption that they are mutually uncorrelated white sequences;

otherwise, any sort of information about the signal (such as identification label, the occupied bandwidth, the time slot in which they are transmitted, etc.) can help to dissect the problem into K separate localization problems that can be solved disjointly. We start our solution development by considering on-grid sources and then we extend it to the case of off-grid sources.

5.3 Sparsity-Aware TDOA Localization

In order to assign the TDOAs to the different sources and simultaneously localize them, we propose a fingerprinting procedure. We start this procedure with an *initialization* phase where the fingerprinting map is determined. Then, in the *runtime* phase, this map is used together with the measured TDOAs to determine the location of the SNs.

5.3.1 Initialization Phase

In the initialization phase, we basically discretize the physical space into GPs and create a map (the so-called fingerprinting map) representing the space by assigning to every GP a set of location-dependent parameters. For the TDOA setup under consideration, the location-dependent parameter set will consist of the TDOA measurements from the APs. For every GP, we determine the $M-1$ TDOAs at the different APs w.r.t. the first AP. Next, by concatenating the measurements from N GPs we construct a fingerprinting map $\mathbf{\Psi}$ of size $(M-1)\times N$ of the form

$$\mathbf{\Psi} = \begin{bmatrix} \Delta^{\mathrm{g}}_{2,1} & \cdots & \Delta^{\mathrm{g}}_{2,N} \\ \vdots & \ddots & \vdots \\ \Delta^{\mathrm{g}}_{M,1} & \cdots & \Delta^{\mathrm{g}}_{M,N} \end{bmatrix}, \tag{5.4}$$

where $\Delta^{\mathrm{g}}_{i,n}$ represents the TDOA of the received signal at the i-th AP and the reference AP from a source located at the n-th GP. Note the difference with $\Delta_{i,k}$ which is the measured TDOA from the k-th source w.r.t. the $(\mathrm{AP}_1,\ \mathrm{AP}_i)$ pair. To determine the $\Delta^{\mathrm{g}}_{i,n}$'s, we can simply use the known geometric configuration of the APs and GPs. This is highly desirable as we can avoid *exhaustive* classical training procedures.

5.3.2 Runtime Phase

For the runtime phase, we make a distinction between a single-source and multi-source scenario as explained in the following.

Single-source scenario

In the single-source case, the location of the source is estimated by comparing the runtime phase TDOA measurements $\mathbf{y} = \mathbf{y}^{(1)} = [\Delta_2^{(1)}, \cdots, \Delta_M^{(1)}]^T$[1] with the fingerprinting map, at a central unit connected to the APs. One way to carry out this comparison is by exploiting the source sparsity and considering that the source can only be located at a single GP. This way, the single-source localization problem can be cast into a sparse representation framework given by $\mathbf{y} = \boldsymbol{\Psi}\boldsymbol{\theta} + \boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon}$ an $(M-1) \times 1$ vector containing the additive noise on the TDOAs, and $\boldsymbol{\theta}$ an $N \times 1$ vector with all elements equal to zero except for one element equal to one corresponding to the index of the GP where the source is located. Thus, $\mathbf{y}$ will be a 1-sparse TDOA vector characterized by the sparsity basis $\boldsymbol{\Psi}$ and the ultimate goal is to recover $\boldsymbol{\theta}$ only by determining the index of its non-zero element.

Solving $\mathbf{y} = \boldsymbol{\Psi}\boldsymbol{\theta} + \boldsymbol{\epsilon}$ with classical LS produces an incorrect estimate due to the underdetermined nature of the problem ($M - 1 \ll N$). Instead, sparse reconstruction techniques (or CS aim to reconstruct $\boldsymbol{\theta}$ from $\mathbf{y}$, by taking the source sparsity concept into account. It is worth mentioning that here we have a natural compression in the problem in the sense that the number of measurements is limited to $M - 1$ which in many practical scenarios is much less than the number of GPs N. Therefore, we will estimate $\boldsymbol{\theta}$ by solving the following ℓ_1-norm minimization problem (similar to [93]) $\min_{\boldsymbol{\theta}} \|\mathbf{y} - \boldsymbol{\Psi}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1$ where λ is a regularization parameter that controls the trade-off between sparsity and reconstruction fidelity of the estimated $\boldsymbol{\theta}$. It is worth mentioning that for a single-source scenario some other simpler methods, like matching pursuit [97], can also be used to recover the location of the source.

Multi-source scenario

Having explained the single-source TDOA localization within a sparse framework, now, the question is how we can extend this single-source localization scheme to a multi-source one. Before explaining the idea, we would like to remind the reader of a natural phenomenon in RSS fingerprinting. Different from TDOA measurements, the RSSs of the source signals will sum up at the APs [26, 89]. On the other hand, TDOA measurements do not simply follow this pattern. Nevertheless, this motivated us to sum up the measured $\Delta_i^{(k)}$ values for different sources at the APs, i.e., $\mathbf{y} = \sum_k \mathbf{y}^{(k)}$. Note that this vector is equal to $\mathbf{y} = \sum_k \mathbf{y}_k$ and thus automatically

[1] Note that only for a single-source scenario $\mathbf{y}^{(1)} = \mathbf{y}_1$, but this cannot be generalized to a multi-source scenario, i.e., in that case we generally have $\mathbf{y}^{(k)} \neq \mathbf{y}_k$.

leads to a similar formulation as for the single-source case

$$\mathbf{y} = \boldsymbol{\Psi}\boldsymbol{\theta} + \boldsymbol{\epsilon}, \tag{5.5}$$

where here $\boldsymbol{\theta}$ is a K-sparse vector (containing all zeros except for K ones) to accommodate the K sources. We would like to emphasize again that in practice we can only measure the $\mathbf{y}^{(k)}$ vectors because it is still unknown to which source they belong, i.e., the $\mathbf{y}_k$ vectors cannot be separately calculated. However, the *beauty* of the proposed sparsity-aware multi-source TDOA localization (SMTL) framework is that since we work with $\mathbf{y} = \sum_k \mathbf{y}^{(k)} = \sum_k \mathbf{y}_k$, it does not really require such assignment information. Therefore, similar to the single-source scenario, (5.5) can also be solved using an ℓ_1-norm minimization

$$\hat{\boldsymbol{\theta}}_{\text{SMTL}} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \boldsymbol{\Psi}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 , \tag{5.6}$$

where λ is defined as earlier. Notably, outliers in the measured TDOAs $\mathbf{y}$ can be handled within our sparsity-aware framework by exploiting the ideas proposed in [98].

Remark 5.1 (*Identifiability of SMTL*)

To elaborate on the identifiability of localization using SMTL, it is worth mentioning that in a classical (2-D) TDOA localization, as long as there are $M > 3$ APs (not lying on a straight line) associated with a source, that source can be uniquely identified and localized. In a multi-source case, however, all possible assignments between TDOAs and sources have to be checked. On the other hand, the sparse reconstruction-based nature of SMTL imposes an extra constraint $M - 1 \geqslant 2K$ ($M > 3$ should also be satisfied) because for a perfect reconstruction we require every $2K$-column subset of $\boldsymbol{\Psi}$ to be full column rank so that we can reconstruct a K-sparse $\boldsymbol{\theta}$. All in all, this leads to $M > max(2K, 3)$ as a necessary condition for identifiability and reconstruction.

5.3.3 Grid Design

In the earlier proposed TDOA formulation an unintentional grid problem shows up. Consider that we have three APs (AP_1 to AP_3) and three source nodes (SN_1 to SN_3) as in Fig. 5.2. Now, assume that SN_1 is located on $(8, 6)$ as shown in the figure. The set of points $\mathbf{x} = [x,\ y]^T$ that represents a constant TDOA w.r.t. AP_1 and AP_i ($\Delta_{i,1}$ is constant) defines a hyperbola given by

$$\text{hyp}_{i,1}: \quad \frac{1}{\nu}\big(d(\text{SN}_1, \text{AP}_1) - d(\text{SN}_1, \text{AP}_i)\big) = \frac{1}{\nu}\big(d(\mathbf{x}, \text{AP}_1) - d(\mathbf{x}, \text{AP}_i)\big), \tag{5.7}$$

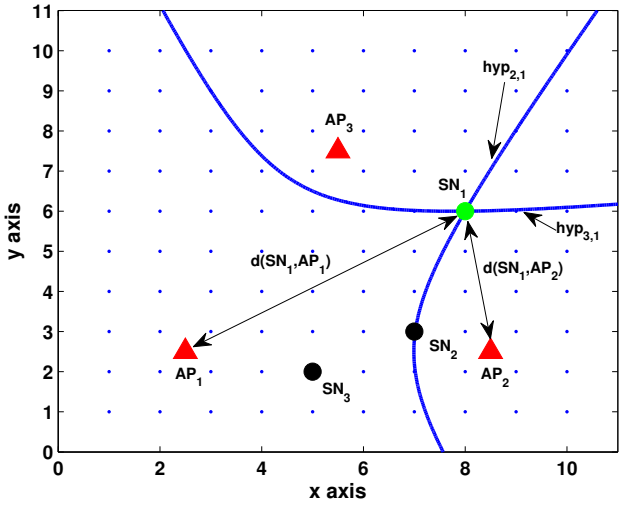


Figure 5.2: Coincident Δ's in a uniform GP configuration.

where $d(\mathrm{A},\mathrm{B}) = \sqrt{(x_\mathrm{A}-x_\mathrm{B})^2+(y_\mathrm{A}-y_\mathrm{B})^2}$ is the Euclidean distance between points A and B and ν denotes the speed of the signal propagation. For $i = 2,\ 3$ this results in the two hyperbolas ($\mathrm{hyp}_{2,1}$ and $\mathrm{hyp}_{3,1}$) plotted with solid blue lines in Fig. 5.2. In general, $\mathrm{hyp}_{i,k}$ denotes the hyperbola related to the TDOA of the source SN_k w.r.t. the $(\mathrm{AP}_1,\ \mathrm{AP}_i)$ pair. Right now, if any other source falls on either one of these two hyperbolas, that source will have a similar TDOA as SN_1 w.r.t. either the $(\mathrm{AP}_1,\ \mathrm{AP}_2)$ or $(\mathrm{AP}_1,\ \mathrm{AP}_3)$ pair. Because in Fig. 5.2 SN_2 lies on $\mathrm{hyp}_{2,1}$ and SN_3 does not, the output of the cross-correlation related to the $(\mathrm{AP}_1,\ \mathrm{AP}_2)$ pair will contain only two dominant peaks instead of three peaks. Obviously, in such a case this coincidence cannot be resolved based on the amplitude of the peaks because the signals arrive at the APs with different amplitudes depending on the fading channel. It is worth mentioning that with the uniform GP configuration as shown in Fig. 5.2, the probability of obtaining such (approximately) equal Δ values in each row of (5.4) is not low and this probability increases with the number of GPs N. Next, we propose a new grid configuration to avoid this issue, if the sources are on-grid. Note that in many practical situations, the APs are part of the existing infrastructure and we do not have the privilege neither to change their number nor their location. This basically motivates the following grid design based on a fixed AP configuration.

For a given AP configuration, we propose a sequential GP placement so that none of the sources will have a similar TDOA w.r.t. any of the AP pairs, i.e., $(\mathrm{AP}_1,\ \mathrm{AP}_i)$,

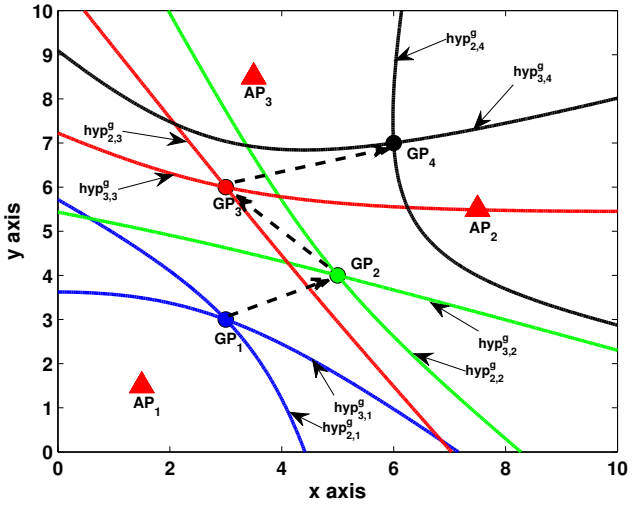


Figure 5.3: Proposed sequential GP placement.

$i = 2, \cdots, M$. Let us consider the simple scenario shown in Fig. 5.3 where again only three APs exist. We start by choosing a desired location for the first GP (GP_1). Note that we have no restriction on the location of GP_1. Now, GP_1 defines $M-1 = 2$ hyperbolas ($\text{hyp}^{\text{g}}_{2,1}$ and $\text{hyp}^{\text{g}}_{3,1}$) with $\text{hyp}^{\text{g}}_{i,n}$ defined similar to (5.7) but for the GPs as

$$\text{hyp}^{\text{g}}_{i,n}: \quad \frac{1}{\nu}\left(d(\text{GP}_n, \text{AP}_1) - d(\text{GP}_n, \text{AP}_i)\right) = \frac{1}{\nu}\left(d(\mathbf{x}, \text{AP}_1) - d(\mathbf{x}, \text{AP}_i)\right), \quad i = 2, 3, \; n = 1, \cdots, N, \tag{5.8}$$

with AP_1 chosen as the reference. Each of these hyperbolas excludes a curve from the 2-D plane of the covered area and leaves the remaining part of the plane as a possible option to place the next GP. Therefore, if we place GP_2 on either $\text{hyp}^{\text{g}}_{2,1}$ or $\text{hyp}^{\text{g}}_{3,1}$ there will be one overlapping peak in the output of the cross-correlation corresponding to the pair $(\text{AP}_1, \text{AP}_2)$ or $(\text{AP}_1, \text{AP}_3)$, respectively. After placing GP_2, two more hyperbolas should be excluded from the 2-D plane for the next GP. This means, we should not place GP_3 on any of $\text{hyp}^{\text{g}}_{2,1}$, $\text{hyp}^{\text{g}}_{3,1}$, $\text{hyp}^{\text{g}}_{2,2}$ and $\text{hyp}^{\text{g}}_{3,2}$, as is also illustrated in Fig. 5.3. The following GPs are placed in a similar fashion and this procedure can be continued until we find N GPs.

Remark 5.2 (*Backward Checking*)

It is important to observe that $hyp^{g}_{i,n}$ of GP_n can never cross $hyp^{g}_{i,n'}$ of $GP_{n'}$. This is because if they could cross, then at the crossing point we would have $\Delta_{i,n} = \Delta_{i,n'}$ and considering (5.8) this would mean that the two hyperbolas should coincide everywhere and thus GP_n and $GP_{n'}$ would be located on the same hyperbola. This is impossible according to our grid design. As a result, a hyperbola related to a GP can never cross a previously deployed GP, which means that our proposed sequential GP placement procedure does not require a backward checking modification when we place the GPs.

5.4 Enhanced Sparsity-Aware Multi-Source Localization (ESMTL)

The proposed sparsity-aware multi-source algorithm of Section 5.3 has a limited source detection capability which comes from the fact that we sum the measured TDOAs at the APs, thereby losing a significant amount of information. This basically limits the number of detectable sources (K) through the number of measurements (see Remark 5.1). The question is how this problem can be solved without taking additional TDOA measurements. The *innovative trick* we use here is to consider not just the sum of the TDOAs as $\mathbf{y} = \sum_k \mathbf{y}^{(k)} = \sum_k \mathbf{y}_k$, but the sum of any function of the TDOAs as

$$\mathbf{y}_{f_l} = \sum_k f_l(\mathbf{y}^{(k)}) = \sum_k f_l(\mathbf{y}_k), \tag{5.9}$$

where

$$f_l(\mathbf{y}^{(k)}) = [f_{l,1}(\Delta_2^{(k)}), \cdots, f_{l,M-1}(\Delta_M^{(k)})]^T \tag{5.10}$$

with $f_{l,i}(.)$ being any possible *measurement function*. If we combine a set of L such sums, i.e.,

$$\bar{\mathbf{y}} = [\mathbf{y}_{f_1}^T, \mathbf{y}_{f_2}^T, \cdots, \mathbf{y}_{f_L}^T]^T, \tag{5.11}$$

this newly defined measurement vector $\bar{\mathbf{y}}$ calls for a new fingerprinting map $\bar{\mathbf{\Psi}}$ which can accordingly be defined as

$$\bar{\mathbf{\Psi}} = \left[f_1(\mathbf{\Psi})^T, \cdots, f_L(\mathbf{\Psi})^T\right]^T, \tag{5.12}$$

where

$$f_l(\mathbf{\Psi}) = \begin{bmatrix} f_{l,1}(\Delta^{\mathrm{g}}_{2,1}) & \cdots & f_{l,1}(\Delta^{\mathrm{g}}_{2,N}) \\ \vdots & \ddots & \vdots \\ f_{l,M-1}(\Delta^{\mathrm{g}}_{M,1}) & \cdots & f_{l,M-1}(\Delta^{\mathrm{g}}_{M,N}) \end{bmatrix}, \tag{5.13}$$

and thus the model (5.5) can be extended to

$$\bar{\mathbf{y}} = \bar{\boldsymbol{\Psi}}\boldsymbol{\theta} + \bar{\boldsymbol{\epsilon}}. \tag{5.14}$$

The new $\bar{\boldsymbol{\Psi}}$ has $L(M-1)$ rows instead of only $M-1$ rows, i.e., it is capable of detecting more sources simultaneously, if the measurement functions $f_{l,i}(.)$ own certain properties. First of all, they should be nonlinear in general since linear functions generate dependent rows in $\bar{\boldsymbol{\Psi}}$ which in principle does not increase the number of independent equations in (5.14). Moreover, these functions should not impair the restricted isometry property (RIP) [59] of $\bar{\boldsymbol{\Psi}}$ required for a high quality reconstruction. Having this issue in mind, an orthonormalization procedure on the resulting $\bar{\boldsymbol{\Psi}}$ can help to improve the RIP, as we also show numerically later on.

Remark 5.3 (*Identifiability of ESMTL*)

For the enhanced model, the expected necessary identifiability condition (as explained in Remark 5.1) will be $L(M-1) \geqslant 2K$ and $M > 3$ which results in $M > max(\lceil (2K+1)/L \rceil + 1, 3)$, where $\lceil . \rceil$ denotes the ceiling operator. A detailed analysis of the dependence of the measurement functions on the identifiability is a complicated mathematical exercise which is outside the scope of this chapter and is left for future work.

In principle, the measurement functions $f_{l,i}(.)$ can be any nonlinear function. We could for instance consider a base set of L non-linear functions denoted as $\{g_l(.)\}_{l=1}^{L}$ (the $g_l(.)$ functions could for example be monomials, i.e, $g_l(.) = (.)^l$) and take $f_{l,i}(.) = g_l(.)$. In addition, to improve the RIP we could further apply the operator $\mathbf{R}$ of size $L(M-1) \times L(M-1)$ to the measurements, i.e., $\tilde{\mathbf{y}} = \mathbf{R}\bar{\mathbf{y}}$, leading to the new map $\tilde{\boldsymbol{\Psi}} = \mathbf{R}\bar{\boldsymbol{\Psi}}$. One option to design $\mathbf{R}$ could be to force the *columns* of $\mathbf{R}\bar{\boldsymbol{\Psi}}$ to be as close as possible to orthonormal by solving

$$\min_{\mathbf{R}} \left\| (\mathbf{R}\bar{\boldsymbol{\Psi}})^T (\mathbf{R}\bar{\boldsymbol{\Psi}}) - \mathbf{I}_N \right\|_F^2 . \tag{5.15}$$

Based on a detailed derivation in Appendix 5.A, if $L(M-1) \leqslant N$ this results in the following solution

$$\mathbf{R} = \boldsymbol{\Sigma}^{\dagger}(1 : L(M-1), :)\mathbf{U}^T, \tag{5.16}$$

while if $L(M-1) > N$ it leads to

$$\mathbf{R} = \begin{bmatrix} \boldsymbol{\Sigma}^{\dagger} \\ \mathbf{0}_{(L(M-1)-N)\times L(M-1)} \end{bmatrix} \mathbf{U}^T, \tag{5.17}$$

where $\mathbf{U}$ and $\mathbf{\Sigma}$ come from the singular value decomposition (SVD) of $\bar{\mathbf{\Psi}}$, i.e., $\bar{\mathbf{\Psi}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Surprisingly, this corresponds to orthonormalizing the rows of $\bar{\mathbf{\Psi}}$ (see also Appendix 5.A), which has indeed been shown to improve the RIP [26]. Having said that, by employing the operator $\mathbf{R}$, (5.14) should be modified to

$$\begin{aligned} \tilde{\mathbf{y}} = \mathbf{R}\bar{\mathbf{y}} &= \mathbf{R}\bar{\mathbf{\Psi}}\boldsymbol{\theta} + \mathbf{R}\bar{\boldsymbol{\epsilon}} \\ &= \tilde{\mathbf{\Psi}}\boldsymbol{\theta} + \tilde{\boldsymbol{\epsilon}}. \end{aligned} \tag{5.18}$$

Finally, (5.18) can be solved by

$$\hat{\boldsymbol{\theta}}_{\text{ESMTL}} = \arg\min_{\boldsymbol{\theta}} \|\tilde{\mathbf{y}} - \tilde{\mathbf{\Psi}}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 , \tag{5.19}$$

where λ is defined as earlier.

5.4.1 RIP Investigation

As we explained earlier, $\mathbf{\Psi}$ and $\bar{\mathbf{\Psi}}$ are proved to be the sparsifying bases for the SMTL and the ESMTL. Having satisfied the sparsity property, the only issue that should be assessed is the mutual incoherence between the columns of $\mathbf{\Psi}$ and $\bar{\mathbf{\Psi}}$ or alternatively the RIP. In this subsection, we try to numerically investigate the RIP property of the proposed fingerprinting maps to illustrate that the reconstruction will indeed have a high quality. As we discussed earlier, to improve the ℓ_1-norm reconstruction problem we apply the orthonormalization operator $\mathbf{R}$ to $\bar{\mathbf{\Psi}}$ (and similarly to $\mathbf{\Psi}$) and that is why we only investigate the RIP of the resulting matrices. As is well documented in literature [59], for $K = 1, 2, \cdots$ the RIP constant δ_K of a matrix $\mathbf{A}$ (with normalized columns) is the smallest number for which

$$-\delta_K \leqslant \frac{\|\mathbf{A}\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} - 1 \leqslant \delta_K, \tag{5.20}$$

for all K-sparse $\mathbf{x} \in \mathbb{R}^N$. Roughly speaking, as long as $0 < \delta_K < 1$ the RIP holds. However, the fact that we need to know all the combinations $\binom{N}{K}$ for $K = 1, 2, \cdots$ makes the problem NP-hard. For the sake of computational complexity, we use the definition in [59] where δ_K is defined as the maximum distance from 1 of all the eigenvalues of the $\binom{N}{K}$ submatrices, $\mathbf{A}_\Lambda^T\mathbf{A}_\Lambda$, derived from $\mathbf{A}$, where Λ is a set of indices with cardinality K which selects those columns of $\mathbf{A}$ indexed by Λ. It means that for each K, the RIP constant is given by

$$\delta_K = \max_{\Lambda} \left(|\lambda_{\max}(\mathbf{A}_\Lambda^T\mathbf{A}_\Lambda) - 1|,\ |\lambda_{\min}(\mathbf{A}_\Lambda^T\mathbf{A}_\Lambda) - 1| \right). \tag{5.21}$$

Matrix	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8	δ_9	δ_{10}
$\mathbf{N}_{5\times25}$	0.000	0.942	1.775	2.497	3.152	3.651	4.112	4.506	4.840	5.162
$\mathbf{R}\,\boldsymbol{\Psi}$	0.000	0.951	1.821	2.664	3.350	4.084	4.398	4.729	5.061	5.389
$\mathbf{N}_{25\times25}$	0.000	0.608	0.959	1.240	1.475	1.709	1.897	2.058	2.186	2.279
$\tilde{\boldsymbol{\Psi}} = \mathbf{R}\,\bar{\boldsymbol{\Psi}}$	0.000	0.517	0.725	0.851	0.946	0.974	0.982	0.989	0.994	0.996

Table 5.2: RIP test

For the sake of feasibility of the computations, we consider the case where $M-1 = 5$, $N = 25$, and $L = 4$ (for the ESMTL), which is also the setup considered in one of our simulation scenarios in Section 5.6. For such a case, we have computed the δ_K with $K = 1, \cdots, 10$ for $\mathbf{R}\,\mathbf{\Psi}$ and $\tilde{\mathbf{\Psi}} = \mathbf{R}\bar{\mathbf{\Psi}}$ as well as for matrices with the same size containing elements drawn from a random normal distribution, i.e., $\mathbf{N}_{5\times 25}$ and $\mathbf{N}_{25\times 25}$. Note that such random matrices are proved to be a good choice in terms of the RIP and that is why we use them as a benchmark. The results are presented in Table 5.2.

As is clear from the table, our proposed fingerprinting map for the SMTL ($\mathbf{R}\,\mathbf{\Psi}$) is almost similar to $\mathbf{N}_{5\times 25}$ and loosely satisfies the RIP up to $K = 2$. However, for $K > 2$, δ_K starts increasing. Interestingly, we see that by the aid of the added rows using our innovative $\tilde{\mathbf{\Psi}} = \mathbf{R}\bar{\mathbf{\Psi}}$, the RIP is met for K up to 10, which is even better than for $\mathbf{N}_{25\times 25}$. It is also worth stressing that this way we could demonstrate that the proposed innovative trick indeed improves the RIP of $\tilde{\mathbf{\Psi}} = \mathbf{R}\bar{\mathbf{\Psi}}$ over $\mathbf{R}\,\mathbf{\Psi}$.

5.4.2 Advantages of ESMTL

Besides the enhanced source detection capability, there are a number of other advantages in using the ESMTL approach as explained in the following.

First of all, an important advantage of this idea is that the recently added elements of $\bar{\mathbf{y}}$ and thus $\tilde{\mathbf{y}}$ are simply generated based on the existing TDOA measurements and *no extra measurements* are required in the runtime phase. The same holds for the new rows of $\bar{\mathbf{\Psi}}$ and thus $\tilde{\mathbf{\Psi}}$ which can be computed from the rows of $\mathbf{\Psi}$. This important characteristic of the proposed TDOA fingerprinting avoids imposing extra *cost-prohibitive* measurements on the central unit.

Another important corollary of this new $\tilde{\mathbf{\Psi}}$ is healing the case of coincident Δ peaks in the output of the cross-correlations. Now that we can have several extra equations, a simple solution to heal the issue of a uniform GP configuration (explained in Section 5.3) is that when computing cross-correlations, say for the $(\mathrm{AP}_1,\ \mathrm{AP}_i)$ pair, if we notice that some peaks are overlapping (number of dominant peaks is less than K), we can ignore the corresponding elements in $\bar{\mathbf{y}}$ and correspondingly the rows in $\bar{\mathbf{\Psi}}$. This means that instead of (5.19), we solve

$$\hat{\boldsymbol{\theta}}_{\mathrm{ESMTL}} = \min_{\boldsymbol{\theta}} \|\tilde{\mathbf{y}}' - \tilde{\mathbf{\Psi}}'\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 , \tag{5.22}$$

with $\tilde{\mathbf{y}}' = \mathbf{R}\mathbf{T}\bar{\mathbf{y}}$ and $\tilde{\mathbf{\Psi}} = \mathbf{R}\mathbf{T}\bar{\mathbf{\Psi}}$ where $\mathbf{T}$ is a selection matrix which removes the elements and rows corresponding to the measurements with coincident peaks from $\bar{\mathbf{y}}$ and $\bar{\mathbf{\Psi}}$, respectively, and with $\mathbf{R}$ computed based on $\mathbf{T}\bar{\mathbf{\Psi}}$ instead of $\bar{\mathbf{\Psi}}$. Note

that this way we are actually removing some APs; however, we can live with the uniform grid configuration until we violate the necessary identifiability condition $M > \max(\lceil (2K+1)/L \rceil + 1, 3)$.

Moreover, it is noteworthy that by finding appropriate measurement functions we can keep on increasing the number of rows so that we can attain a full column rank $\tilde{\boldsymbol{\Psi}}$ matrix. In such a case, we can drop the sparsity-awareness when complexity is an issue or K is very large and recover $\boldsymbol{\theta}$ as

$$\hat{\boldsymbol{\theta}}_{\text{LS}} = \tilde{\boldsymbol{\Psi}}^{\dagger}\tilde{\mathbf{y}}. \tag{5.23}$$

Further, if we are given the statistics of $\tilde{\boldsymbol{\epsilon}}$ (e.g., the mean $\mathbf{m}_{\tilde{\boldsymbol{\epsilon}}}$ and the covariance matrix $\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}$), we obtain $\mathbb{E}\{\hat{\boldsymbol{\theta}}_{\text{LS}}\} = \tilde{\boldsymbol{\Psi}}^{\dagger}\mathbb{E}\{\tilde{\mathbf{y}}\} = \tilde{\boldsymbol{\Psi}}^{\dagger}(\tilde{\boldsymbol{\Psi}}\boldsymbol{\theta} + \mathbf{m}_{\tilde{\boldsymbol{\epsilon}}}) = \boldsymbol{\theta} + \tilde{\boldsymbol{\Psi}}^{\dagger}\mathbf{m}_{\tilde{\boldsymbol{\epsilon}}}$, and $\text{MSE}(\hat{\boldsymbol{\theta}}_{\text{LS}}) = \mathbb{E}\{\|\hat{\boldsymbol{\theta}}_{\text{LS}} - \boldsymbol{\theta}\|_2^2\} = \text{tr}\left\{\tilde{\boldsymbol{\Psi}}^{\dagger}\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}(\tilde{\boldsymbol{\Psi}}^{\dagger})^T\right\} + \mathbf{m}_{\tilde{\boldsymbol{\epsilon}}}^T(\tilde{\boldsymbol{\Psi}}^{\dagger})^T\tilde{\boldsymbol{\Psi}}^{\dagger}\mathbf{m}_{\tilde{\boldsymbol{\epsilon}}}$, where $\mathbb{E}\{.\}$ stands for the statistical expectation and $\text{tr}(.)$ denotes the trace operator. This information can also be employed to solve the problem using weighted LS (WLS) as

$$\hat{\boldsymbol{\theta}}_{\text{WLS}} = \left(\tilde{\boldsymbol{\Psi}}^T\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}\tilde{\boldsymbol{\Psi}}\right)^{-1}\tilde{\boldsymbol{\Psi}}^T\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}\left[\tilde{\mathbf{y}} - \mathbf{m}_{\tilde{\boldsymbol{\epsilon}}}\right]. \tag{5.24}$$

A detailed analysis of the mean and the covariance of the error on TDOA estimation using cross-correlations can be found in [23].

5.5 Tackling Grid Mismatch for Off-Grid Sources

The classical idea of TDOA fingerprinting as well as our proposed multi-source localization ideas (SMTL and ESMTL) are based on the assumption that the sources are located on the GPs. However, as we will show in Section 5.6, the considered models defined by (5.5), (5.14) and (5.18) return inaccurate estimates if the sources are not located on their postulated GPs. This motivated us to tackle this problem for the case of multi-source TDOA localization. One generic possibility to deal with off-grid sources is to employ the adaptive grid refinement in [24], but this requires several steps of refinement. Hence, we try to interpret this phenomenon in the form of grid or map mismatch where the measurements of the sources in $\mathbf{y}$, instead of (5.5), follow a perturbed model as

$$\mathbf{y} = \left[\boldsymbol{\Psi} + \mathbf{E}\right]\boldsymbol{\theta} + \boldsymbol{\epsilon}, \tag{5.25}$$

which means that $\mathbf{y}$ is now K-sparse within the sparsity basis $\boldsymbol{\Psi} + \mathbf{E}$. To develop our idea of mismatch recovery, we start by analyzing the relation between the measurements from off-grid sources and $\mathbf{E}$ for a noiseless case. For our TDOA model,

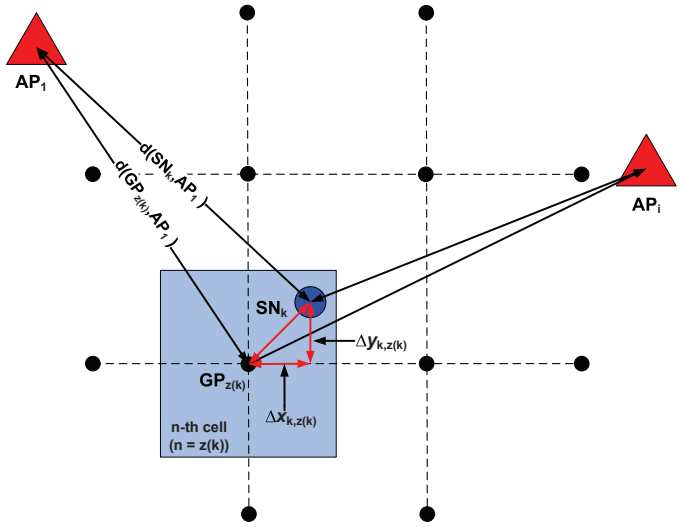


Figure 5.4: Grid mismatch.

Fig. 5.4 illustrates the case of an off-grid source in a simple setup consisting of a source SN_k and two APs (AP_1 and AP_i). As can be seen, every GP defines a so-called cell where the GP forms the center of the cell. Here, we consider that the source SN_k lies in the cell related to the n-th GP, denoted as GP_n, with $n = z(k)$ where $z(.)$ indicates the mapping between sources and GPs. Assuming that the variations of $\Delta_{i,k}$ are small within the cell related to $\text{GP}_{z(k)}$, we propose to estimate the value of the perturbed TDOA ($\Delta_{i,k}$) by considering a first-order Taylor expansion as (assuming a noiseless case)

$$\Delta_{i,k} = \Delta^{\text{g}}_{i,z(k)} + \left[\frac{\partial \Delta_i(\mathbf{x})}{\partial x}|_{\mathbf{x}=\mathbf{x}^{\text{g}}_{z(k)}} \quad \frac{\partial \Delta_i(\mathbf{x})}{\partial y}|_{\mathbf{x}=\mathbf{x}^{\text{g}}_{z(k)}} \right] \begin{bmatrix} x_k - x^{\text{g}}_{z(k)} \\ y_k - y^{\text{g}}_{z(k)} \end{bmatrix}, \tag{5.26}$$

where $\mathbf{x}^{\text{g}}_{z(k)} = [x^{\text{g}}_{z(k)},\ y^{\text{g}}_{z(k)}]^T$ denotes the location of $\text{GP}_{z(k)}$, $\mathbf{x}_k = [x_k,\ y_k]^T$ denotes the location of SN_k, and $\Delta_i(\mathbf{x})$ is the TDOA at the location $\mathbf{x} = [x,\ y]^T$ w.r.t. the $(\text{AP}_1,\ \text{AP}_i)$ pair given by

$$\begin{aligned} \Delta_i(\mathbf{x}) &= \frac{1}{\nu}\left(d(\text{AP}_1, \mathbf{x}) - d(\text{AP}_i, \mathbf{x})\right) \\ &= \frac{1}{\nu}\left(\sqrt{(x - x_{\text{AP}_1})^2 + (y - y_{\text{AP}_1})^2} - \sqrt{(x - x_{\text{AP}_i})^2 + (y - y_{\text{AP}_i})^2}\right), \end{aligned} \tag{5.27}$$

and thus its partial derivatives will be

$$\frac{\partial \Delta_i(\mathbf{x})}{\partial x} = \frac{1}{\nu}\left(\frac{x - x_{\mathrm{AP}_1}}{\sqrt{(x - x_{\mathrm{AP}_1})^2 + (y - y_{\mathrm{AP}_1})^2}} - \frac{x - x_{\mathrm{AP}_i}}{\sqrt{(x - x_{\mathrm{AP}_i})^2 + (y - y_{\mathrm{AP}_i})^2}}\right), \tag{5.28a}$$

$$\frac{\partial \Delta_i(\mathbf{x})}{\partial y} = \frac{1}{\nu}\left(\frac{y - y_{\mathrm{AP}_1}}{\sqrt{(x - x_{\mathrm{AP}_1})^2 + (y - y_{\mathrm{AP}_1})^2}} - \frac{y - y_{\mathrm{AP}_i}}{\sqrt{(x - x_{\mathrm{AP}_i})^2 + (y - y_{\mathrm{AP}_i})^2}}\right). \tag{5.28b}$$

In order to fit this into our network model, we can extend (5.26) for the case of M APs again by considering AP_1 to be the reference AP as

$$\underbrace{\begin{bmatrix} \Delta_{2,k} \\ \vdots \\ \Delta_{M,k} \end{bmatrix}}_{\mathbf{y}_k} = \underbrace{\begin{bmatrix} \Delta^{\mathrm{g}}_{2,z(k)} \\ \vdots \\ \Delta^{\mathrm{g}}_{M,z(k)} \end{bmatrix}}_{\mathbf{y}^{\mathrm{g}}_{z(k)}} + \underbrace{\begin{bmatrix} \frac{\partial \Delta_2(\mathbf{x})}{\partial x}\big|_{\mathbf{x}=\mathbf{x}^{\mathrm{g}}_{z(k)}} & \frac{\partial \Delta_2(\mathbf{x})}{\partial y}\big|_{\mathbf{x}=\mathbf{x}^{\mathrm{g}}_{z(k)}} \\ \vdots & \vdots \\ \frac{\partial \Delta_M(\mathbf{x})}{\partial x}\big|_{\mathbf{x}=\mathbf{x}^{\mathrm{g}}_{z(k)}} & \frac{\partial \Delta_M(\mathbf{x})}{\partial y}\big|_{\mathbf{x}=\mathbf{x}^{\mathrm{g}}_{z(k)}} \end{bmatrix}}_{\boldsymbol{\Delta\Psi}_{z(k)}} \underbrace{\begin{bmatrix} x_k - x^{\mathrm{g}}_{z(k)} \\ y_k - y^{\mathrm{g}}_{z(k)} \end{bmatrix}}_{\boldsymbol{\Delta x}_{k,z(k)}}. \tag{5.29}$$

It is notable that the first term on the right-hand-side of (5.29) is $\mathbf{y}^{\mathrm{g}}_{z(k)}$ and corresponds to the measurements received from an on-grid source. Clearly, in order to be able to compute the grid mismatch for SN_k ($\boldsymbol{\Delta x}_{k,z(k)}$), we first have to find the closest GP corresponding to that source given by the mapping $z(k)$. The closer this GP is to the real source location, the better the first-order Taylor expansion will work. We will come back to this problem after extending (5.29) for a multi-source scenario.

In a multi-source scenario what happens is that we receive $\sum_{k=1}^{K} \mathbf{y}_k$ instead of $\mathbf{y}_k$ which explicitly means that (5.29) should be solved for all the sources simultaneously as modeled by

$$\sum_{k=1}^{K} \mathbf{y}_k = \sum_{k=1}^{K} \mathbf{y}^{\mathrm{g}}_{z(k)} + \sum_{k=1}^{K} \boldsymbol{\Delta\Psi}_{z(k)}\, \boldsymbol{\Delta x}_{k,z(k)}, \tag{5.30}$$

where assuming that the sources (through the mapping $z(.)$) are related to different GPs, we have

$$\sum_{k=1}^{K} \mathbf{y}_k = \sum_{k=1}^{K} \mathbf{y}^{\mathrm{g}}_{z(k)} + \boldsymbol{\Delta\Psi}\,\boldsymbol{\Delta\mathbf{X}}\,\boldsymbol{\theta}, \tag{5.31}$$

with $\boldsymbol{\Delta\Psi} = [\boldsymbol{\Delta\Psi}_1, \cdots, \boldsymbol{\Delta\Psi}_N]$ and $\boldsymbol{\Delta\mathbf{X}} = \mathrm{diag}(\boldsymbol{\Delta}\mathbf{x}_1, \cdots, \boldsymbol{\Delta}\mathbf{x}_N)$, which defines a block-diagonal matrix with $\boldsymbol{\Delta}\mathbf{x}_1$ to $\boldsymbol{\Delta}\mathbf{x}_N$ as its blocks where

$$\boldsymbol{\Delta}\mathbf{x}_n = \begin{cases} \boldsymbol{\Delta}\mathbf{x}_{k,z(k)}, & \exists\, k : n = z(k) \\ \text{don't care}, & \text{otherwise} \end{cases}. \tag{5.32}$$

By exploiting $\mathbf{y} = \sum_{k=1}^{K} \mathbf{y}_k$ and $\boldsymbol{\Psi\theta} = \sum_{k=1}^{K} \mathbf{y}^{\mathrm{g}}_{z(k)}$, (5.31) can be rewritten as

$$\mathbf{y} - \boldsymbol{\Psi\theta} = \boldsymbol{\Delta\Psi}\,\boldsymbol{\Delta\mathbf{X}}\,\boldsymbol{\theta}, \tag{5.33}$$

and this can fit into the mismatch model (5.25) by taking $\mathbf{E} = \boldsymbol{\Delta\Psi}\,\boldsymbol{\Delta\mathbf{X}}$ which immediately gives an insight about the *structure* of the mismatch in our model.

In order to recover the mismatch, we now propose two approaches, both relying on the idea that if we know the indices of the closest GPs to the sources (i.e., the set $\{z(k)|\ k = 1, \cdots, K\}$) given by $\boldsymbol{\theta}$, (5.33) is overdetermined and can efficiently be solved using classical LS. More specifically, since we can derive that

$$\begin{aligned} \boldsymbol{\Delta\Psi}\,\boldsymbol{\Delta\mathbf{X}}\,\boldsymbol{\theta} &= \sum_{n=1}^{N} \boldsymbol{\Delta\Psi}_n\,\boldsymbol{\Delta}\mathbf{x}_n\,[\boldsymbol{\theta}]_n, \\ &= \sum_{n=1}^{N} \boldsymbol{\Delta\Psi}_n\,\mathrm{diag}([\boldsymbol{\theta}]_n \otimes \mathbf{1}_2)\,\boldsymbol{\Delta}\mathbf{x}_n = \boldsymbol{\Delta\Psi}\,\mathrm{diag}(\boldsymbol{\theta} \otimes \mathbf{1}_2)\,\boldsymbol{\Delta}\mathbf{x}, \end{aligned} \tag{5.34}$$

where $[\boldsymbol{\theta}]_n$ stands for the n-th element of $\boldsymbol{\theta}$ and $\boldsymbol{\Delta}\mathbf{x} = [\boldsymbol{\Delta}\mathbf{x}_1^T, \cdots, \boldsymbol{\Delta}\mathbf{x}_N^T]^T$, we obtain

$$\boldsymbol{\Delta}\mathbf{x} = [\boldsymbol{\Delta\Psi}\,\mathrm{diag}(\boldsymbol{\theta} \otimes \mathbf{1}_2)]^{\dagger}\,[\mathbf{y} - \boldsymbol{\Psi\theta}]\,. \tag{5.35}$$

In order to solve (5.35), we have to find $\boldsymbol{\theta}$ under a grid mismatch. One way to do this is to solve the following sparse total least squares (STLS) problem using ESMTL

$$\min_{\tilde{\mathbf{E}}, \tilde{\boldsymbol{\epsilon}}, \boldsymbol{\theta}} \|[\tilde{\mathbf{E}},\ \tilde{\boldsymbol{\epsilon}}]\|_F^2 + \lambda\|\boldsymbol{\theta}\|_1, \tag{5.36a}$$

$$\text{s.t. } \tilde{\mathbf{y}} = [\tilde{\boldsymbol{\Psi}} + \tilde{\mathbf{E}}]\boldsymbol{\theta} + \tilde{\boldsymbol{\epsilon}}, \tag{5.36b}$$

using the coordinate descent (CD) algorithm in [95] for the enhanced model (5.18).

Algorithm 5.1 Mismatch recovery using STLS-LS

1. Run the iterative STLS CD algorithm given by (5.38) and (5.39).
2. Find the indices of the GPs corresponding to the sources.
3. Compute $\boldsymbol{\Delta\Psi}$ and solve (5.35) to recover the off-grid locations.

Note that $[\tilde{\mathbf{E}},\ \tilde{\boldsymbol{\epsilon}}]$ denotes the augmented matrix composed of $\tilde{\mathbf{E}}$ and $\tilde{\boldsymbol{\epsilon}}$. It is worth pointing out that $\tilde{\mathbf{E}}$ is different from $\mathbf{E}$ and similarly can be written as $\tilde{\mathbf{E}} = \boldsymbol{\Delta}\tilde{\boldsymbol{\Psi}}\,\boldsymbol{\Delta}\mathbf{X}$. Therefore, instead of $\boldsymbol{\Delta\Psi}$ we have to compute $\boldsymbol{\Delta}\tilde{\boldsymbol{\Psi}}$ which instead of $\partial\Delta_i(\mathbf{x})/\partial x$ and $\partial\Delta_i(\mathbf{x})/\partial y$ evaluated at $\mathbf{x} = \mathbf{x}^{\mathrm{g}}_{z(k)}$ would contain

$$\frac{\partial f_{l,i}\big(\Delta_i(\mathbf{x})\big)}{\partial x}\Big|_{\mathbf{x}=\mathbf{x}^{\mathrm{g}}_{z(k)}} = \frac{\partial f_{l,i}\big(\Delta_i(\mathbf{x})\big)}{\partial \Delta_i(\mathbf{x})}\Big|_{\mathbf{x}=\mathbf{x}^{\mathrm{g}}_{z(k)}} \frac{\partial \Delta_i(\mathbf{x})}{\partial x}\Big|_{\mathbf{x}=\mathbf{x}^{\mathrm{g}}_{z(k)}}, \tag{5.37a}$$

$$\frac{\partial f_{l,i}\big(\Delta_i(\mathbf{x})\big)}{\partial y}\Big|_{\mathbf{x}=\mathbf{x}^{\mathrm{g}}_{z(k)}} = \frac{\partial f_{l,i}\big(\Delta_i(\mathbf{x})\big)}{\partial \Delta_i(\mathbf{x})}\Big|_{\mathbf{x}=\mathbf{x}^{\mathrm{g}}_{z(k)}} \frac{\partial \Delta_i(\mathbf{x})}{\partial y}\Big|_{\mathbf{x}=\mathbf{x}^{\mathrm{g}}_{z(k)}}. \tag{5.37b}$$

As can be seen from (5.37), the elements of $\boldsymbol{\Delta}\tilde{\boldsymbol{\Psi}}$ are scaled by a multiplicative term $\partial f_{l,i}\big(\Delta_i(\mathbf{x})\big)/\partial\Delta_i(\mathbf{x})$.

Now, if we do not exploit the explored structure of the perturbations in $\tilde{\mathbf{E}}$, the STLS problem of (5.36) can be solved by an iterative block CD algorithm yielding successive estimates of $\boldsymbol{\theta}$ with $\tilde{\mathbf{E}}$ fixed, and alternately of $\tilde{\mathbf{E}}$ with $\boldsymbol{\theta}$ fixed. Given $\tilde{\mathbf{E}}(m)$ the cost in (5.36) has the form of a LASSO problem

$$\hat{\boldsymbol{\theta}}_{\mathrm{STLS}}(m) = \arg\min_{\boldsymbol{\theta}} \left\| \tilde{\mathbf{y}} - [\tilde{\boldsymbol{\Psi}} + \tilde{\mathbf{E}}(m)]\boldsymbol{\theta} \right\|_2^2 + \lambda \left\|\boldsymbol{\theta}\right\|_1, \tag{5.38}$$

while given $\boldsymbol{\theta}(m)$ it reduces to a quadratic form with optimal $\tilde{\mathbf{E}}(m+1)$ given by

$$\hat{\tilde{\mathbf{E}}}_{\mathrm{STLS}}(m+1) = \Big(1 + \|\boldsymbol{\theta}(m)\|_2^2\Big)^{-1} [\tilde{\mathbf{y}} - \tilde{\boldsymbol{\Psi}}\,\boldsymbol{\theta}(m)]\boldsymbol{\theta}(m)^T, \tag{5.39}$$

where (m) denotes the m-th iteration. As explained in [95], the CD algorithm tries to find the values of $\tilde{\mathbf{E}}$ as well as $\boldsymbol{\theta}$ which has only K non-zero values corresponding to the K sources. Therefore, we propose to use a two-step algorithm to solve the problem of grid mismatch called STLS-LS. First, we use the CD algorithm to end up with $\boldsymbol{\theta}$. Next, using the indices of the non-zero elements of the recovered $\boldsymbol{\theta}$, i.e., the set $\{z(k)|\ k = 1, \cdots, K\}$, as the location of the GPs, we run the grid mismatch recovery proposed in (5.35). It is worth mentioning that the convergence of the CD algorithm is investigated in [95]. The overall STLS-LS algorithm is summarized in Algorithm 5.1.

More accurate results for $\boldsymbol{\theta}$ can be acquired if we estimate the location of the closest GPs by considering the structure of the perturbations. This becomes even more precise if we have knowledge about the covariance matrix of $\boldsymbol{\Delta}\mathbf{x}$ ($\mathbf{C}_{\boldsymbol{\Delta}\mathbf{x}}$) and $\tilde{\boldsymbol{\epsilon}}$ ($\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}$), assuming that $\mathbf{m}_{\tilde{\boldsymbol{\epsilon}}} = \mathbf{0}$ as is shown in [23]. Note that similar to the case of $\boldsymbol{\epsilon}$ explained at the end of Section 5.4, such information about the statistics of $\boldsymbol{\Delta}\mathbf{x}$ can be computed by considering the fact that the elements of $\boldsymbol{\Delta}\mathbf{x}_{k,z(k)}$ should lie within a cell around the GP with a uniform distribution $\mathcal{U}(-a/2, a/2)$ in each dimension, where a is the length of a square cell. This in turn yields $\mathbf{C}_{\boldsymbol{\Delta}\mathbf{x}} = a^2\mathbf{I}_{2N}/12$. Next, we solve the following weighted structured STLS (WSSTLS) problem for the enhanced model

$$\min_{\boldsymbol{\Delta}\mathbf{x},\tilde{\boldsymbol{\epsilon}},\boldsymbol{\theta}} \boldsymbol{\Delta}\mathbf{x}^T\,\mathbf{C}_{\boldsymbol{\Delta}\mathbf{x}}^{-1}\,\boldsymbol{\Delta}\mathbf{x} + \tilde{\boldsymbol{\epsilon}}^T\,\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\,\tilde{\boldsymbol{\epsilon}} + \lambda\|\boldsymbol{\theta}\|_1, \tag{5.40a}$$

$$\text{s.t. } \tilde{\mathbf{y}} - \tilde{\boldsymbol{\Psi}}\,\boldsymbol{\theta} = \Big[\boldsymbol{\Delta}\tilde{\boldsymbol{\Psi}}\;\boldsymbol{\Delta}\mathbf{X}\Big]\,\boldsymbol{\theta} + \tilde{\boldsymbol{\epsilon}}. \tag{5.40b}$$

By taking the structure of $\tilde{\mathbf{E}}$ into account, we again use (5.34), which helps us to rewrite (5.40) as

$$\min_{\boldsymbol{\Delta}\mathbf{x},\tilde{\boldsymbol{\epsilon}},\boldsymbol{\theta}} \boldsymbol{\Delta}\mathbf{x}^T\,\mathbf{C}_{\boldsymbol{\Delta}\mathbf{x}}^{-1}\,\boldsymbol{\Delta}\mathbf{x} + \tilde{\boldsymbol{\epsilon}}^T\,\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\,\tilde{\boldsymbol{\epsilon}} + \lambda\|\boldsymbol{\theta}\|_1, \tag{5.41a}$$

$$\text{s.t. } \tilde{\mathbf{y}} - \tilde{\boldsymbol{\Psi}}\boldsymbol{\theta} = \boldsymbol{\Delta}\tilde{\boldsymbol{\Psi}}\,\text{diag}(\boldsymbol{\theta}\otimes\mathbf{1}_2)\,\boldsymbol{\Delta}\mathbf{x} + \tilde{\boldsymbol{\epsilon}}. \tag{5.41b}$$

Let us start with $\boldsymbol{\Delta}\mathbf{x}(m)$ (similarly $\boldsymbol{\Delta}\mathbf{X}(m)$) known, which results in

$$\min_{\tilde{\boldsymbol{\epsilon}},\boldsymbol{\theta}} \tilde{\boldsymbol{\epsilon}}^T\,\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\,\tilde{\boldsymbol{\epsilon}} + \lambda\|\boldsymbol{\theta}\|_1, \tag{5.42a}$$

$$\text{s.t. } \tilde{\mathbf{y}} = \Big[\tilde{\boldsymbol{\Psi}} + \boldsymbol{\Delta}\tilde{\boldsymbol{\Psi}}\,\boldsymbol{\Delta}\mathbf{X}(m)\Big]\,\boldsymbol{\theta} + \tilde{\boldsymbol{\epsilon}}, \tag{5.42b}$$

which by substituting $\tilde{\boldsymbol{\epsilon}}$ from (5.42b) in (5.42a) is equivalent to solving the following convex problem (quadratic form regularized by ℓ_1-norm as in LASSO)

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\text{WSSTLS}}(m) = \arg\min_{\boldsymbol{\theta}} \Big(\tilde{\mathbf{y}} - \Big[\tilde{\boldsymbol{\Psi}} + \boldsymbol{\Delta}\tilde{\boldsymbol{\Psi}}\,\boldsymbol{\Delta}\mathbf{X}(m)\Big]\,\boldsymbol{\theta}\Big)^T \times \\ \mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\Big(\tilde{\mathbf{y}} - \Big[\tilde{\boldsymbol{\Psi}} + \boldsymbol{\Delta}\tilde{\boldsymbol{\Psi}}\,\boldsymbol{\Delta}\mathbf{X}(m)\Big]\,\boldsymbol{\theta}\Big) + \lambda\|\boldsymbol{\theta}\|_1,\end{aligned} \tag{5.43}$$

Having $\boldsymbol{\theta}(m)$ in hand, the next step is to solve

$$\min_{\boldsymbol{\Delta}\mathbf{x},\tilde{\boldsymbol{\epsilon}}} \boldsymbol{\Delta}\mathbf{x}^T\mathbf{C}_{\boldsymbol{\Delta}\mathbf{x}}^{-1}\,\boldsymbol{\Delta}\mathbf{x} + \tilde{\boldsymbol{\epsilon}}^T\,\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\,\tilde{\boldsymbol{\epsilon}}, \tag{5.44a}$$

Algorithm 5.2 Mismatch recovery using WSSTLS-LS

1. Run the proposed WSSTLS CD algorithm given by (5.43) and (5.45)
2. Find the indices of the GPs corresponding to the sources
3. Compute $\boldsymbol{\Delta}\tilde{\boldsymbol{\Psi}}$ and solve (5.35) to recover the off-grid locations.

$$\text{s.t. } \tilde{\mathbf{y}} - \tilde{\boldsymbol{\Psi}}\boldsymbol{\theta}(m) = \boldsymbol{\Delta}\tilde{\boldsymbol{\Psi}}\,\text{diag}(\boldsymbol{\theta}(m)\otimes\mathbf{1}_2)\,\boldsymbol{\Delta}\mathbf{x} + \tilde{\boldsymbol{\epsilon}}, \tag{5.44b}$$

which is quadratic in $\boldsymbol{\Delta}\mathbf{x}$ and results in

$$\boldsymbol{\Delta}\hat{\mathbf{x}}_{\text{WSSTLS}}(m+1) = \left[\mathbf{C}_{\boldsymbol{\Delta}\mathbf{x}}^{-1} + \mathbf{S}^T\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\mathbf{S}\right]^{\dagger}\mathbf{S}^T\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\mathbf{q}, \tag{5.45}$$

where $\mathbf{S} = \boldsymbol{\Delta}\tilde{\boldsymbol{\Psi}}\,\text{diag}(\boldsymbol{\theta}(m)\otimes\mathbf{1}_2)$ and $\mathbf{q} = \tilde{\mathbf{y}} - \tilde{\boldsymbol{\Psi}}\boldsymbol{\theta}(m)$. The detailed derivation of (5.45) is explained in Appendix 5.B. All in all, the two-step mismatch recovery procedure by using WSSTLS (called WSSTLS-LS) is summarized in Algorithm 5.2.

5.6 Simulation Results

In this section, we investigate the performance of our proposed sparsity-aware multi-source localization algorithms (SMTL and ESMTL) in terms of the localization accuracy and the number of identifiable sources. To this aim, we consider a wireless network of size 10×10 m^2 divided into $N = 100$ GPs and we consider M APs covering the whole area and K SNs to be simultaneously localized in our simulations. Instead of taking infinite integrals (as in (5.3)), in practice we work with discrete-time signals of limited length and hence the computations of the autocorrelations as well as the cross-correlations will not be ideal as in the derivations of Section 5.2. As a result, the noise terms $n_i(t)$ will not be completely eliminated and will affect our performance through $\boldsymbol{\epsilon}$ and $\tilde{\boldsymbol{\epsilon}}$. Here, we consider a baseband signal (satisfying the properties mentioned in Section 5.2) sampled at $T_s = 1$ms and compute the autocorrelations and cross-correlations during a time-slot of length $T = 1$s. This is equal to recording $N_s = T/T_s = 1000$ samples for our computations. The speed of signal propagation is $\nu = 340$m/s. Meanwhile, we assume that none of the received signals is so weak that it will be considered as noise in $r_i(\Delta)$ and cannot be detected. We define the signal to noise ratio (SNR) at the i-th AP as the ratio of the received signal power to the noise power. Notably, we consider a distance-independent noise on the received signals at the different APs which according to [23, 99] results in an $\boldsymbol{\epsilon}$ on the TDOA measurements specified by its covariance matrix

$$[\mathbf{C}_{\boldsymbol{\epsilon}}]_{i,j} \approx \begin{cases} \frac{3\,T_s^2\,(1+2\text{SNR})}{\pi^2 N_s \text{SNR}^2}, & i = j \\ \frac{3\,T_s^2}{\pi^2 N_s \text{SNR}}, & i \neq j \end{cases}. \tag{5.46}$$

In order to be able to quantitatively compare the performances of the algorithms under consideration, we consider the positioning root mean squared error (PRMSE) defined by PRMSE $= \sqrt{\sum_{p=1}^{P}\sum_{k=1}^{K} e_{k,p}^2/(PK)}$, where $e_{k,p}$ represents the distance between the real location of the k-th source and its estimated location at the p-th Monte Carlo (MC) trial. All simulations are averaged over $P = 100$ MC runs where in each run the sources are deployed on different random locations. In the following simulations, we consider both the uniform grid structure as well as our proposed grid design, where for the former case if it happens that we encounter coincident Δ values, we use the solution proposed in Subsection 5.4.2, i.e., we remove the effect of the corresponding APs from the measurement vector and the map. For the next simulations, whenever we talk about ESMTL, we consider $L = 2, \cdots, 5$ monomial base functions, i.e., $f_{l,i}(.) = g_l(.) = (.)^l$, $l = 1, \cdots, L$ to enhance the proposed SMTL by introducing new rows in $\bar{\mathbf{\Psi}}$. Further, we use the explained orthonormalization technique (using $\mathbf{R}$) to compute $\tilde{\mathbf{\Psi}}$. This way, $\tilde{\mathbf{\Psi}}$ will be of size $5(M-1) \times N = 45 \times 100$, while $\mathbf{\Psi}$ is of size $(M-1) \times N = 9 \times 100$. For all reconstruction problems, we try to find the best λ by cross-validation [67].

For the purpose of comparison, we also simulate the conventional TDOA positioning method proposed in [100] (called TDOA), as well as an optimal constrained weighted least squares method (called TDOA-CWLS) [22]. Notably, both algorithms localize the sources disjointly which gives them an edge over the proposed algorithms but of course this requires that the TDOAs can be exactly assigned to the correct sources. We would like to point out that we do not compare our results with the KNN, the BC, or even semi-definite relaxation (SDR)-based algorithms because the superiority of the ℓ_1-norm minimization approach compared to KNN, BC and SDR-based algorithms for similar contexts (e.g., RSS-based localization) is respectively illustrated in [26] and [93]. Instead, motivated by the consideration of the aforementioned disjoint conventional methods, as a *benchmark*, we compute the Cramér-Rao lower bound (CRLB) [78] for the location of a single source, but averaged over the positions of the multiple sources. The corresponding fisher information matrix (FIM) associated with SN_k can be given by

$$\mathbf{I}_k = \begin{bmatrix} (\frac{\partial \mathbf{y}_k}{\partial x_k})^T \mathbf{C}_{\boldsymbol{\epsilon}}^{-1} (\frac{\partial \mathbf{y}_k}{\partial x_k}) & (\frac{\partial \mathbf{y}_k}{\partial x_k})^T \mathbf{C}_{\boldsymbol{\epsilon}}^{-1} (\frac{\partial \mathbf{y}_k}{\partial y_k}) \\ (\frac{\partial \mathbf{y}_k}{\partial y_k})^T \mathbf{C}_{\boldsymbol{\epsilon}}^{-1} (\frac{\partial \mathbf{y}_k}{\partial x_k}) & (\frac{\partial \mathbf{y}_k}{\partial y_k})^T \mathbf{C}_{\boldsymbol{\epsilon}}^{-1} (\frac{\partial \mathbf{y}_k}{\partial y_k}) \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \text{tr}[\mathbf{C}_{\boldsymbol{\epsilon}}^{-1} \frac{\partial \mathbf{C}_{\boldsymbol{\epsilon}}}{\partial x_k} \mathbf{C}_{\boldsymbol{\epsilon}}^{-1} \frac{\partial \mathbf{C}_{\boldsymbol{\epsilon}}}{\partial x_k}] & \text{tr}[\mathbf{C}_{\boldsymbol{\epsilon}}^{-1} \frac{\partial \mathbf{C}_{\boldsymbol{\epsilon}}}{\partial x_k} \mathbf{C}_{\boldsymbol{\epsilon}}^{-1} \frac{\partial \mathbf{C}_{\boldsymbol{\epsilon}}}{\partial y_k}] \\ \text{tr}[\mathbf{C}_{\boldsymbol{\epsilon}}^{-1} \frac{\partial \mathbf{C}_{\boldsymbol{\epsilon}}}{\partial y_k} \mathbf{C}_{\boldsymbol{\epsilon}}^{-1} \frac{\partial \mathbf{C}_{\boldsymbol{\epsilon}}}{\partial x_k}] & \text{tr}[\mathbf{C}_{\boldsymbol{\epsilon}}^{-1} \frac{\partial \mathbf{C}_{\boldsymbol{\epsilon}}}{\partial y_k} \mathbf{C}_{\boldsymbol{\epsilon}}^{-1} \frac{\partial \mathbf{C}_{\boldsymbol{\epsilon}}}{\partial y_k}] \end{bmatrix}, \tag{5.47}$$

where $\text{tr}(.)$ stands for the trace operator and the elements of $\mathbf{y}_k$ as well as their

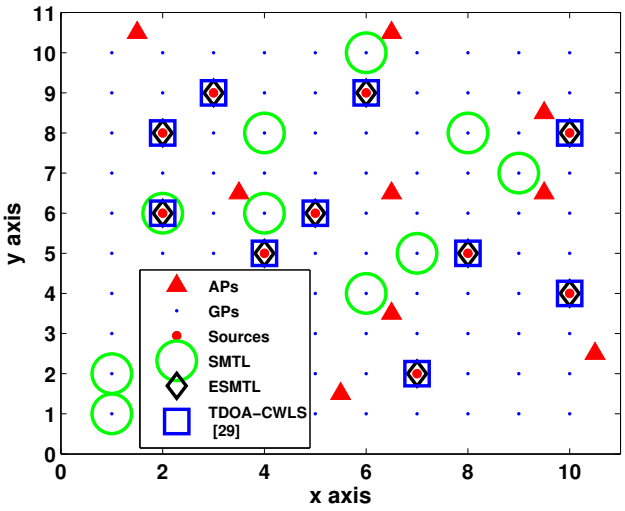


Figure 5.5: Multi-source ($K = 10$) localization with $M = 10$ APs.

derivatives are defined earlier using (5.27)-(5.29). From (5.46), $\mathbf{C}_{\boldsymbol{\epsilon}}$ is independent of the location of the source and hence the second term on the right-hand-side of (5.47) will be equal to zero. Therefore, corresponding to the PRMSE, the total root-CRLB (RCRLB) of the K sources is given by

$$\text{RCRLB} = \sqrt{\frac{\sum_{k=1}^{K} \text{tr}(\mathbf{I}_k^{-1})}{K}}. \tag{5.48}$$

5.6.1 Localization of On-Grid Sources

We start by investigating the performance of the proposed algorithms for the case of on-grid sources. In the first simulation, as shown by Fig. 5.5, we consider $K = 10$ sources randomly deployed over the covered area and $M = 10$ APs which are deployed uniformly at random. The SNR is assumed to be 20dB for all the APs. We recover $\boldsymbol{\theta}$ using both SMTL and ESMTL algorithms and we expect ESMTL to be able to locate more sources simultaneously. This is shown in Fig. 5.5 where SMTL can only localize a single source with minimum error. However, by using the ESMTL algorithm we can locate all the $K = 10$ sources and this clearly illustrates the enhanced performance of ESMTL compared to SMTL. As can be seen, the disjoint TDOA-CWLS is capable of reaching a high accuracy, as well.

This highlights the fact that our ESMTL can perform as good as a disjoint algorithm, which is assisted with signal assignment information and treats the sources separately.

In order to further investigate this improvement in terms of the number of identifiable sources, in Fig. 5.6, we illustrate the PRMSE of localization vs. the number of sources increasing up to $K = 10$. We have $M = 10$ APs and the simulation results of the ESMTL are presented for $L = 2, \cdots, 5$. The SNR is again set to 20dB. As can be seen from the figure, by increasing K, the PRMSE of localization for SMTL increases sharply while ESMTL (with $L = 5$) can handle almost all the sources simultaneously with minimum error. A notable (and expected) observation is that by increasing L from 2 to 5 the potential capability of ESMTL gradually increases from $K = 2$ sources being localized to $K = 10$. The figure also illustrates the considerable improvement of TDOA-CWLS over TDOA which helps it to almost attain the CRLB. Note that we do not plot the results for $K > 10$ sources since for those cases $\boldsymbol{\theta}$ is not really sparse, i.e., we do not have $K \ll N$.

In order to investigate the localization accuracy, we also plot the PRMSE vs. SNR for the same previous setup but with $K = 5$ SNs in Fig. 5.7. As can be seen, increasing the SNR leads to a gradual improvement in the performance of the ESMTL so that for SNR $>$ 5dB we attain zero error. However, SMTL is in principle incapable of localizing $K = 5$ sources simultaneously, as it was also shown in Fig. 5.6, and that is why its performance does not improve with SNR. It is worth mentioning that the performance of the TDOA and TDOA-CWLS schemes is better than the one of ESMTL for lower SNRs. One reason for this is that the conventional approaches are disjoint, i.e., they treat the sources separately. Therefore, the measurement noise does not have any effect on the disambiguation of the sources. However, in the ESMTL, the measurement noise affects the values of the TDOAs (from the cross-correlations) as well as the disambiguation which is solved using ℓ_1-norm minimization. Therefore, the disambiguation (assignment problem) can be badly affected by noise for low SNRs, which can in turn lead to a large error. Another important point worthy of being mentioned is that we attain zero error for SNR $>$ 5dB, which means that we go below the benchmark CRLB. This can be justified by the fact that we consider the on-grid scenario and have a limited number of candidates for the locations of the sources, i.e., the GPs. This feature helps the ℓ_1-norm minimization to exactly locate the sources, as long as the noise is not too strong. More specifically, since within the region of a cell, there is only one possible point for the location of a source, the ℓ_1-norm minimization becomes robust against small noise values.

In the next simulation, we investigate the performance of the ESMTL solved with

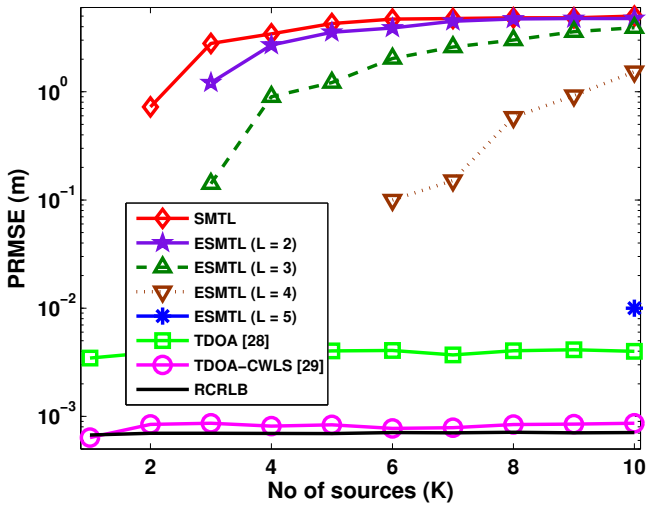


Figure 5.6: PRMSE vs. K for $L = 2, \cdots, 5$ and SNR $= 20$dB. The unplotted data points correspond to zero error in logarithmic scale.

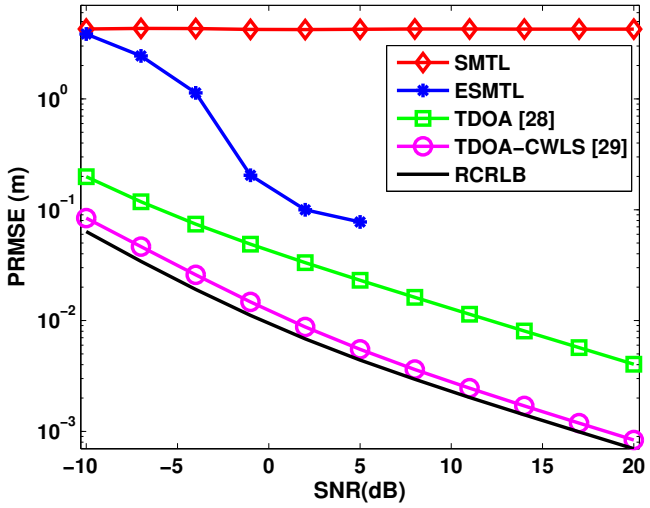


Figure 5.7: PRMSE vs. SNR for $K = 5$. The unplotted data points correspond to zero error in logarithmic scale.

classical LS, which means we have to make sure that $\tilde{\mathbf{\Psi}}$ has full column rank. Hence, we prefer to keep the generated rows instead of removing them for coincident Δ's and use our proposed grid design of Subsection 5.3.3. To simplify

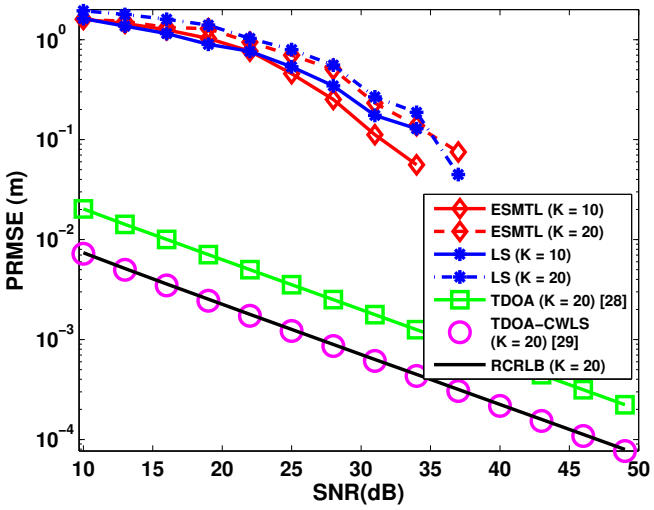


Figure 5.8: PRMSE vs. SNR for $K = 10$ and 20. The unplotted data points correspond to zero error in logarithmic scale.

our simulations and reach a full rank with less complexity, we consider $N = 25$ GPs and only $M = 6$ APs, i.e., we require only $L = 5$ functions as defined earlier (in that case $L(M-1) = 25$). The results are shown in Fig. 5.8, where we consider $K = 10$ and 20 sources. As is clear from the figure, even though with $K/N = 10/25$ (or even further with $K/N = 20/25$) $\boldsymbol{\theta}$ is not sparse anymore, the ESMTL (solved with LS) is capable of localizing the sources with minimum error for SNR > 35dB. However, increasing K increases the probability of wrong Δ computations for a limited bin length N_s and thus leads to a performance degradation for $K = 20$ compared to $K = 10$. As can be seen, the ESMTL (no LS) will still work here but no gain is expected over LS as the problem is not sparse. Obviously, SMTL fails to operate here and is omitted for the sake of clarity. Notably, we observe that for specific AP configurations, it might happen that the newly generated rows with monomials do not necessarily lead to fully independent columns. As there is no restriction on the type of measurement functions, this can be healed to some extent by using different types of nonlinear functions.

5.6.2 Localization of Off-Grid Sources

The following simulations are devoted to the performance evaluation for the case of off-grid sources, i.e., tackling the grid mismatch problem. In Figs. 5.9 and 5.10, the setup is almost the same as in the previous subsection ($M = 10$ APs), except

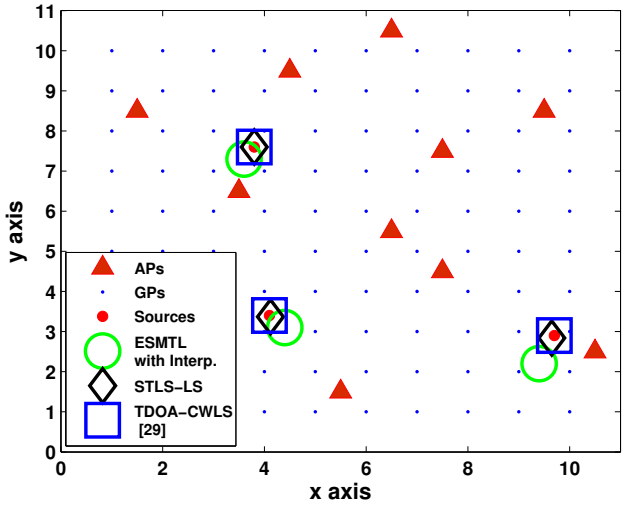


Figure 5.9: Multi-source (K = 3) localization with grid mismatch.

that here we consider a different AP configuration. As is clear from Fig. 5.9, the sources are randomly placed within the cells. The first solution, along the lines of existing literature, consists of using ESMTL (or SMTL) and interpolating the peaks in the recovered $\boldsymbol{\theta}$ as is also used in [95] for a single off-grid source. For the multi-source scenario under consideration, to avoid overlapping peaks in the recovered $\boldsymbol{\theta}$, we have considered less sources (only $K = 3$) and we keep them distant from each other. We expect that if the sources are located far enough from each other, as in this case, we would have 4 peaks in the recovered $\boldsymbol{\theta}$ corresponding to each off-grid source (altogether 12 peaks in the recovered $\boldsymbol{\theta}$ for $K = 3$ sources) and then based on those peaks (shown in Fig. 5.10-(b)) we can conduct a linear interpolation to locate each source (ESMTL-Interp.).

On the other hand, in order to locate the off-grid sources, we use the first proposed approach of Section 5.5 using STLS-LS summarized in Algorithm 5.1. In the first step, the CD algorithm is used to recover a $\boldsymbol{\theta}$ which satisfies (5.25). The recovered $\boldsymbol{\theta}$ is depicted in Fig. 5.10-(a) and as can be seen, the main $K = 3$ peaks correspond to the closest GPs to the sources, i.e, $z(k) = 33, 38, 93$ located on $(4, 3)$, $(4, 8)$ and $(10, 3)$. In the second step, knowing the closest GPs, we compute $\boldsymbol{\Delta\Psi}$ and estimate the mismatch. As is clear from Fig. 5.9, our proposed mismatch recovery algorithm is successful to locate the off-grid sources with a reasonable accuracy and much better than the ESMTL-Interp. A notable observation is that we still face difficulties to resolve two sources located in one cell.

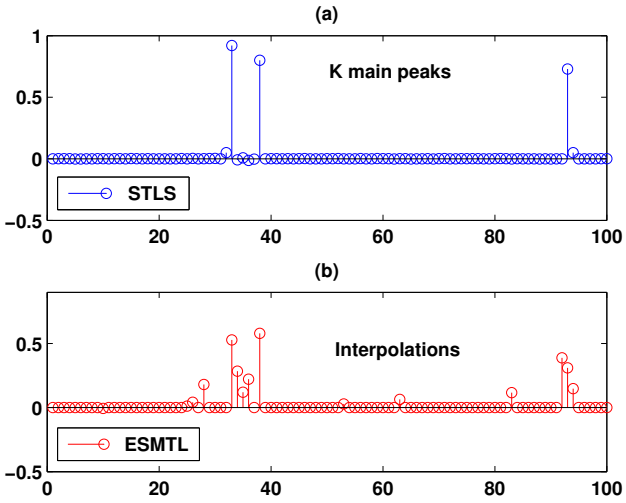


Figure 5.10: Recovered θ with ESMTL and STLS.

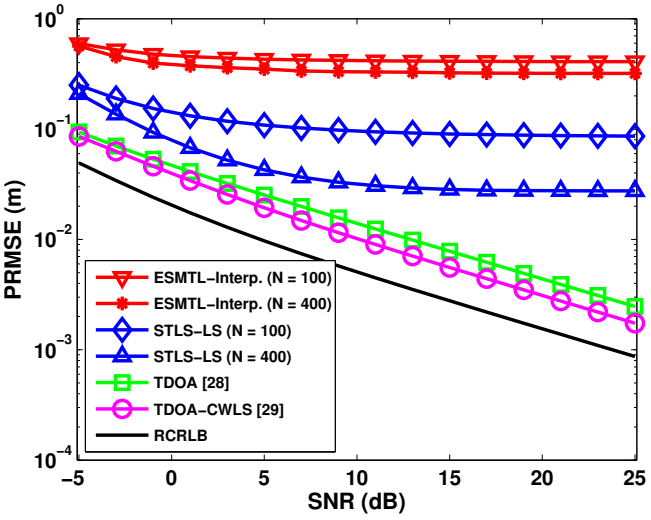


Figure 5.11: Mismatch recovery with ESMTL and STLS.

Finally, Fig. 5.11 illustrates the PRMSE performance vs. SNR for ESMTL-Interp. as well as for the proposed mismatch recovery algorithm STLS-LS when there exist $K = 3$ off-grid sources randomly deployed over the covered area. As can be seen from the figure, while STLS-LS is capable of locating the off-grid sources with a PRMSE of about 9cm for a large span of SNRs, ESMTL-Interp. cannot attain

an accuracy better than 45cm for high SNRs. This stresses the fact that in order to obtain centimeter accuracy, the ESMTL should be modified with the proposed mismatch recovery process for the case of *multiple off-grid* sources. Notably, the conventional disjoint TDOA algorithms (TDOA and TDOA-CWLS) outperform both ESMTL-Interp. and STLS-LS because they are provided with the signal assignment information and they are independent of the GPs and hence indifferent w.r.t. the off-grid effect. We highlight that for more accurate results, the second proposed approach based on WSSTLS-LS (summarized in Algorithm 5.2) can be used, but it is more demanding in terms of computational cost. We would also like to comment on the attainable accuracy of the STLS-LS for large SNRs. As is clear from the figure, the attainable accuracy does not considerably improve with SNR for large SNRs. This effect originates from the 1st-order Taylor expansion. Obviously, the larger the size of the cells, the larger the variations of the TDOA in the cell and hence the worse a 1st-order Taylor expansion will work. This effect can be healed to some extent by decreasing the cell size as is confirmed by the simulation results for $N = 400$ where a PRMSE of 3cm (three times better that $N = 100$) is attained by STLS-LS for large SNRs.

5.7 Conclusions

This chapter tackles the problem of multi-source TDOA localization. We have proposed to simplify the involved issues (i.e., solving hyperbolic equations and multi-source disambiguation) by introducing a novel TDOA fingerprinting and grid design paradigm to convert this non-convex problem to a convex ℓ_1-norm minimization. Moreover, we have proposed a novel trick to enhance the proposed model to be capable of localizing more sources. As a result, we even become able to convert the problem to an overdetermined one which can be efficiently solved using classical LS, if wanted. Finally, in order to extend our ideas, we have proposed two algorithms to handle off-grid sources. Our extensive simulation results corroborate the efficiency of the proposed algorithms in terms of localization accuracy as well as detection capability.

Appendix

5.A The Optimal Operator R

Let us start by rewriting (5.15) as

$$\min_{\mathbf{R}} \left\|(\mathbf{R}\bar{\mathbf{\Psi}})^T(\mathbf{R}\bar{\mathbf{\Psi}}) - \mathbf{I}_N\right\|_F^2 = \min_{\mathbf{\Gamma}} \left\|\bar{\mathbf{\Psi}}^T\mathbf{\Gamma}\bar{\mathbf{\Psi}} - \mathbf{I}_N\right\|_F^2, \tag{5.49}$$

which is due to the fact that the solution of the right-hand side is always symmetric and allows for a decomposition as $\mathbf{\Gamma} = \mathbf{R}^T\mathbf{R}$. By applying $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A})\text{vec}(\mathbf{B})$, with $\otimes$ denoting the Kronecker product, we can further write

$$\text{vec}(\bar{\mathbf{\Psi}}^T\mathbf{\Gamma}\bar{\mathbf{\Psi}}) = (\bar{\mathbf{\Psi}}^T \otimes \bar{\mathbf{\Psi}}^T)\boldsymbol{\gamma}, \tag{5.50}$$

where $\boldsymbol{\gamma} = \text{vec}(\mathbf{\Gamma})$ and $\text{vec}(.)$ denotes the standard vectorization operator. Therefore, (5.15) can be rewritten as the following LS problem

$$\min_{\boldsymbol{\gamma}} \left\|(\bar{\mathbf{\Psi}}^T \otimes \bar{\mathbf{\Psi}}^T)\boldsymbol{\gamma} - \text{vec}(\mathbf{I}_N)\right\|_2^2,$$

with its solution given by

$$\boldsymbol{\gamma} = \left[\bar{\mathbf{\Psi}}^T \otimes \bar{\mathbf{\Psi}}^T\right]^\dagger \text{vec}(\mathbf{I}_N). \tag{5.51}$$

Now, using $(\mathbf{A} \otimes \mathbf{B})^\dagger = \mathbf{A}^\dagger \otimes \mathbf{B}^\dagger$ and $(\mathbf{B}^T)^\dagger = (\mathbf{B}^\dagger)^T$, we can further simplify (5.51) as

$$\boldsymbol{\gamma} = \left[(\bar{\mathbf{\Psi}}^\dagger)^T \otimes (\bar{\mathbf{\Psi}}^\dagger)^T\right] \text{vec}(\mathbf{I}_N).$$

Next, we have

$$\begin{aligned} \mathbf{\Gamma} = \text{ivec}(\boldsymbol{\gamma}) &= \text{ivec}\Big(\left[(\bar{\mathbf{\Psi}}^\dagger)^T \otimes (\bar{\mathbf{\Psi}}^\dagger)^T\right] \text{vec}(\mathbf{I}_N)\Big) \\ &= (\bar{\mathbf{\Psi}}^\dagger)^T\, \mathbf{I}_N\, \bar{\mathbf{\Psi}}^\dagger = (\bar{\mathbf{\Psi}}^\dagger)^T\, \bar{\mathbf{\Psi}}^\dagger, \end{aligned} \tag{5.52}$$

with $\text{ivec}(.)$ denoting the inverse $\text{vec}(.)$ operation, which is indeed a symmetric matrix as claimed earlier. Note that we are now looking for an $\mathbf{R}$ of size $L(M-1) \times L(M-1)$ such that $\mathbf{\Gamma} = \mathbf{R}^T\mathbf{R}$; therefore, the solution is not $\bar{\mathbf{\Psi}}^\dagger$. We need to employ the singular value decomposition (SVD) to decompose $\bar{\mathbf{\Psi}}$ as $\bar{\mathbf{\Psi}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, and thus $\bar{\mathbf{\Psi}}^\dagger = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^T$, which allows us to rewrite (5.52) as

$$\begin{aligned} \mathbf{\Gamma} &= \mathbf{U}(\mathbf{\Sigma}^\dagger)^T\, \mathbf{V}^T\, \mathbf{V}\, \mathbf{\Sigma}^\dagger\, \mathbf{U}^T \\ &= \mathbf{U}(\mathbf{\Sigma}^\dagger)^T\mathbf{\Sigma}^\dagger\mathbf{U}^T = \mathbf{R}^T\mathbf{R}. \end{aligned}$$

Therefore, the desired operator $\mathbf{R}$ of size $L(M-1) \times L(M-1)$ if $L(M-1) \leqslant N$ is given by

$$\mathbf{R} = \boldsymbol{\Sigma}^{\dagger}(1 : L(M-1), :)\mathbf{U}^T, \tag{5.53}$$

while if $L(M-1) > N$, it is given by

$$\mathbf{R} = \begin{bmatrix} \boldsymbol{\Sigma}^{\dagger} \\ \mathbf{0}_{(L(M-1)-N)\times L(M-1)} \end{bmatrix} \mathbf{U}^T. \tag{5.54}$$

This means that $\mathbf{R}\bar{\boldsymbol{\Psi}} = \mathbf{R}\,\mathbf{U}\,\boldsymbol{\Sigma}\,\mathbf{V}^T$ is given by $\mathbf{V}^T(1 : L(M-1), :)$ if $L(M-1) \leqslant N$ or $[\mathbf{V},\ \mathbf{0}_{N\times(L(M-1)-N)}]^T$ if $L(M-1) > N$, which surprisingly means that this is equal to row orthonormalization as proposed in [26]. ■

5.B Computation of the Optimal $\boldsymbol{\Delta x}$

Substituting $\tilde{\boldsymbol{\epsilon}}$ from (5.44b) into (5.44a) while using $\mathbf{q} = \tilde{\mathbf{y}} - \tilde{\boldsymbol{\Psi}}\boldsymbol{\theta}(m)$ and $\mathbf{S} = \boldsymbol{\Delta}\tilde{\boldsymbol{\Psi}}\,\mathrm{diag}(\boldsymbol{\theta}(m) \otimes \mathbf{1}_2)$ leads to minimizing

$$\begin{aligned} J &= \boldsymbol{\Delta x}^T\,\mathbf{C}_{\boldsymbol{\Delta x}}^{-1}\,\boldsymbol{\Delta x} + [\mathbf{q} - \mathbf{S}\,\boldsymbol{\Delta x}]^T\,\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\,[\mathbf{q} - \mathbf{S}\,\boldsymbol{\Delta x}] \\ &= \boldsymbol{\Delta x}^T\mathbf{C}_{\boldsymbol{\Delta x}}^{-1}\boldsymbol{\Delta x} + \mathbf{q}^T\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\mathbf{q} - \mathbf{q}^T\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\mathbf{S}\boldsymbol{\Delta x} \\ &\qquad + \boldsymbol{\Delta x}^T\mathbf{S}^T\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\mathbf{S}\boldsymbol{\Delta x} - \boldsymbol{\Delta x}^T\mathbf{S}^T\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\mathbf{q}. \end{aligned}$$

By taking the partial derivative of J w.r.t. $\boldsymbol{\Delta x}$ and setting it equal to zero we obtain

$$\frac{\partial J}{\partial \boldsymbol{\Delta x}} = 2\mathbf{C}_{\boldsymbol{\Delta x}}^{-1}\,\boldsymbol{\Delta x} - 2\mathbf{S}^T\,\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\,\mathbf{q} + 2\mathbf{S}^T\,\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\,\mathbf{S}\,\boldsymbol{\Delta x} = \mathbf{0},$$

which results in

$$\boldsymbol{\Delta}\hat{\mathbf{x}}_{\mathrm{WSSTLS}}(m+1) = \left[\mathbf{C}_{\boldsymbol{\Delta x}}^{-1} + \mathbf{S}^T\,\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\,\mathbf{S}\right]^{\dagger}\mathbf{S}^T\,\mathbf{C}_{\tilde{\boldsymbol{\epsilon}}}^{-1}\,\mathbf{q}. \quad ■$$

6

SPARSITY-AWARE MULTI-SOURCE RSS LOCALIZATION

Abstract

We tackle the problem of localizing *multiple sources* in multipath environments using received signal strength (RSS) measurements. The existing *sparsity-aware* fingerprinting approaches only use the RSS measurements (autocorrelations) at different access points (APs) separately and ignore the potential information present in the cross-correlations of the received signals. We propose to reformulate this problem to exploit this information by introducing a novel fingerprinting paradigm which leads to a significant gain in terms of number of identifiable sources. Besides, we further enhance this newly proposed approach by incorporating the information present in the other time lags of the autocorrelation and cross-correlation functions. An interesting by-product of the proposed approaches is that under some conditions we can convert the given underdetermined problem to an overdetermined one and efficiently solve it using classical least squares (LS). Moreover, we also approach the problem from a frequency-domain perspective and propose a method which is *blind* to the statistics of the source signals. Finally, we incorporate the so-called concept of finite-alphabet sparsity in our framework for the case where the sources have a similar power. Our extensive simulation results illustrate a good performance as well as a significant detection gain for the introduced multi-source RSS fingerprinting methods.

6.1 Introduction

Precise localization of multiple sources is a fundamental problem which has received a lot of attention recently [84]. Many different approaches have been proposed in literature to recover the location of the sources based on time-of-flight (ToF), time-difference-of-arrival (TDOA) or received-signal-strength (RSS) measurements. A traditional wisdom in RSS-based localization tries to extract distance

information from the RSS measurements. However, this approach fails to provide accurate location estimates due to the complexity and unpredictability of the wireless channel. This has motivated another category of RSS-based positioning, the so-called location fingerprinting, which discretizes the physical space into grid points (GPs) and creates a map representing the space by assigning to every GP a location-dependent RSS parameter, one for every access point (AP). The location of the source is then estimated by comparing real-time measurements with the fingerprinting map at the source or APs, for instance using K-nearest neighbors (KNN) [12] or Bayesian classification (BC) [13].

A deeper look into the grid-based fingerprinting localization problem reveals that the source location is unique in the spatial domain, and can thus be represented by a 1-sparse vector. This motivated the use of compressive sampling (CS) [85] to recover the location of the source using a few measurements by solving an ℓ_1-norm minimization problem. This idea illustrated promising results for the first time in [86, 87] as well as in the following works [26–29, 88, 101, 102]. In [26, 101, 102], a two-step CS-based indoor localization algorithm for multiple targets is proposed. In the first coarse localization step, the idea of cluster matching is used to determine in which cluster the targets are located. This is followed by a fine localization step in which CS is used to recover sparse signals from a small number of noisy measurements. In [27, 88] it is proposed to use a joint distributed CS (JDCS) method in a practical localization scenario in order to exploit the common sparse structure of the received measurements to localize *one* target. Further, for a similar localization scenario as [27, 88], in [28] the encryption capability of CS is demonstrated as CS shows robustness to potential intrusions of unauthorized entities. In [29], finally, a greedy matching pursuit algorithm is proposed for RSS-based target counting and localization with high accuracy.

Although our focus is on RSS-based source localization in this chapter, let us also shortly review some existing sparsity-aware studies in the TDOA domain. Interestingly, not much work can be found on TDOA-based source localization within a sparse representation framework. In [25], single-source TDOA-based localization is proposed wherein the sparsity of the multipath channel is exploited for time-delay estimation. On the other hand, in [93], the source sparsity is exploited to simplify the hyperbolic source localization problem into an ℓ_1-norm minimization. However, the algorithm in [93] treats different sources separately, i.e., it is in principle a single-source localization approach. In [103], we have investigated the problem of sparsity-aware passive localization of multiple sources from TDOA measurements.

Coming back to RSS-based sparsity-aware localization, existing algorithms only make use of the signal/RSS readings at the different receivers (or APs), separately.

However, there is potential information in the cross-correlations of these received signals at the different APs, which has not yet been exploited. In [89], we have proposed to reformulate the sparse localization problem within a *single-path* channel environment so that we can make use of the cross-correlations of the signal readings at the different APs. In this chapter, we extend our basic idea in [89] by presenting the following main contributions:

i) First of all, in contrast to [89], we consider a realistic multipath channel model (simulated by a *room impulse response* (RIR) generator [104]), and we show that our idea can also be employed in a realistic multipath environment.

ii) Second, we analytically show that this new framework can provide a considerable amount of extra information compared to classical algorithms which leads to a significant improvement in terms of the number of identifiable sources as well as localization accuracy. To guarantee a high quality reconstruction, we also numerically assess the restricted isometry property (RIP) of our proposed fingerprinting maps.

iii) In order to further improve the potential of our novel framework in terms of number of identifiable sources, we also propose to exploit extra information in the time domain. Particularly, the information in other lags than the zeroth lag of the autocorrelation and cross-correlation functions can be exploited to construct a larger fingerprinting map.

iv) We propose a novel idea to deal with the cases where there is no knowledge about the statistics of the transmitted signals by the source nodes (SNs). This basically makes it possible to perform fingerprinting in a blind fashion with respect to (w.r.t.) the statistics of the transmitted signals. This blind approach is mainly based on a proper filter bank design to approach the fingerprinting problem from the frequency domain. Moreover, we also show that incorporating this information in the frequency domain improves the performance in terms of number of identifiable sources compared to the original proposed approach.

v) We show that if the sources transmit the same signal power the sparse vector of interest will contain finite-alphabet elements. In such cases, we propose to recover the locations by taking the finite-alphabet property of the non-zero elements of the sparse vector into account, which we refer to as finite-alphabet sparsity. We show that including this information leads to a considerable reconstruction gain.

Note that the sources considered here are *non-cooperative*, i.e., the sources do not emit radio signals with the purpose of localization, but the signals are intended for communications and we exploit them for localization. The proposed algorithms can be applied in indoor or outdoor environments. For instance, monitoring non-cooperative sources broadcasting CDMA signals can be an example of our application domain. However, there is no limitation to employ the proposed ideas in wireless LAN (WLAN) or wireless sensor networks (WSNs) operating in a centralized fashion with a wired backbone.

The rest of this chapter is organized as follows. In Section 6.2, the signal and network model under consideration are explained. In Section 6.3, the classical RSS-based fingerprinting localization as well as our proposed fingerprinting idea are explained. The RIP of the proposed fingerprinting maps is also numerically assessed in this section. Section 6.4 explains how extra information in the time domain can be exploited to further enhance the performance of our proposed fingerprinting idea. The idea of blind fingerprinting using frequency domain information is presented in Section 6.5. Section 6.6 explains the idea of using finite-alphabet sparsity. Extensive simulation results in Section 6.7 corroborate our analytical claims in several scenarios. Finally in Section 6.8, after a short discussion on computational complexity of the proposed algorithms, this chapter is wrapped up with brief concluding remarks.

6.2 Problem Definition

Consider that we have M access points (APs) distributed over an area which is discretized into N cells each represented by its central grid point (GP). The APs can be located anywhere. We consider K *non-cooperative* source nodes (SNs) which are randomly located either on these GPs (*on-grid* scenario) or anywhere (*off-grid* scenario). We assume that the APs are connected to each other by a wired backbone so that they can cooperate by exchanging their signal readings. We also assume that the APs are synchronized, which is feasible especially considering the wired backbone. Now, if the k-th SN broadcasts a time domain signal $s_k(t)$, the received signal at the m-th AP can be expressed by

$$x_{m,k}(t) = \sum_{l=1}^{L} h_{l,m,k} s_k(t - \tau_{l,m,k}) + n_m(t), \tag{6.1}$$

where we consider an L-path channel with $h_{l,m,k}$ and $\tau_{l,m,k}$ respectively denoting the channel coefficient and time-delay of the l-th path from the k-th SN to the m-th

AP; $n_m(t)$ is the additive noise at the m-th AP. Our assumptions on the signal and noise models are as follows:

A.1 The signals $s_k(t)$ are assumed to be ergodic, mutually uncorrelated sequences, i.e., $\mathbb{E}\{s_k(t)\ s^*_{k'}(t')\} = \eta_k r_k(t-t')\delta_{k-k'}$, with η_k being the k-th signal power, $r_k(\tau)$ the normalized signal correlation function with $r_k(0) = 1$, and δ_k the unit impulse function. Meanwhile, $\mathbb{E}\{.\}$ denotes the statistical expectation which is equal to temporal averaging due to the ergodic property of the signals.

A.2 The noises $n_m(t)$ are assumed to be ergodic, mutually uncorrelated white sequences, i.e., $\mathbb{E}\{n_m(t)\ n^*_{m'}(t')\} = \sigma_n^2\delta(t-t')\delta_{m-m'}$, with $\sigma_n^2 = N_0 B$ the variance of the additive noise with density N_0 within the operating frequency bandwidth B, and $\delta(t)$ the Dirac impulse function.

A.3 The transmitted signals are uncorrelated with the additive noise, i.e., $\mathbb{E}\{s_k(t)\ n_m(t')\} = 0,\ \forall t,\ t'$ and $\forall m, k$.

A.4 Throughout this chapter we consider $r_k(\tau) = r(\tau),\ \forall k$ and we assume it to be known a priori or acquired through training, unless otherwise mentioned. Note that under this assumption the SNs can still be considered *non-cooperative* in the sense that they do not cooperate by exchanging information.

A.5 As a more general case, we sometimes also consider $r_k(\tau) \neq r_{k'}(\tau)$ and assume they are unknown. This requires an approach which is *blind* to the $r_k(\tau)$'s.

From (6.1), the total received signal at the m-th AP can be written as

$$x_m(t) = \sum_{k=1}^{K} x_{m,k}(t) = \sum_{k=1}^{K}\sum_{l=1}^{L} h_{l,m,k}\, s_k(t-\tau_{l,m,k}) + n_m(t). \tag{6.2}$$

It is worth pointing out that in a general sense, the problem under consideration is a *passive* localization problem as $x_m(t)$ cannot be be decomposed into its components $x_{m,k}(t)$. The problem here is to use the total received signals at the APs to localize the SNs simultaneously. In the following, we propose a novel RSS-based fingerprinting paradigm to localize the SNs within a multipath environment.

6.3 Sparsity-Aware RSS Localization

Localizing multiple SNs using their received signals is a non-trivial problem which can be converted into a linear problem by taking into account the sparsity of the SNs in the spatial domain. In order to be able to incorporate the sparsity, we define a grid structure in space consisting of N GPs. Next, we perform localization in two phases; first, we construct the fingerprinting map in an initialization phase by either training or if possible analytical computation. More specifically, if training is considered, a training SN (transmitting $s_0(t)$ with signal correlation $r(\tau)$ and power $\eta_0 = 1$) is put on every GP, one after the other, and the signal readings at all the APs are used to construct the map. Alternatively, the channel coefficients and the time-delays of the received signals at all the APs can be computed analytically (e.g., using the RIR generator [104]) whereas the statistics of the $s_k(t)$'s, i.e., the $r(\tau)$, are assumed to be known (or measured) beforehand. Notably, an important advantage of analytically computing the map is avoiding an exhaustive training procedure. In the second phase, the so-called *run-time phase*, real-time multi-source measurements of the sources with similar statistics as in the initialization phase are collected and processed to recover the locations of the SNs.

It is also worth highlighting that the case of *off-grid* source localization can for instance be handled using adaptive mesh refinement algorithms as explained in [24] or by finding the "grid mismatch" using sparse total least squares (STLS) ideas as we proposed in [103], but this is left as future work due to space limitations. In this chapter, we confine ourselves to finding the closest GPs to the off-grid sources as explained in Subsection 6.7.3.

6.3.1 Classical Sparsity-Aware RSS Localization (SRL)

One way to compute the RSS is by taking the zeroth lag of the autocorrelation function of the received time-domain signals at the APs as

$$\begin{aligned} y_m &= \mathbb{E}\left\{x_m(t)\, x_m^*(t)\right\} \\ &= \mathbb{E}\Bigg\{ \left(\sum_{k=1}^{K} \sum_{l=1}^{L} h_{l,m,k} s_k(t-\tau_{l,m,k}) + n_m(t) \right) \\ &\qquad \times \left(\sum_{k'=1}^{K} \sum_{l'=1}^{L} h^*_{l',m,k'} s^*_{k'}(t-\tau_{l',m,k'}) + n^*_m(t) \right) \Bigg\} \end{aligned}$$

$$= \mathbb{E}\left\{\sum_{k=k'=1}^{K}\sum_{l=1}^{L}\sum_{l'=1}^{L} h_{l,m,k}h^*_{l',m,k}s_k(t-\tau_{l,m,k})s^*_k(t-\tau_{l',m,k})\right\} + \mathbb{E}\{n_m(t)n^*_m(t)\}$$

$$= \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{l'=1}^{L} h_{l,m,k}h^*_{l',m,k}r(\tau_{l',m,k}-\tau_{l,m,k})\eta_k + \sigma_n^2, \tag{6.3}$$

which for a single-path channel model boils down to $y_m = \sum_{k=1}^{K}|h_{m,k}|^2\eta_k + \sigma_n^2$. Notably, the third equality follows from A.1 and A.3 and the last equality follows from A.2 and A.4, as detailed in Section 6.2. Interestingly, if we ignore the effect of the noise for the time being, the last expression in (6.3) shows that the RSS at AP_m is a summation of K location-dependent (through delays and channel coefficients) terms $\sum_{l=1}^{L}\sum_{l'=1}^{L} h_{l,m,k}h^*_{l',m,k}r(\tau_{l',m,k}-\tau_{l,m,k})$. This means that if these K components could be recognized, the locations can be estimated from them, which motivates choosing them as *fingerprints* of the sources. Now, in order to be able to do this, we consider that the SNs can only be located on a finite set of positions determined by N GPs. Therefore, if we measure/compute the fingerprints of the N GPs, the corresponding N

$$\boldsymbol{\psi}_m = \Bigg[\sum_{l=1}^{L}\sum_{l'=1}^{L} h^g_{l,m,1}h^{g*}_{l',m,1}r(\tau^g_{l',m,1}-\tau^g_{l,m,1}), \cdots, \sum_{l=1}^{L}\sum_{l'=1}^{L} h^g_{l,m,N}h^{g*}_{l',m,N}r(\tau^g_{l',m,N}-\tau^g_{l,m,N})\Bigg]^T, \tag{6.4}$$

where $(.)^g$ denotes values being measured/computed for the GPs. Thus, using (6.4), (6.3) can be rewritten for a grid structure as

$$y_m = \boldsymbol{\psi}_m^T\boldsymbol{\theta} + \sigma_n^2, \tag{6.5}$$

where $\boldsymbol{\theta}$ is an $N \times 1$ vector containing all zeros except for K non-zero elements with indices related to the locations of the K sources and values equal to the η_k's. The same holds for the other APs with the same $\boldsymbol{\theta}$, which helps us to stack the y_m's and $\boldsymbol{\psi}_m$'s for different APs as

$$\mathbf{y} = \boldsymbol{\Psi}\boldsymbol{\theta} + \mathbf{p}_n, \tag{6.6}$$

where $\mathbf{p}_n = \sigma_n^2\mathbf{1}_M$ with $\mathbf{1}_M$ the $M \times 1$ vector of all ones, $\mathbf{y} = [y_1, \cdots, y_M]^T$ and $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_M]^T$. Defining $\mathbf{x}(t) = [x_1(t), \cdots, x_M(t)]^T$, it is clear that

$$\mathbf{y} = \mathbb{E}\{\mathbf{x}(t) \odot \mathbf{x}^*(t)\}, \tag{6.7}$$

where $\odot$ denotes the element-wise Hadamard product. As is clear from (6.6), $\mathbf{y}$ is the K-sparse RSS characterized by the fingerprinting map $\boldsymbol{\Psi}$ as given by

$$\boldsymbol{\Psi}^T = \sum_{l=1}^{L}\sum_{l'=1}^{L} \left\{ \begin{bmatrix} h^g_{l,1,1}h^{g*}_{l',1,1}r(\tau^g_{l',1,1}-\tau^g_{l,1,1}) & \cdots & h^g_{l,M,1}h^{g*}_{l',M,1}r(\tau^g_{l',M,1}-\tau^g_{l,M,1}) \\ h^g_{l,1,2}h^{g*}_{l',1,2}r(\tau^g_{l',1,2}-\tau^g_{l,1,2}) & \cdots & h^g_{l,M,2}h^{g*}_{l',M,2}r(\tau^g_{l',M,2}-\tau^g_{l,M,2}) \\ \vdots & \ddots & \vdots \\ h^g_{l,1,N}h^{g*}_{l',1,N}r(\tau^g_{l',1,N}-\tau^g_{l,1,N}) & \cdots & h^g_{l,M,N}h^{g*}_{l',M,N}r(\tau^g_{l',M,N}-\tau^g_{l,M,N}) \end{bmatrix} \right\}. \tag{6.8}$$

Note that if the SNs have different signal powers, estimating $\boldsymbol{\theta}$ will also return the signal powers as a by-product. Solving (6.6) with classical LS produces a poor estimate due to the under-determined nature of the problem ($M < N$). Instead, sparse reconstruction techniques (or CS) aim to reconstruct $\boldsymbol{\theta}$ by taking the source sparsity concept into account. It is worth mentioning that here we have a natural compression in the problem, in the sense that the number of measurements is limited to the number of APs (M), which in many practical scenarios is much less than the number of GPs (N). Hence, using (6.6), $\boldsymbol{\theta}$ can be well-recovered by solving the following ℓ_1-norm minimization

$$\hat{\boldsymbol{\theta}}_{\text{SRL}} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \boldsymbol{\Psi}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 , \tag{6.9}$$

where λ is a regularization parameter that controls the trade-off between sparsity and reconstruction fidelity of the estimated $\boldsymbol{\theta}$. The problem (6.9) can efficiently be solved using several algorithms including the well-known LASSO [101]. We would like to stress that (even though modified to fit our setup) the discussed SRL represents the existing classical sparsity-aware RSS localization idea in literature [101] and it is modified and presented here for the sake of comparison.

Remark 6.1 (*Identifiability of SRL*)

To elaborate on the identifiability of localization using SRL, it is worth mentioning that for classical multi-source (2-dimensional) RSS-based localization, as long as there are $M \geqslant 3$ APs (not lying on a straight line), the SNs can be uniquely identified and localized. On the other hand, the sparse reconstruction-

based nature of SRL imposes an extra constraint $M \geqslant 2K$ ($M \geqslant 3$ should also be satisfied) because for a perfect reconstruction we require every $2K$-column subset of $\boldsymbol{\Psi}$ to be full column rank so that we can reconstruct a K-sparse $\boldsymbol{\theta}$. All in all, this leads to the necessary condition $M \geqslant max(2K, 3)$ for identifiability and reconstruction.

6.3.2 Sparsity-Aware RSS Localization via Cooperative APs (SRLC)

As explained in the previous subsection, the existing sparsity-aware RSS-based algorithms represented by the SRL, only make use of the zeroth lag of the autocorrelation function (signal strength) of the signals received at each AP separately and ignore the potential information present in the cross-correlation of this information. We propose to reformulate the problem so that we can exploit this extra information by a cooperation among the APs. This new model requires the construction of a new fingerprinting map as will be explained subsequently. Let us instead of the autocorrelations of the received signals at each AP, this time also compute the cross-correlations as

$$
\begin{aligned}
y_{m,m'} &= \mathbb{E}\left\{x_m(t)\, x^*_{m'}(t)\right\} \\
&= \mathbb{E}\Bigg\{ \left(\sum_{k=1}^{K} \sum_{l=1}^{L} h_{l,m,k} s_k(t-\tau_{l,m,k}) + n_m(t) \right) \\
&\qquad\qquad \times \left(\sum_{k'=1}^{K} \sum_{l'=1}^{L} h^*_{l',m',k'} s^*_{k'}(t-\tau_{l',m',k'}) + n^*_{m'}(t) \right) \Bigg\} \\
&= \mathbb{E}\left\{ \sum_{k=k'=1}^{K} \sum_{l=1}^{L} \sum_{l'=1}^{L} h_{l,m,k} h^*_{l',m',k} s_k(t-\tau_{l,m,k}) s^*_k(t-\tau_{l',m',k}) \right\} \\
&\qquad\qquad + \mathbb{E}\left\{n_m(t) n^*_{m'}(t)\right\} \\
&= \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{l'=1}^{L} h_{l,m,k} h^*_{l',m',k} r(\tau_{l',m',k} - \tau_{l,m,k}) \eta_k + \sigma_n^2 \delta_{m-m'}, \qquad (6.10)
\end{aligned}
$$

which for a single-path channel model boils down to $y_{m,m'} = \sum_{k=1}^{K} h_{m,k} h^*_{m',k}$ $r(\tau_{m',k} - \tau_{m,k})\eta_k + \sigma_n^2 \delta_{m-m'}$. Again, the third equality follows from A.1 and A.3 and the last equality follows from A.2 and A.4, as detailed in Section 6.2. Similar to the case of the SRL, if we ignore the noise effect for the time being, (6.10) again introduces a location-dependent fingerprint $\sum_{l=1}^{L} \sum_{l'=1}^{L} h_{l,m,k} h^*_{l',m',k} r(\tau_{l',m',k} - \tau_{l,m,k})$ for the K sources. Thus, by considering the GPs as the only possible locations of the SNs, if we measure/compute the fingerprints of the N GPs, the

corresponding N fingerprints can be stacked in a vector as given by

$$\tilde{\boldsymbol{\psi}}_{m,m'} = \Bigg[\sum_{l=1}^{L}\sum_{l'=1}^{L} h^{g}_{l,m,1} h^{g*}_{l',m',1} r(\tau^{g}_{l',m,1} - \tau^{g}_{l,m',1}), \\ \cdots, \sum_{l=1}^{L}\sum_{l'=1}^{L} h^{g}_{l,m,N} h^{g*}_{l',m',N} r(\tau^{g}_{l',m,N} - \tau^{g}_{l,m',N}) \Bigg]^{T}, \tag{6.11}$$

and therefore using (6.11), (6.10) can be rewritten for a grid structure as

$$y_{m,m'} = \tilde{\boldsymbol{\psi}}^{T}_{m,m'}\boldsymbol{\theta} + \sigma_n^2 \delta_{m-m'}, \tag{6.12}$$

where $\boldsymbol{\theta}$ is the same K-sparse vector as in the case of the SRL. In order to end up with a similar expression as (6.6), we can stack the M^2 different $y_{m,m'}$'s and $\tilde{\boldsymbol{\psi}}_{m,m'}$'s leading to

$$\tilde{\mathbf{y}} = \tilde{\boldsymbol{\Psi}}\boldsymbol{\theta} + \tilde{\mathbf{p}}_n, \tag{6.13}$$

where

$$\tilde{\mathbf{y}} = [y_{1,1}, \cdots, y_{1,M}, \cdots, y_{M,1}, \cdots, y_{M,M}]^{T}, \tag{6.14}$$

$$\tilde{\boldsymbol{\Psi}} = [\tilde{\boldsymbol{\psi}}_{1,1}, \cdots, \tilde{\boldsymbol{\psi}}_{1,M}, \cdots, \tilde{\boldsymbol{\psi}}_{M,1}, \cdots, \tilde{\boldsymbol{\psi}}_{M,M}]^{T}, \tag{6.15}$$

and $\tilde{\mathbf{p}}_n = \text{vec}(\sigma_n^2 \mathbf{I}_M)$. Clearly, in contrast to $\mathbf{y} = \mathbb{E}\{\mathbf{x}(t) \odot \mathbf{x}^*(t)\}$, this time we compute $\tilde{\mathbf{y}} = \mathbb{E}\{\mathbf{x}(t) \otimes \mathbf{x}^*(t)\}$ where $\otimes$ represents the Kronecker product. Hence, now $\tilde{\mathbf{y}}$ is a K-sparse vector parametrized using a fingerprinting map of size $M^2 \times N$:

$$\tilde{\boldsymbol{\Psi}}^{T} = \sum_{l=1}^{L}\sum_{l'=1}^{L} \left\{ \begin{bmatrix} h^{g*}_{l,1,1} h^{g}_{l',1,1} r(\tau^{g}_{l,1,1} - \tau^{g}_{l',1,1}) & \cdots & h^{g*}_{l,M,1} h^{g}_{l',M,1} r(\tau^{g}_{l,M,1} - \tau^{g}_{l',M,1}) \\ h^{g*}_{l,1,2} h^{g}_{l',1,2} r(\tau^{g}_{l,1,2} - \tau^{g}_{l',1,2}) & \cdots & h^{g*}_{l,M,2} h^{g}_{l',M,2} r(\tau^{g}_{l,M,2} - \tau^{g}_{l',M,2}) \\ \vdots & \ddots & \vdots \\ h^{g*}_{l,1,N} h^{g}_{l',1,N} r(\tau^{g}_{l,1,N} - \tau^{g}_{l',1,N}) & \cdots & h^{g*}_{l,M,N} h^{g}_{l',M,N} r(\tau^{g}_{l,M,N} - \tau^{g}_{l',M,N}) \end{bmatrix} \right\}. \tag{6.16}$$

Remark 6.2 (*Identifiability of SRLC*)

For the enhanced model, we require $M^2 \geqslant 2K$ and $M \geqslant 3$ which results in the necessary identifiability condition $M \geqslant max(\sqrt{2K},\ 3)$. Notably, for the special case where the channel coefficients are real, i.e., $y_{m,m'} = y_{m',m},\ \forall m,\ m'$,

> *we obtain only* $M(M+1)/2$ *different elements in* $\tilde{\mathbf{y}}$ *and the same number of rows in* $\tilde{\boldsymbol{\Psi}}$. *For such a case, we require* $M(M+1)/2 \geqslant 2K$ *and* $M \geqslant 3$ *which results in the necessary identifiability condition* $M \geqslant max(\lceil -1/2 + \sqrt{16K+1}/2 \rceil, 3)$, *where* $\lceil . \rceil$ *denotes the ceiling operator.*

As can be seen, in general, the newly proposed fingerprinting model given by (6.13) provides us with a set of M^2 linear equations instead of only M as in (6.6). This added information ($M^2 - M$ extra rows), obtained by taking cross-correlations of the received signals at the different APs into account, makes it possible for the system to localize a larger number of SNs with a fixed number of APs. This particularly becomes even more important when the physical conditions of the covered area limit the number of possible APs. By considering the statements of Remark 6.1 and Remark 6.2, this gain is illustrated in Fig. 6.1 using the minimum number of APs required to identify K SNs simultaneously. As can be seen, the proposed fingerprinting paradigm is theoretically capable of localizing the same number of SNs with much fewer APs. The new sparsity-aware localization problem in (6.13) can now be solved by considering the following two cases:

– **Case I:** $N > M^2$; In this case, by considering the sparse structure of $\boldsymbol{\theta}$, the extra information enables us to locate more SNs by solving the following

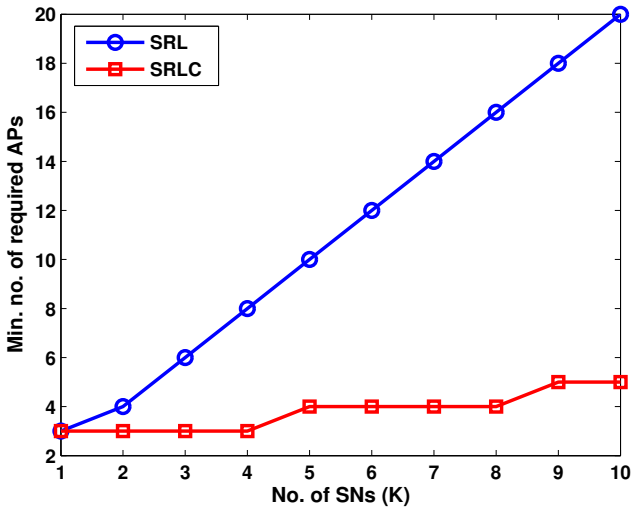


Figure 6.1: Identifiability gain of SRLC compared to SRL

ℓ_1-norm minimization problem (for instance using LASSO):

$$\hat{\boldsymbol{\theta}}_{\text{SRLC}} = \arg\min_{\boldsymbol{\theta}} \left\| \tilde{\mathbf{y}} - \tilde{\boldsymbol{\Psi}}\boldsymbol{\theta} \right\|_2^2 + \lambda \left\| \boldsymbol{\theta} \right\|_1 . \tag{6.17}$$

– **Case II:** $N \leqslant M^2$; Since $\tilde{\boldsymbol{\Psi}}$ has generally full column rank in this case, no matter what the structure of $\boldsymbol{\theta}$ might be, even if it is not sparse, it can be efficiently recovered by ordinary LS as

$$\hat{\boldsymbol{\theta}}_{\text{LS}} = \tilde{\boldsymbol{\Psi}}^{\dagger}\tilde{\mathbf{y}}, \tag{6.18}$$

where $(.)^{\dagger}$ represents the pseudo-inverse.

It is worth pointing out that the idea proposed in this subsection can further be improved by exploiting extra information from the time and frequency domains. This basically motivates Sections 6.4 and 6.5.

6.3.3 RIP Investigation

As we explained earlier, $\boldsymbol{\Psi}$ and $\tilde{\boldsymbol{\Psi}}$ are proved to be the sparsifying bases for the SRL and the SRLC. Having satisfied the sparsity property, the only issue that should be assessed to guarantee a high quality reconstruction is the mutual incoherence between the columns of $\boldsymbol{\Psi}$ and $\tilde{\boldsymbol{\Psi}}$ or alternatively the RIP. One way to approach the problem is following the same trend as explained in [29] because our channel coefficients can often be considered as drawn from a random distribution (such as Rayleigh). As is well-documented in literature [54], for $K = 1, 2, \cdots$ the RIP constant δ_K of a matrix $\mathbf{A}$ (with normalized columns) is the smallest number that satisfies

$$-\delta_K \leqslant \frac{\|\mathbf{A}\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} - 1 \leqslant \delta_K, \tag{6.19}$$

for all K-sparse $\mathbf{x} \in \mathbb{R}^N$. Roughly speaking, as long as $0 < \delta_K < 1$, the RIP holds. In [29], by exploiting the effect of the random channel coefficients it is shown that if $M = O(K \log(N/K))$ the probability that there exists a K-sparse vector that satisfies $|\|\mathbf{A}\mathbf{x}\|_2^2/\|\mathbf{x}\|_2^2 - 1| > \delta_K$ for a $0 < \delta_K < 1$ tends to 0, which means that with a high probability the RIP is satisfied. The same holds in our case for $\boldsymbol{\Psi}$. As an alternative, we have tried to numerically investigate the RIP property of the proposed fingerprinting maps to illustrate that the reconstruction will indeed have a high quality. To this aim, we can use the computationally less demanding definition in [59] where δ_K is defined as the maximum distance from 1 of all the eigenvalues of the $\binom{N}{K}$ submatrices, $\mathbf{A}_{\Lambda}^H\mathbf{A}_{\Lambda}$, derived from $\mathbf{A}$, where Λ is a set of indices with

Table 6.1: RIP Test

Matrix	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6
$\mathbf{N}_{5\times36}$	0	0.8696	1.6167	2.2038	2.7692	3.2472
$\mathbf{\Psi}$	0	0.9820	1.7978	2.5670	3.2589	3.8070
$\mathbf{N}_{25\times36}$	0	0.3442	0.5362	0.6581	0.7291	0.7793
$\tilde{\mathbf{\Psi}}$	0	0.5069	0.6966	0.8065	0.8609	0.9408

cardinality K which selects those columns of $\mathbf{A}$ indexed by Λ. It means that for each K, the RIP constant is given by

$$\delta_K = \max\left(|\lambda_{\max}(\mathbf{A}_\Lambda^H \mathbf{A}_\Lambda) - 1|,\ |\lambda_{\min}(\mathbf{A}_\Lambda^H \mathbf{A}_\Lambda) - 1|\right). \tag{6.20}$$

For the sake of computational feasibility, we consider the case where $M = 5$, and hence $M^2 = 25$, and $N = 36$ to generate a typical $\mathbf{\Psi}$ and $\tilde{\mathbf{\Psi}}$ using the other parameters adopted in Section 6.7. For such a case, we have computed the δ_K with $K = 1, \cdots, 6$ for $\mathbf{\Psi}$ and $\tilde{\mathbf{\Psi}}$. We also compute the δ_K for matrices with the same size containing elements drawn from a random normal distribution, i.e., $\mathbf{N}_{5\times36}$ and $\mathbf{N}_{25\times36}$. Note that such random matrices are proved to be a good choice in terms of the RIP and that is why we use them as a benchmark. In order to slightly heal the RIP, we apply the orthonormalization operation proposed in [26, 103] to all the matrices before testing the RIP. The results are presented in Table 6.1. As is clear from the table, our proposed fingerprinting map for the SRL $\mathbf{\Psi}$ performs almost similar to $\mathbf{N}_{5\times36}$ and loosely satisfies the RIP up to $K = 2$. However, for $K > 2$, δ_K starts increasing for both of them. Interestingly, we observe that for $\tilde{\mathbf{\Psi}}$ (also for $\mathbf{N}_{25\times36}$) the RIP is met for K up to 6, which shows a considerable improvement as compared to $\mathbf{\Psi}$.

6.4 Exploiting Additional Time Domain Information (SRLC-TD)

For a fixed network, with known locations of the APs and GPs, the maximum delay difference can be computed during the initialization phase. It can be expressed by

$$\begin{aligned} \Delta\tau_{\max} &= \max_{m,m',n} \left(|\tau_{m,n} - \tau_{m',n}|\right) \\ &= \max_{m,m',n} \left(\left| \frac{d(\mathrm{AP}_m,\ \mathrm{GP}_n) - d(\mathrm{AP}_{m'},\ \mathrm{GP}_n)}{\nu} \right| \right), \end{aligned} \tag{6.21}$$

where $d(.)$ denotes the Euclidean distance and ν is the velocity of signal propagation. As a result, the maximum delay difference experienced by any signal from a

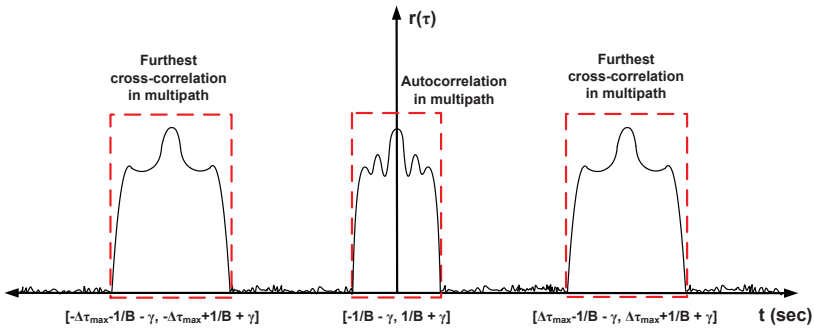


Figure 6.2: Autocorrelations and cross-correlations in a multipath environment

multipath channel is $\Delta\tau_{\max} + \gamma$, where γ denotes the maximum delay spread of the multipath channel. In principle, what we do in Subsection 6.3.2 is to compute the autocorrelations as well as the cross-correlations or in other words the zeroth lag of the autocorrelation and cross-correlation functions. Let us start by explaining what happens when we consider the complete autocorrelation function at each AP and the complete cross-correlation functions of the received signals at the different APs (SRLC-TD). As can be seen in Fig. 6.2, for a multipath channel, when we consider the complete autocorrelation function, the output is non-zero within the time span $[-1/B - \gamma,\ +1/B + \gamma]$. This means that there is potential information present in other lags than the zeroth lag which could further be exploited. Similarly, for the cross-correlations at the different APs, depending on the location of the SNs, we have to scan the time span $[-\Delta\tau_{\max} - 1/B - \gamma,\ \Delta\tau_{\max} + 1/B + \gamma]$ to make sure that we have at least some non-zero $r(\tau)$ values. Particularly, here we are interested in the elements of

$$\tilde{\mathbf{y}}^{(n)} = \mathbb{E}\{\mathbf{x}(t) \otimes \mathbf{x}(t - nT_s)^*\}, \tag{6.22}$$

which are given by

$$\begin{aligned} y_{m,m'}^{(n)} &= \mathbb{E}\left\{x_m(t)\, x_{m'}^*(t - nT_s)\right\} \\ &= \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{l'=1}^{L} h_{l,m,k} h_{l',m',k}^* r(\tau_{l',m,k} - \tau_{l,m',k} + nT_s)\eta_k + \sigma_n^2 \delta_n \delta_{m-m'}, \end{aligned}$$

where T_s is the smallest time fraction in the system which in practice will be the sampling time since we implement the algorithms using temporal averaging.

Therefore, we take $N_s = 1/(T_s B)$ samples per inverse bandwidth. Accordingly, by omitting the intermediate steps similar to the SRLC, we can compute the finger-

printing map for $\tilde{\mathbf{y}}^{(n)}$ as in (6.23) (shown below). As a result, we could consider all lags $n \in \{-N_s - \lceil \gamma/T_s \rceil, \cdots, N_s + \lceil \gamma/T_s \rceil\}$ of the complete auto-correlation functions and all lags $n \in \{\lfloor -(\Delta\tau_{\max} + \gamma)/T_s \rfloor - N_s, \cdots, \lceil (\gamma + \Delta\tau_{\max})/T_s \rceil + N_s\}$ of the complete cross-correlation functions. This way we will compute the auto-correlations for $N^{ac} = 2(N_s + \lceil \gamma/T_s \rceil)$ lags whereas we have to compute cross-correlations for $N^{cc} = 2(N_s + \lceil \Delta\tau_{\max}/T_s \rceil + \lceil \gamma/T_s \rceil)$ lags. Here, for the sake of simplicity of notation, we also assume we compute N^{cc} autocorrelation lags and set the value of the autocorrelation function for the remaining $N^{cc} - N^{ac} = 2\lceil (\Delta\tau_{\max})/T_s \rceil$ lags to zero.

The additional time lags contain new information which was not used in the SRLC. To exploit this potential information, we propose to incorporate all lags by solving

$$\tilde{\mathbf{y}}_{\text{TD}} = \tilde{\mathbf{\Psi}}_{\text{TD}} \boldsymbol{\theta} + \mathbf{1}_{N^{cc}} \otimes (\delta_n \, \tilde{\mathbf{p}}_n), \tag{6.24}$$

where

$$\tilde{\mathbf{y}}_{\text{TD}} = \left[\left(\tilde{\mathbf{y}}^{(\lfloor -(\Delta\tau_{\max}+\gamma)/Ts \rfloor - N_s)} \right)^T, \cdots, \left(\tilde{\mathbf{y}}^{(\lceil (\gamma+\Delta\tau_{\max})/T_s \rceil + N_s)} \right)^T \right]^T,$$

and

$$\tilde{\mathbf{\Psi}}_{\text{TD}} = \left[\left(\tilde{\mathbf{\Psi}}^{(\lfloor -(\Delta\tau_{\max}+\gamma)/Ts \rfloor - N_s)} \right)^T, \cdots, \left(\tilde{\mathbf{\Psi}}^{(\lceil (\gamma+\Delta\tau_{\max})/T_s \rceil + N_s)} \right)^T \right]^T,$$

are the augmented versions of the measurement vectors and fingerprinting maps computed at the different time lags. Hence, this time $\tilde{\mathbf{\Psi}}_{\text{TD}}$ is a $N^{cc}M^2 \times N$ matrix and thus (6.24) can be solved using LASSO or classical LS if it is underdetermined or overdetermined, respectively. It is noteworthy that, depending on the computational complexity constraints, at the expense of the identifiability gain we can also consider the lags to be spaced by the symbol time $1/B$ instead of T_s which would result in a smaller number of lags.

6.5 Blind SRLC Using Frequency Domain Information (SRLC-FD)

Remember that for both SRLC and SRLC-TD, $r(\tau)$ should be the same and known for all the sources (A.4 in Section 6.2) to make us capable of measuring/computing the fingerprinting map. This imposes some a priori knowledge on the problem which might be lacking in some practical situations, and thus we are also interested in an approach which is blind to the $r_k(\tau)$'s. Here, we tackle the issue which is specified by A.5 in Section 6.2, while we also try to take advantage of the large bandwidth of the received signal to gain some extra information, similar to Sec-

$$
\left(\tilde{\boldsymbol{\Psi}}^{(n)}\right)^T = \left\{ \sum_{l=1}^{L} \sum_{l'=1}^{L} \begin{bmatrix} h_{l,1,1}^{g*} h_{l',1,1}^{g} r(\tau_{l,1,1}^{g} - \tau_{l',1,1}^{g} + nT_s) & \cdots & h_{l,M,1}^{g*} h_{l',M,1}^{g} r(\tau_{l,M,1}^{g} - \tau_{l',M,1}^{g} + nT_s) \\ h_{l,1,2}^{g*} h_{l',1,2}^{g} r(\tau_{l,1,2}^{g} - \tau_{l',1,2}^{g} + nT_s) & \cdots & h_{l,M,2}^{g*} h_{l',M,2}^{g} r(\tau_{l,M,2}^{g} - \tau_{l',M,2}^{g} + nT_s) \\ \vdots & \ddots & \vdots \\ h_{l,1,N}^{g*} h_{l',1,N}^{g} r(\tau_{l,1,N}^{g} - \tau_{l',1,N}^{g} + nT_s) & \cdots & h_{l,M,N}^{g*} h_{l',M,N}^{g} r(\tau_{l,M,N}^{g} - \tau_{l',M,N}^{g} + nT_s) \end{bmatrix} \right\}. \tag{6.23}
$$

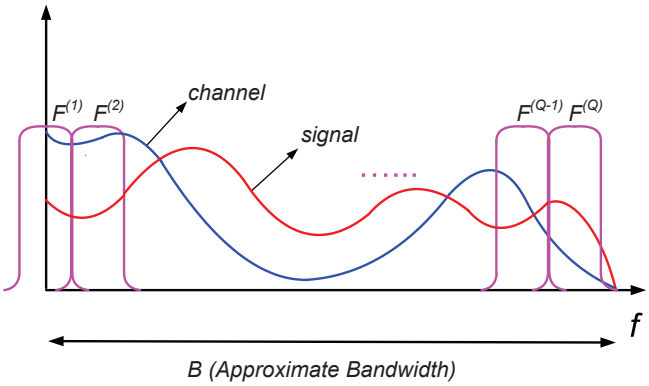


Figure 6.3: Frequency domain filtering. $F^{(q)}$ denotes for the Fourier transform of $f^{(q)}(t)$.

tion 6.4, and enhance the SRLC, this time by approaching the problem from the frequency domain (SRLC-FD).

Let us start by explaining an appropriate filter bank design which plays an important role in the following analysis. Assume that we do not have any knowledge about the $r_k(\tau)$'s. Instead, at each AP we can efficiently estimate the bandwidth of the total received signal using appropriate spectrum estimation techniques [105]; we call it B and for the sake of simplicity of exposure it is assumed to be the same at different APs. Next, we use a set of filters $\{f^{(q)}(t)\}_{q=1}^{Q}$ to divide B into $Q = \lceil B(\Delta\tau_{\max} + \gamma)\rceil$ adjacent subbands $\mathcal{B}^{(q)} = [(q-1)B/Q,\ qB/Q)$ with bandwidth B/Q. A schematic view of an arbitrary signal, channel and the filter bank is shown in Fig. 6.3. Notably, since $B/Q = B/\lceil B(\Delta\tau_{\max} + \gamma)\rceil < 1/\gamma$ with $1/\gamma$ representing the approximate coherence bandwidth of the channel, the output of the q-th filter at the m-th AP experiences a flat fading channel $H_{m,k}^{(q)}$ for every SN_k. Therefore, the related output signal can be written as

$$\begin{aligned} x_m^{(q)}(t) &= \sum_{k=1}^{K} \left[s_k(t) * f^{(q)}(t) \right] H_{m,k}^{(q)} + n_m(t) * f^{(q)}(t) \\ &= \sum_{k=1}^{K} s_k^{(q)}(t)\, H_{m,k}^{(q)} + n_m^{(q)}(t), \end{aligned} \tag{6.25}$$

where $*$ denotes the convolution operator, and $s_k^{(q)}(t)$ and $n_m^{(q)}(t)$ respectively denote the filtered versions of $s_k(t)$ and $n_m(t)$. Further, by simply stacking the results for different APs, the total received signal vector can be expressed as $\mathbf{x}^{(q)}(t) =$

$[x_1^{(q)}(t), \cdots, x_M^{(q)}(t)]^T$. Therefore, we have Q signals $\mathbf{x}^{(q)}(t)$ to compute $\tilde{\mathbf{y}}^{(q)} = \mathbb{E}\{\mathbf{x}^{(q)}(t) \otimes \mathbf{x}^{(q)*}(t)\}$ with its elements given by

$$\begin{aligned} y_{m,m'}^{(q)} &= \mathbb{E}\left\{x_m^{(q)}(t)\, x_{m'}^{(q)*}\right\} \\ &= \mathbb{E}\left\{\sum_{k=1}^{K} s_k^{(q)}(t)\, H_{m,k}^{(q)} \sum_{k'=1}^{K} s_{k'}^{(q)*}(t)\, H_{m',k'}^{(q)*}\right\} + \mathbb{E}\left\{n_m^{(q)}(t) n_{m'}^{(q)}(t)\right\} \\ &= \sum_{k=1}^{K} H_{m,k}^{(q)}\, H_{m',k}^{(q)*}\, \eta_k^{(q)} + \frac{\sigma_n^2}{Q}\, \delta_{m-m'}, \end{aligned} \tag{6.26}$$

where the second equality follows from A.1 and A.3 and the last equality follows from A.2, as detailed in Section 6.2. Now, let us ignore the effect of the noise in (6.26) for the time being, and discover the fingerprints. Interestingly, owing to our proposed filtering, the location-dependent fingerprints $H_{m,k}^{(q)} H_{m',k}^{(q)*}$ do not depend on the $r_k(\tau)$'s and the effect of the different $r_k(\tau)$'s appears in the $\eta_k^{(q)}$'s, which can be handled within the sparse vector of interest. Now, if we consider that the sources can only be located on N GPs, we can use any training or analytical method to compute the $r_k(\tau)$-independent fingerprints at the m-th AP for the q-th subband as

$$\boldsymbol{\psi}_{m,m'}^{(q)} = \left[H_{m,1}^{(q)} H_{m',1}^{(q)*}, \cdots, H_{m,N}^{(q)} H_{m',N}^{(q)*}\right]^T. \tag{6.27}$$

As a result, (6.26) can be rewritten as

$$y_{m,m'}^{(q)} = (\boldsymbol{\psi}_{m,m'}^{(q)})^T\, \boldsymbol{\theta}^{(q)} + \frac{\sigma_n^2}{Q} \delta_{m-m'}, \tag{6.28}$$

where $\boldsymbol{\theta}^{(q)}$ is the K-sparse vector of interest for the q-th subband. The ensuing steps are similar to those of the SRLC and we can compute the fingerprinting map for $\tilde{\mathbf{y}}^{(q)}$ as

$$(\tilde{\boldsymbol{\Psi}}^{(q)})^T = \begin{bmatrix} |H_{1,1}^{(q)g}|^2 & H_{1,1}^{(q)g} H_{2,1}^{(q)g*} & \cdots & |H_{M,1}^{(q)g}|^2 \\ |H_{1,2}^{(q)g}|^2 & H_{1,2}^{(q)g} H_{2,2}^{(q)g*} & \cdots & |H_{M,2}^{(q)g}|^2 \\ \vdots & \vdots & \ddots & \vdots \\ |H_{1,N}^{(q)g}|^2 & H_{1,N}^{(q)g} H_{2,N}^{(q)g*} & \cdots & |H_{M,N}^{(q)g}|^2 \end{bmatrix}.$$

Now, based on this analysis, depending on the statistical properties of the received signals, i.e., spectrum of the $s_k(t)$'s, the following three cases can happen.

6.5.1 Flat Spectrum

Looking at Fig. 6.3, we understand that if the spectrum of the sum of the $s_k(t)$'s is (almost) flat, the $\eta_k^{(q)}$'s will be (almost) the same in the different frequency bands $\mathcal{B}^{(q)}$. This basically makes it possible to construct an augmented version of the measurements as well as the fingerprinting maps, as $\eta_k^{(q)} \approx \eta_k$ will appear again in $\boldsymbol{\theta}^{(q)} = \boldsymbol{\theta}$ for all q. This means $\boldsymbol{\theta}$ will be a K-sparse signal with all elements equal to zero except for K elements equal to η_k. Thus, the ensuing steps are similar to those of the SRLC-TD as by constructing the augmented version of the run-time measurements as $\tilde{\mathbf{y}}_{\text{FD}} = [(\tilde{\mathbf{y}}^{(1)})^T, \cdots, (\tilde{\mathbf{y}}^{(Q)})^T]^T$ and the one of the fingerprinting maps as $\tilde{\boldsymbol{\Psi}}_{\text{FD}} = [(\tilde{\boldsymbol{\Psi}}^{(1)})^T, \cdots, (\tilde{\boldsymbol{\Psi}}^{(Q)})^T]^T$. Finally, we solve

$$\tilde{\mathbf{y}}_{\text{FD}} = \tilde{\boldsymbol{\Psi}}_{\text{FD}}\, \boldsymbol{\theta} + \mathbf{1}_Q \otimes \tilde{\mathbf{p}}_n. \tag{6.29}$$

As we explained, this time $\tilde{\boldsymbol{\Psi}}_{\text{FD}}$ is a $QM^2 \times N$ matrix and thus (6.29) can be solved using LASSO or classical LS if it is underdetermined or overdetermined, respectively. It is worth pointing out that even for the case where the signals have a partially flat spectrum, we can design the filters for that flat part of the spectrum and again construct (6.29) where in such a case we will have less subbands.

6.5.2 Varying Spectrum; The Simple Solution $Q = 1$

In contrast to the case where the signals have a flat spectrum, for the non-flat case, we cannot construct augmented versions of the measurements and the maps for a unique $\boldsymbol{\theta}$ and solve a linear system similar to (6.29). Particularly, because of the different $\eta_k^{(q)}$'s in the different bands, the $\tilde{\boldsymbol{\Psi}}^{(q)}$'s and $\tilde{\mathbf{y}}^{(q)}$'s are related to different $\boldsymbol{\theta}^{(q)}$'s. In this case, as a straightforward solution, we can simply take one of the bands, for instance the first band $\mathcal{B}^{(1)}$, and solve

$$\tilde{\mathbf{y}}^{(1)} = \tilde{\boldsymbol{\Psi}}^{(1)}\, \boldsymbol{\theta}^{(1)} + \tilde{\mathbf{p}}_n. \tag{6.30}$$

This way, we at least have the same identifiability gain as SRLC, but more importantly, we are blind to the $r_k(\tau)$'s. However, we still have some information present in the adjacent subbands which has not been exploited. This motivates the following subsection.

6.5.3 Varying Spectrum; Enhancing the Identifiability Gain

The question is how we can exploit the information present in all the subbands to attain an identifiability gain. An important observation which helps us to develop a

solution is the fact that even though different subbands lead to different $\eta_k^{(q)}$'s for a non-flat spectrum, all the bands construct linear models, similar to (6.30), where in all of them the sparse $\boldsymbol{\theta}^{(q)}$'s share a common support, i.e., the support of $\boldsymbol{\theta}^{(q)}$ is the same $\forall q$. This important property motivates a group-LASSO (G-LASSO) type of solution to incorporate all the bands. However, note that different from classical G-LASSO, we have different maps $\tilde{\boldsymbol{\Psi}}^{(q)}$ for different subbands. Similar cases occur in the framework of the multiple measurement vectors (MMV) problem [106]. To deal with this, we propose a modified version of G-LASSO as defined by

$$\hat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \sum_{q=1}^{Q} \|\tilde{\mathbf{y}}^{(q)} - \tilde{\boldsymbol{\Psi}}^{(q)}[\boldsymbol{\Theta}]_{:,q}\|_2^2 + \lambda \sum_{n=1}^{N} \|[\boldsymbol{\Theta}]_{n,:}\|_2, \tag{6.31}$$

where $\boldsymbol{\Theta} = [\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(Q)}]$. The first term on the right hand side of (6.31) is the LS part which minimizes the error for the different subbands and the second term enforces group sparsity. It is worth pointing out that an analysis of the algorithms to solve (6.31) is outside the scope of this chapter and here we restrict ourselves to standard convex optimization tools such as CVX [107] to solve the problem. Based on the discussions presented in [106] for MMV, incorporating all the subbands within (6.31) will result in a gain in terms of identifiability compared to (6.30), as is also corroborated by our simulation results in Section 6.7.

6.6 Improved Localization Using Finite-Alphabet Sparsity

In particular cases where the SNs have a known equal signal power ($\eta_k = \eta,\ \forall k$) we can accommodate η within $\boldsymbol{\Psi}$ (or $\tilde{\boldsymbol{\Psi}}$, $\tilde{\boldsymbol{\Psi}}_{\text{TD}}$ and $\tilde{\boldsymbol{\Psi}}_{\text{FD}}$) and therefore $\boldsymbol{\theta}$ will be a K-sparse vector with 0 everywhere except for K elements which are 1. This means that our sparse vector (to be reconstructed using LASSO) has a finite-alphabet property which is not included in the optimization problem. Incorporating this extra information can help to improve the reconstruction quality and hence the localization performance for SRL (likewise, SRLC, SRLC-TD and SRLC-FD). The problem of sparse reconstruction under finite-alphabet constraints is investigated in [108, 109]. In [109], efficient algorithms for multiuser detection (MUD) under sparsity and finite-alphabet constraints are developed. More general, sparse reconstruction under finite-alphabet constraints is investigated in [108] through two different approaches; sphere decoding and semi-definite relaxation (SDR), with a main emphasis on the former approach. Here, we re-derive and employ the SDR-based approach. Interestingly, when the alphabet set is $\{0, 1\}$, $\|\boldsymbol{\theta}\|_0 = \|\boldsymbol{\theta}\|_1 = \|\boldsymbol{\theta}\|_2^2$

and $\|\boldsymbol{\theta}\|_1 = \boldsymbol{\theta}^T\mathbf{1} = \mathbf{1}^T\boldsymbol{\theta}$. This helps us to rewrite (6.9) (similarly also (6.17)) as

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}\in\{0,1\}^N} \|\mathbf{y} - \boldsymbol{\Psi}\boldsymbol{\theta}\|_2^2 + \frac{\lambda-\epsilon}{2}(\boldsymbol{\theta}^T\mathbf{1} + \mathbf{1}^T\boldsymbol{\theta}) + \epsilon\,\|\boldsymbol{\theta}\|_2^2\,, \tag{6.32}$$

where $0 < \epsilon \leqslant \lambda$. We can express the right-hand-side of (6.32) in a quadratic form as

$$J(\boldsymbol{\theta}) = \begin{bmatrix}\boldsymbol{\theta}\\ 1\end{bmatrix}^T \underbrace{\begin{bmatrix}\boldsymbol{\Psi}^H\boldsymbol{\Psi} + \epsilon\mathbf{I} & -\boldsymbol{\Psi}^H\mathbf{y} + \frac{\lambda-\epsilon}{2}\mathbf{1}\\ -\mathbf{y}^H\boldsymbol{\Psi} + \frac{\lambda-\epsilon}{2}\mathbf{1}^T & \mathbf{y}^H\mathbf{y}\end{bmatrix}}_{\mathbf{Q}_{\boldsymbol{\theta}}} \underbrace{\begin{bmatrix}\boldsymbol{\theta}\\ 1\end{bmatrix}}_{\tilde{\boldsymbol{\theta}}}. \tag{6.33}$$

Note that minimizing (6.33) is a Boolean quadratic programming problem which permits several efficient algorithms including the quasi-maximum-likelihood SDR of [110]. However, to be able to employ SDR we have to express $J(\boldsymbol{\theta})$ as a function of $\boldsymbol{\alpha} = 2\boldsymbol{\theta} - \mathbf{1} \in \{-1, 1\}^N$. More specifically, after some simplifications we can write $J(\boldsymbol{\theta})$ as $I(\boldsymbol{\alpha}) = \tilde{\boldsymbol{\alpha}}^T\mathbf{Q}_{\boldsymbol{\alpha}}\tilde{\boldsymbol{\alpha}}$ with $\tilde{\boldsymbol{\alpha}} = [\boldsymbol{\alpha}^T,\ 1]^T$ and

$$\mathbf{Q}_{\boldsymbol{\alpha}} = \begin{bmatrix}(\boldsymbol{\Psi}^H\boldsymbol{\Psi} + \epsilon\mathbf{I})/4 & (\boldsymbol{\Psi}^H\boldsymbol{\Psi} + \lambda\mathbf{I})\mathbf{1}/4 - \boldsymbol{\Psi}^H\mathbf{y}/2\\ \mathbf{1}^T(\boldsymbol{\Psi}^H\boldsymbol{\Psi} + \lambda\mathbf{I})/4 - \mathbf{y}^H\boldsymbol{\Psi}/2 & \mathbf{1}^T(\boldsymbol{\Psi}^H\boldsymbol{\Psi} + \lambda\mathbf{I})\mathbf{1}/4 - \mathbf{y}^H\boldsymbol{\Psi}\mathbf{1}/2\\ & -\mathbf{1}^T\boldsymbol{\Psi}^H\mathbf{y}/2 + \mathbf{y}^H\mathbf{y}\end{bmatrix}. \tag{6.34}$$

After relaxing the rank-1 constraint on $\tilde{\mathbf{A}}$ ($\tilde{\mathbf{A}} = \tilde{\boldsymbol{\alpha}}\tilde{\boldsymbol{\alpha}}^T$), we solve the following semi-definite programming (SDP) problem

$$\begin{aligned}\min_{\tilde{\mathbf{A}}} \quad & \text{trace}(\mathbf{Q}_{\boldsymbol{\alpha}}\tilde{\mathbf{A}})\\ \text{s.t.} \quad & \tilde{\mathbf{A}} \succeq \mathbf{0},\\ & [\tilde{\mathbf{A}}]_{i,i} = 1,\ \ i = 1, \cdots, N.\end{aligned}$$

The next step will be to factorize $\tilde{\mathbf{A}}$ to estimate the best $\tilde{\boldsymbol{\alpha}}$ via randomization as explained in [110]. Next, $\hat{\boldsymbol{\theta}}$ can simply be calculated using $\hat{\boldsymbol{\theta}} := (\hat{\boldsymbol{\alpha}} + \mathbf{1})/2$. We expect that including this unused information (finite-alphabet sparsity) within our reconstruction model leads to a performance gain, as is validated by our simulation results. This basically motivates using this model for reconstructing a finite-alphabet sparse $\boldsymbol{\theta}$.

6.7 Numerical Results

In this section, we investigate the performance of the proposed algorithms in terms of probability of detection (P_d), probability of false alarm (P_{fa}) and positioning

root mean squared error (PRMSE) against $1/\sigma_n^2$, the number of existing SNs K and the number of GPs N.

To this aim, we consider a room of size $10 \times 10 \times 3$ m^3 even though our goal is to find the location of the sources on the floor (in 2-D) of size 10×10 m^2. This 2-D area is divided into $N = 100$ cells represented by their central GPs. The APs are randomly placed on the ceiling at a height of 3m and our (up to $K = 10$) non-cooperative sources are considered to be on the floor at a height of 1.8m. Two different scenarios are considered where in the first scenario the sources are randomly placed but they are always on-grid whereas in the second scenario they can be located anywhere, i.e., they can also be off-grid.

The following assumptions about the signal, channel and measurements are respectively in place:

- We consider wideband BPSK signals with a rectangular pulse shape, 3dB bandwidth of $B = 10$MHz and power $\eta = 1$. This means $r(\tau) = 1 - \frac{|\tau|}{B}$ for the baseband equivalent signal. The carrier frequency for the passband signal is 2.4GHz. For all simulations, $r(\tau)$ is assumed to be the same and fixed for all the sources, unless otherwise mentioned. We compute the autocorrelation and cross-correlation functions during a time-slot of length $T = 0.1$ms. This is equal to recording $T \times B = 10^{-4} \times 10^7 = 1000$ BPSK symbols for our computations. Hence, even for moving sources with low dynamics, which is a realistic assumption for the networks under consideration, the length of the time-slot ($T = 0.1$ms) will not put a large constraint on the dynamics of the sources.

- In order to assess the algorithms for a realistic channel model (with no simplifying assumptions), we use synthetic data from the RIR generator provided by [104] for the wireless system explained earlier.

- Instead of taking ideal expectations $\mathbb{E}\{.\}$ in the measurement phase, we work with discrete-time signals of limited length and hence the computations of the autocorrelations as well as the cross-correlations will not be ideal as in the derivations of Section 6.3. As a result, the noise terms $n_m(t)$ will not be completely eliminated in the cross-correlations and they will be an approximation of what is considered for the autocorrelations, and therefore, this will slightly affect our performance. Likewise, the value of the autocorrelations and cross-correlations (in $\mathbf{y}$ or $\tilde{\mathbf{y}}$) will also be approximations of the ideal computations due to this finite-length error.

All simulations are averaged over $P = 100$ independent Monte Carlo (MC) runs where in each run the sources are deployed on different random locations. For all the reconstruction problems, we choose λ by cross-validation as explained in [111]. For the case of on-grid sources, we concentrate on the *detection* performance, i.e., we are only interested to know which elements of the estimated $\boldsymbol{\theta}$ correspond to a source and which elements are zeros, i.e., we only care about the support of $\boldsymbol{\theta}$. Based on this, we define P_{err}, P_d and P_{fa} as follows [112]:

- P_{err} := the probability that a source is detected when the source is in fact *not* present or it is *not* detected when it is in fact present.
- P_d := the probability that a source is detected when the source is in fact present.
- P_{fa} := the probability that a source is detected when the source in fact *not* present.

Basically, P_d and P_{fa} specify all the probabilities of interest. However, we need a detection threshold to be able to compute them. To find the best threshold, we carry out a linear search within the range $[0,\ \max(\hat{\boldsymbol{\theta}}))$ and select the value which minimizes P_{err}. On the other hand, for off-grid sources we plot both P_d and the positioning root mean squared error (PRMSE) defined by

$$\text{PRMSE} = \sqrt{\frac{1}{PK}\sum_{p=1}^{P}\sum_{k=1}^{K} e_{k,p}^2},$$

where $e_{k,p}$ represents the distance between the real location of the k-th source and its estimated location at the p-th MC trial.

Finally, we would like to point out that we do not compare our results with the KNN, the BC, or even semi-definite relaxation (SDR)-based algorithms because the superiority of the ℓ_1-norm minimization approaches (at least for the SRL) compared to KNN, BC and SDR-based algorithms is respectively illustrated in [26, 102] and [93]. Instead, the SRL will be used as the benchmark multi-source RSS-based localization algorithm.

6.7.1 Performance Evaluation with $M = 15$ APs

We start by investigating the performance of the proposed algorithms for the case that there are $M = 15$ APs. In this case, $M^2 = 225 > N$, and therefore, the

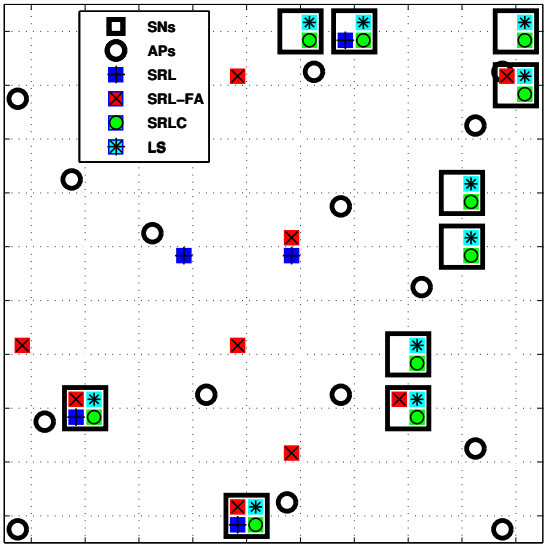


Figure 6.4: Schematic view; $M = 15$, $K = 10$ and $1/\sigma_n^2 = 20$dB

SRLC is expected to perform very well and be capable of recovering $\boldsymbol{\theta}$ with LS too. Note that here LS refers to the classical LS applied within the framework of the SRLC. For the sake of simplicity, we consider the SNs to have equal power, i.e., $\eta_k = \eta \; \forall k$. This allows us to employ and assess the idea of finite-alphabet sparsity to recover $\boldsymbol{\theta}$, as well.

In the first simulation, we consider $K = 10$ sources randomly located on the GPs. As is clear from the schematic view of Fig. 6.4 for $1/\sigma_n^2 = 20$dB, while the SRL can only localize 3 sources, the proposed SRLC (solved by LASSO) and the SRLC (solved by LS) are capable of localizing all the sources. This has motivated us to assess the performance of the SRL solved by the finite-alphabet sparsity idea (we call it SRL-FA) and as is clear from the figure, SRL-FA could localize 4 sources which is improved compared to SRL. Obviously, this improvement also holds for the case of the SRLC with finite-alphabet sparsity; however, since the SRLC is already performing good enough, we do not plot those results. Note that in all the simulations with finite-alphabet sparsity $\epsilon = 0.5\lambda$ and we perform 100 randomization trials.

In order to further investigate the performance of the aforementioned algorithms, we plot the detection and false alarm performance of the algorithms against $1/\sigma_n^2$ as well as the number of existing sources K. In Fig. 6.5, we assume $K = 4$

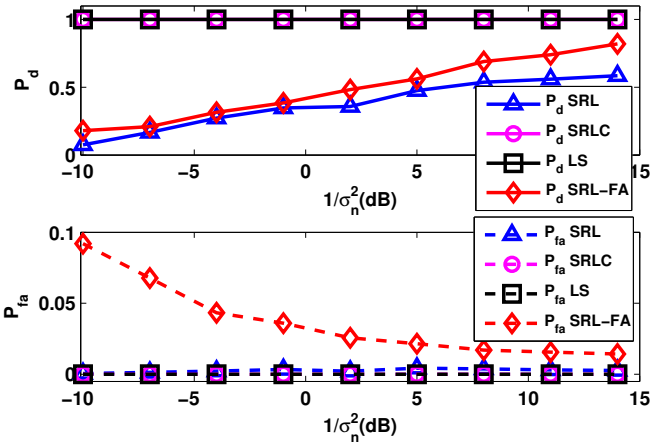


Figure 6.5: Performance vs. $1/\sigma_n^2$ for $M = 15$ and $K = 4$

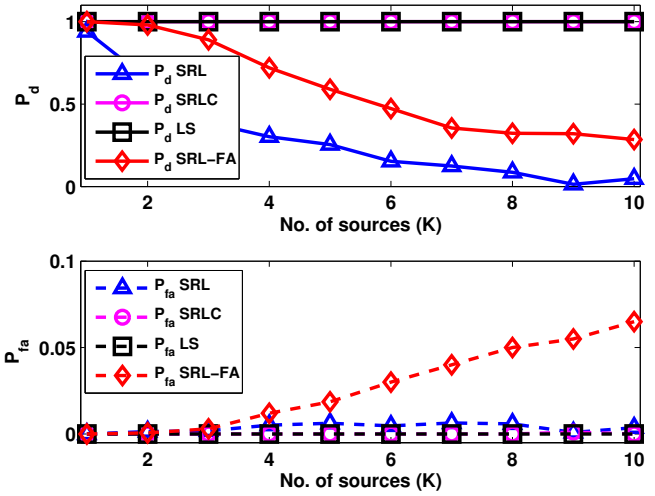


Figure 6.6: Performance vs. K for $M = 15$ and $1/\sigma_n^2 = 20$dB

sources. As is clear from the figure, the SRLC approaches (solved by LASSO and LS) perform very good as they attain $P_d = 1$ and $P_{fa} \approx 0$ for a large span of $1/\sigma_n^2$. The SRL-FA is clearly achieving a better P_d compared to SRL; however, it has a higher P_{fa} as well when its P_d is low. Notably, for all the algorithms, the general trend is an improvement with $1/\sigma_n^2$.

Now, let us get a better understanding by taking a look at the performance of the

algorithms for $1/\sigma_n^2 = 20$dB vs. K in Fig. 6.6. As can be seen, SRLC (in either case) can efficiently localize all the sources while for the SRL the performance drops by increasing the number of sources. The important observation here is that SRL-FA is almost capable of localizing up to $K = 3$ sources with a very high P_d and minimum P_{fa} while this number reduces to $K = 1$ for the stand-alone SRL. However, for $K > 3$ even though the P_d is always better for the SRL-FA, the P_{fa} also increases. Based on the observations in Figs. 6.5 and 6.6, we can conclude that the finite-alphabet sparsity idea is useful for the range $K \leqslant 3$ in this setup. At this point, it is noteworthy that we do not plot the results for $K > 10$ sources since for those cases $\boldsymbol{\theta}$ is not really sparse, i.e., we do not have $K \ll N$.

6.7.2 Further Improvement with $M = 5$ APs and Blindness to $r(\tau)$

In this subsection, we consider the case where we have only $M = 5$ APs available. For such a case, $M^2 = 25 < N$ and thus it is expected that even the SRLC might not be capable of localizing all the $K = 10$ sources. This basically motivates employing the SRLC-TD to incorporate other time lags and hopefully improve the performance over the proposed SRLC. Moreover, this subsection is also meant to investigate the performance of the SRLC-FD algorithm. To this aim, we assume that all the sources have different η_k's with a uniform distribution in the range of $[0.8,\ 1.2]$ and we assume that $r(\tau)$ is unknown to SRLC-FD. This calls for a different fingerprinting map as explained in Section 6.5. We would like to emphasize that SRLC-FD can be employed even for cases where all the sources have different $r_k(\tau)$'s. However, since this cannot be handled by the SRLC and the SRLC-TD, we omit those results here.

Similar to the previous subsection, we consider $K = 10$ sources randomly located on the GPs. Fig. 6.7 depicts a schematic view of localization for $1/\sigma_n^2 = 20$dB. As can be seen, while SRLC is only capable of localizing $K = 2$ sources, the other three enhanced algorithms, i.e., SRLC-TD (solved with LASSO), SRLC-TD (solved with LS) and the blind algorithm (SRLC-FD) could localize all the sources simultaneously. Notably, for the sake of a lower computational complexity, we consider only 6 time lags for the SRLC-TD which are spaced by $1/(2B)$ ($> T_s$) in our simulations. For the SRLC-FD, we have designed $Q = 10$ filters and the proposed G-LASSO solution (explained in Subsection 6.5.3) is employed. It is also worth mentioning that since all the sources have different η_k's, finite-alphabet sparsity is not applicable in this subsection.

As in the previous subsection, we would also like to further assess the proposed algorithms in terms of P_d and P_{fa}. Fig. 6.8 compares the performance of the

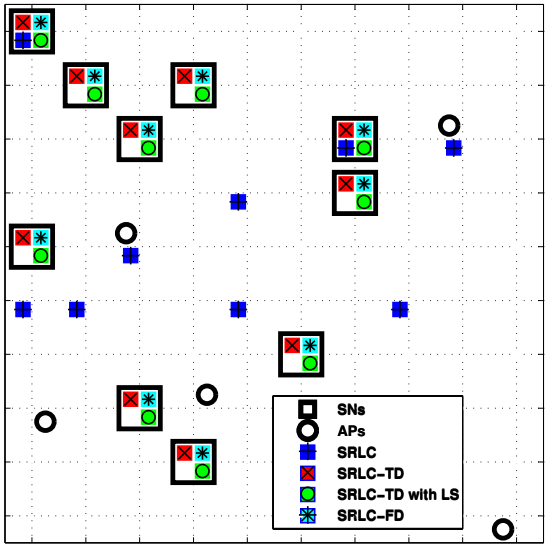


Figure 6.7: Schematic view; $M = 5$, $K = 10$ and $1/\sigma_n^2 = 20$dB

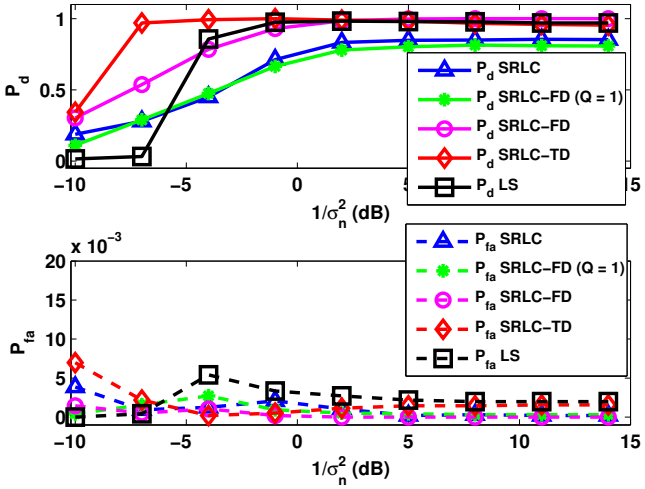


Figure 6.8: Performance vs. $1/\sigma_n^2$ for $M = 5$ and $K = 4$

aforementioned algorithms against $1/\sigma_n^2$ for $K = 4$. SRLC-FD ($Q = 1$) denotes the idea of exploiting only one frequency band as explained in Subsection 6.5.2. As

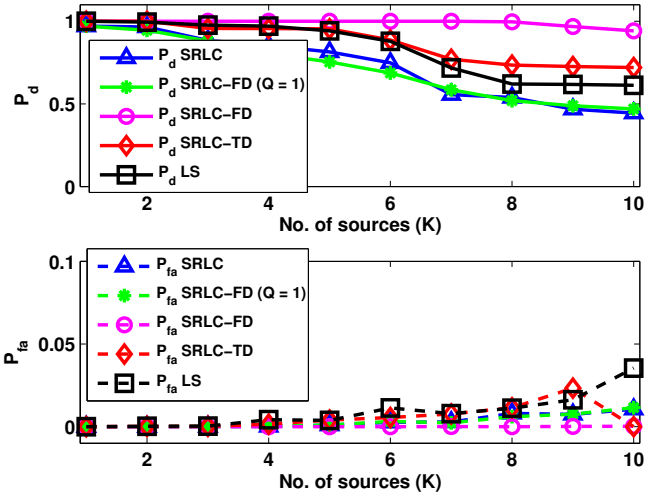


Figure 6.9: Performance vs. K for $M = 5$ and $1/\sigma_n^2 = 20$dB

is clear from the figure, SRLC-FD ($Q = 1$) is performing very close to SRLC while it is blind to $r(\tau)$. Interestingly, SRLC-FD is performing better than SRLC while it is blind. Notably, SRLC-TD (solved with LASSO) and SRLC-TD (solved with LS) both are performing good and attain the best possible performance for $1/\sigma_n^2$ values larger than -1dB. This observation that SRLC-TD is less affected by noise can be justified by referring to (6.24) where we have shown that only measurements in the zeroth time lag are contaminated with noise and the rest of the lags are almost clean.

Let us get a more complete picture of the performance of the algorithms by taking a look at Fig. 6.9 where the detection and false alarm probabilities are depicted against K for $1/\sigma_n^2 = 20$dB. As can be seen, the performance drops for the SRLC and the SRLC-FD ($Q = 1$) with K and thus P_d starts decreasing whereas P_{fa} rises for $K > 3$. Interestingly, for a large enough $1/\sigma_n^2$ (i.e., small enough noise), SRLC-FD attains an optimal performance even for K up to 10. This result corroborates the fact that our blind algorithm with no information about $r(\tau)$, by exploiting the information of the $Q = 10$ frequency subbands could outperform SRLC in terms of the number of identifiable sources. Note that there is a major improvement in SRLC-FD compared to SRLC-FD (Q = 1). The SRLC-TD (both with LASSO and LS) starts degrading for $K \geqslant 5$ which can indeed be improved at the expense of complexity by increasing the number of time lags if the signal and channel properties permit.

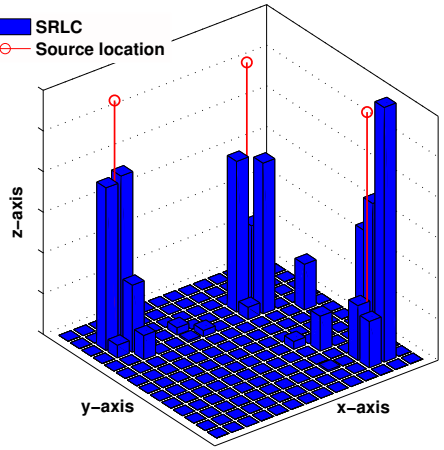


Figure 6.10: 3-D view of $\hat{\boldsymbol{\theta}}_{\text{SRLC}}$ for $M = 7$, $N = 196$, $K = 3$ and $1/\sigma_n^2 = 20$dB

6.7.3 Performance Evaluation for Off-Grid Sources

In this subsection, we intend to investigate the effect of off-grid sources on the performance of the proposed localization paradigm. Having assessed the improvements by exploiting time lags and frequency domain information via respectively SRLC-TD and SRLC-FD, here we only concentrate on the primary algorithm SRLC. Notably, the following off-grid experiments also demonstrate the performance of the SRLC when the measurements are inconsistent with the fingerprinting map. In an off-grid scenario, we expect to observe non-zero values in $\hat{\boldsymbol{\theta}}_{\text{SRLC}}$ corresponding to the GPs around an off-grid source if the channels observed by the neighboring grid points are correlated with the measurements. In order to increase this regional correlation, we should work at lower frequencies and that is why for the following simulations $f_c = 100$MHz and $B = 1$MHz. This means that for the same number of BPSK symbols as before, we have to record $T = 1000/10^6 = 1$ms of the received signals. This is shown in Fig. 6.10 where we depict a 3-D snapshot of $\hat{\boldsymbol{\theta}}_{\text{SRLC}}$ for $M = 7$, $N = 196$, $K = 3$ and $1/\sigma_n^2 = 20$dB. As can be seen, mostly the GPs around the sources return non-zero values which helps us to localize the off-grid sources. Now that we can have continuous locations of the sources in the 2-D area of interest, it makes sense to also plot the PRMSE of our estimates where we only constrain ourselves to finding the nearest GP to the off-grid sources. To further elaborate on the performance, we also plot P_d where a source is considered to be detected if it is estimated to be in a circle with a radius of $\sqrt{2}$ around its real

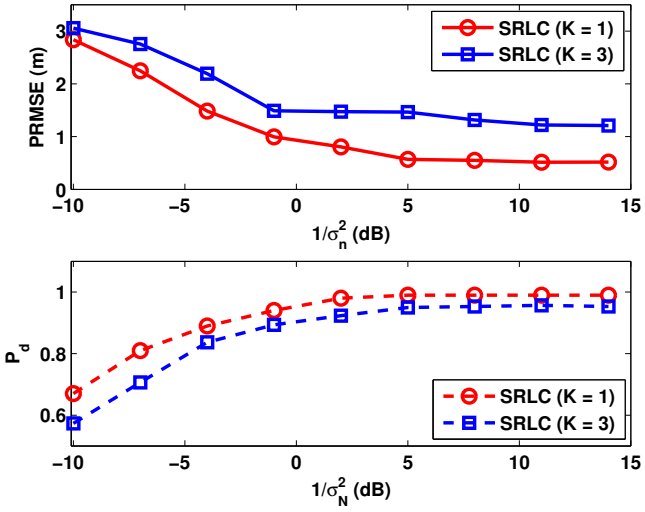


Figure 6.11: Off-Grid Performance vs. $1/\sigma_n^2$; $M = 7$, $K = 1$ and 3

location. To this aim, for the sake of picturing out irrelevant location estimates to achieve a meaningful PRMSE estimate, we consider that we know K and that is why we omit P_{fa} curves. The rest of the parameters is the same as in previous simulations, unless otherwise mentioned.

Fig. 6.11 illustrates the performance against $1/\sigma_n^2$ for $M = 7$ APs with $K = 1$ and 3 SNs randomly located on the floor (at a height of 1.8m) of the room. As can be seen, for a single-source scenario the PRMSE goes below 1m (the cell size) and this means the source can be very-well localized as is corroborated by the corresponding P_d curve. However, for $K = 3$ SNs PRMSE and P_d are slightly degraded. It is worthy of being noted that for the multiple off-grid source localization, the more distant the sources are, the better we can relate the nonzero values of the estimated $\boldsymbol{\theta}$ to the closest GP. This shows a shortcoming of SRLC for localizing off-grid sources which constrains us to artificially avoid the sources to be located in neighboring cells.

Further, in Fig. 6.12, we try to investigate the performance of the SRLC against $N = 36, 64, 100, 144, 196, 324, 484, 676$ and 900 while the room size is kept fixed. The main intention is to assess how an increased correlation between the GPs affects the performance. Note that, however, for a fair comparison in terms of reconstruction (and hence localization), we should also keep the ratio M/N (sometimes called compression rate) constant. In this simulation, we keep a fairly reasonable ratio $M/N = 1/4$. As can be seen, the results are plotted for two differ-

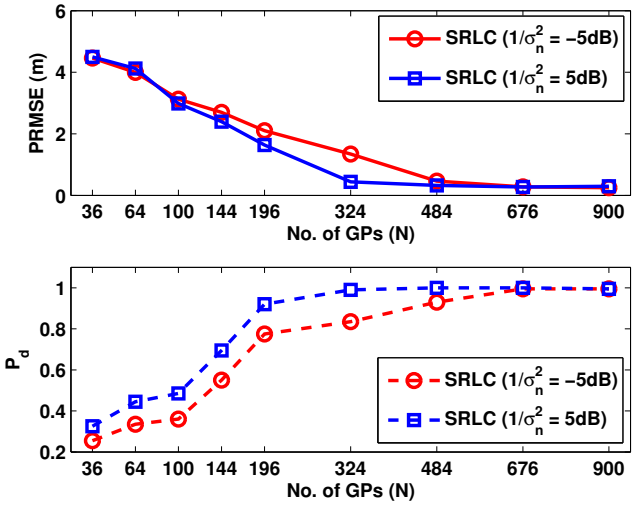


Figure 6.12: Off-Grid Performance vs. N with $K = 2$ for $1/\sigma_n^2 = -5$dB and 5dB

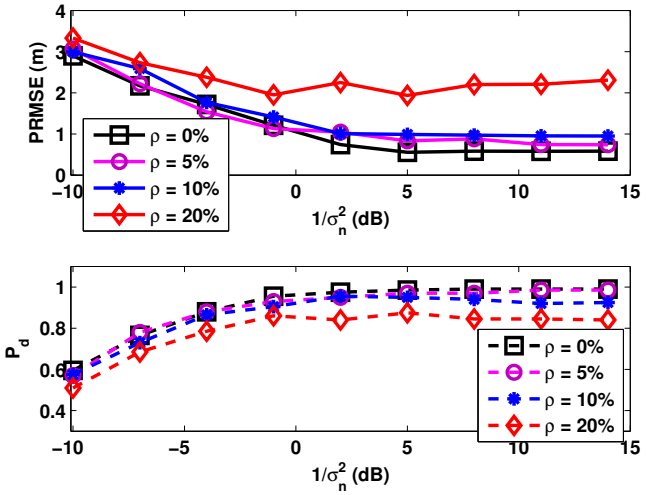


Figure 6.13: Off-Grid Performance vs. $1/\sigma_n^2$ for $\rho = 0\%, 5\%, 15\%$ and 20%

ent noise levels $1/\sigma_n^2 = -5$dB and 5dB. As expected the performance is relatively better in the lower noise level. However, even with $N = 900$, the correlation between the columns of the dictionary is not so severe to spoil the reconstruction, and the performance keeps improving with N. We would like to highlight though that further increasing K will indeed lead to a situation where the RIP will be dras-

tically affected and SRLC will fail. In principle, this is an inherent limitation of any sparsity-aware localization algorithm which should be taken into account at the preliminary system level design.

Finally, we assess the sensitivity of the SRLC w.r.t. perturbations in the trained/computed fingerprinting map $\tilde{\mathbf{\Psi}}$. Such perturbations can for instance happen due to variations in the environment during the run-time phase. To this aim, a perturbation matrix $\mathbf{\Delta}$ drawn from a complex random Gaussian distribution is added to $\tilde{\mathbf{\Psi}}$. Accordingly, a perturbation ratio ρ is defined by $\rho = \|\mathbf{\Delta}\|/\|\tilde{\mathbf{\Psi}}\|$ which is set to 0% (no perturbation), 5%, 10% and 20% in our simulations. As can be seen from Fig. 6.13, the perturbations show their effect mostly in the lower $1/\sigma_n^2$'s. Particularly, for $K = 2$, ρ's up to 10%, and $1/\sigma_n^2 \geqslant 5$dB, the same localization accuracy (less than 1m and $P_d = 1$) as when there is no perturbation can be attained. However, increasing ρ to $\rho > 10\%$ leads to a performance degradation even for high $1/\sigma_n^2$'s. It is noteworthy that this experiment illustrates that our proposed idea can even work when all three model non-idealities simultaneously exist, i.e, measurement noise, off-grid sources and a slightly varying environment.

6.8 Computational Complexity and Conclusions

Before concluding this chapter, we would like to comment on the complexity of the proposed approaches (SRLC, SRLC-TD, SRLC-FD) compared to the classical approach (SRL). Obviously, the enhanced source detection capability of the proposed approaches comes at a price and that is increased complexity. The proposed approaches (SRLC, SRLC-TD and SRLC-FD) respectively require a larger dictionary of size $M^2 \times N$, $N^{cc}M^2 \times N$, and $QM^2 \times N$ compared to the smaller one of SRL of size $M \times N$. Solving our sparse reconstruction problems using LASSO or similarly basis pursuit denoising (BPDN) using the approach of [113] for example requires a complexity that is linear in the number of rows of the dictionary. Therefore, the aforementioned algorithms are respectively M, MN^{cc}, and QM times more demanding in terms of computational cost than the SRL.

This chapter studies the problem of localizing *multiple sources* using their RSS measurements in multipath environments. We have proposed a novel fingerprinting paradigm to exploit the information present in the cross-correlations of the received signals at the different APs which is ignored in existing sparsity-aware fingerprinting approaches. Besides, we have also proposed to further enhance the novel paradigm by incorporating other lags than the zeroth lag of the auto-correlation/cross-correlation functions. Moreover, we have extended our proposed idea to be able to operate when we are blind to the statistics of the source signals.

Finally, we have employed the concept of finite-alphabet sparsity in our framework to deal with the sparse vectors of interest, if they contain finite-alphabet elements. Our extensive simulation results corroborate the efficiency of the proposed algorithms in terms of localization accuracy as well as detection capability.

7

SPARSITY-AWARE MULTIPLE MICROSEISMIC EVENT LOCALIZATION BLIND TO THE SOURCE TIME-FUNCTION

Abstract

We consider the problem of simultaneously estimating three parameters of multiple microseimic events, i.e., the hypocenter, moment tensor, and origin-time. This problem is of great interest because its solution could provide a better understanding of reservoir behavior and can help to optimize the hydraulic fracturing process. The existing approaches employing spatial source sparsity have advantages over traditional full-wave inversion-based schemes; however, their validity and accuracy dependents on the knowledge of the source time-function, which is lacking in practical applications. This becomes even more challenging when multiple microseimic sources appear simultaneously. To cope with this shortcoming, we propose to approach the problem from a frequency-domain perspective and develop a novel sparsity-aware framework which is blind to the source time-function. Through our simulation results with synthetic data, we illustrate that our proposed approach can handle multiple microseismic sources and can estimate their hypocenters with an acceptable accuracy. The results also show that our approach can estimate the normalized amplitude of the moment tensors as a by-product, which can provide worthwhile information about the nature of the sources.

7.1 Introduction

Microseismic event monitoring is a fundamental problem that has received an upsurge of attention in literature. Parameter estimation of microseismic events (also called sources), i.e., estimating their hypocenter, moment tensor components, and origin-time, provides important information about volumetric stress/strain and failure mechanisms in reservoirs [114]. This parameter estimation is also of special interest for earthquake monitoring in seismically active areas [115], for hazard mit-

igation in mining operations [116] and for monitoring and assessing the amount of adjustments during and after a hydraulic fracturing process [117], to name a few.

Most of the previous studies in this context are based on fast inversion or full-wave inversion [118] which suffer from the following main shortcomings: they cannot provide a simultaneous estimate of the three source parameters, they are mostly single-source algorithms, and they are not real-time because of the large bulk of measured seismic traces they have to deal with. Moreover, most of these methods include iterative algorithms which are sensitive to an appropriate initialization. All these issues motivated researchers to think about grid-based approaches [119] where run-time measurement traces are compared with a pre-constructed database of seismic traces also known as a dictionary. On the other hand, constructing such a dictionary requires an extra computational effort.

A deeper look into the grid-based problem reveals that (in a single-source setup) the source hypocenter is unique in the spatial domain, and can thus be represented by a 1-sparse (containing only one non-zero element) vector. This motivated the use of compressive sampling [85] to recover the hypocenter of the source using a few measurements by solving an ℓ_1-norm minimization problem. This idea illustrated promising results for the first time in [87] for localization in a signal processing context, and also in some subsequent studies on multi-source localization [26, 103, 120], where multiple sources could occur at the same time and the received signals could not be decomposed according to their respective sources.

Recently, in a geophysical context, similar ideas have been employed to simultaneously recover the aforementioned three source parameters. In [121], a sparse representation framework is proposed to model the microseismic source activities and it is shown that employing sparse reconstruction techniques makes it possible to jointly estimate the source parameters with an acceptable accuracy. In [122], the same ideas as in [121] are presented; however, by applying a further compression step (leading to a compressive sensing framework) it is shown that the proposed framework in [121] becomes real-time and considerably less demanding in terms of computational cost. In [123], compressive sensing is combined with migration-based techniques to simultaneously estimate the three source parameters. The resulting migration-based problem is then analyzed in the frequency domain. Notably, handling multiple microseismic sources has not been explicitly considered in [121–123]. We should further emphasize that handling a multi-source setup in the frequency domain, as we develop here, calls for a structured approach which has not been derived in [123].

The validity of the sparsity-aware approach presented in [121] and [122] relies

heavily on whether a good estimation of the source time-function is available. More specifically, the approach of [121, 122] only works if there exists one (or more) source(s) with a source time-function exactly the same as the one used to construct the dictionary. Practically speaking, this is a rather hard constraint because different sources have different natures and thus different source time-functions; this limits the application domain of this approach. The same holds for [123] when it comes to handling multiple sources. To overcome this limitation, in this chapter, we propose a novel idea to eliminate this crucial need for the knowledge of the source time-function by approaching the problem from the frequency domain. We show that our proposed approach is capable of estimating the hypocenter of multiple microseismic sources with a high accuracy. The results are also promising in the sense that they motivate a further study to extract the other parameters, i.e., exact moment tensor components and source origin-times.

This chapter is structured as follows. In Section 7.2, we explain the acquisition setup and signal model under consideration. Section 7.3 briefly reviews the proposed approach in [121, 122]. Next, our proposed frequency-domain approach (blind to the source time-function) is explained. Section 7.4 illustrates several simulation results and finally this chapter is summarized in Section 7.5 by discussing a few possible future research directions.

7.2 Acquisition Geometry and Signal Model

An area of interest (normally 3-D, in x, y and z), which is prone to microseismic events (e.g., fractures) is discretized into N grid points. These grid points are the potential candidates for the hypocenter of a microseismic event. The area of interest lies somewhere underground in the vicinity of a well. Traditionally, the grid structure is chosen to be a uniform one with a fixed grid spacing, even though a non-uniform structure (depending on the properties of the area) can also be considered. The other components of our acquisition system are the geophones used to measure the displacements in 3-D; we consider L of them in total. Geophones can be arranged in the form of multiple linear (horizontal or vertical) arrays in the traditional way or they can be more arbitrarily distributed, either on the surface or buried underground.

The phenomena of interest, as explained earlier, are microseismic events, which we model by a time-dependent moment tensor $\mathbf{M}(t)$. Quite often it is assumed in seismology that the time variation of the moment tensor can be separated from its geometry (see [124] and [125]) which leads to $\mathbf{M}(t) = \mathbf{M}s(t)$ with $s(t)$ defined as the source time-function and for a general seismic source (three orthogonal linear

dipoles) $\mathbf{M}$ is specified by a symmetric tensor of rank 2 given by [124]

$$\mathbf{M} = \begin{bmatrix} m_{xx} & m_{xy} & m_{xz} \\ m_{yx} & m_{yy} & m_{yz} \\ m_{zx} & m_{zy} & m_{zz} \end{bmatrix}. \tag{7.1}$$

Now, by considering the 6 diagonal and upper-diagonal elements of $\mathbf{M}$, the n-th component of the displacement at time t measured at a geophone located at $\mathbf{x}$ from a source located at $\boldsymbol{\zeta}$ can be computed by

$$\begin{aligned} u_n(\mathbf{x}, t) &= \sum_{pq} m_{pq}(t) * \frac{\partial}{\partial \zeta_q} G_{np}(\mathbf{x},\ \zeta,\ t,\ \tau) \\ &= \sum_{pq} m_{pq}\, s(t) * \frac{\partial}{\partial \zeta_q} G_{np}(\mathbf{x},\ \zeta,\ t,\ \tau), \end{aligned} \tag{7.2}$$

where $\frac{\partial}{\partial \zeta_q} G_{np}(.)$ denotes the spatial derivative of the Green's function characterizing the medium between the n-th component of the geophone and the p-th component of the source hypocenter with respect to the q-th component of the source hypocenter. Notably, the n, p and q indices denote x, y or z. Further, τ denotes the source origin-time, and $*$ stands for the time-domain convolution. We consider up to K simultaneous microseismic sources to appear within each measurement time interval. As a convention, from now on, we simply use the term source instead of microseismic event/source.

7.3 Sparsity-Aware Parameter Estimation

The idea behind involving sparse reconstruction is the fact that in practice the number of simultaneous sources K is much smaller than the total number of grid points N. In order to incorporate this spatial source sparsity, the received time-domain displacement traces at the different geophones from all possible candidate source hypocenters (grid points) are simulated (or measured) to construct a dictionary of displacement traces. In a dictionary learning context, this is sometimes called the "training phase". Next, in the so-called "run-time phase", the real-time received displacements are compared with the content of the pre-constructed dictionary to estimate the unknown parameters of interest; i.e., moment tensor components, source hypocenter and source origin-time. To carry out this comparison, the embedded sparsity is promoted by introducing the ℓ_1-norm and by taking into account the group structure of the variables involved. The resulting reconstruction problem

will then be solved using the group least absolute shrinkage and selection operator (G-LASSO) [69] or alternatively with the block orthogonal matching pursuit (BOMP) [126]. This method has already been studied in [121] and [122] for our application of interest; however, their approach suffers from the following practical limitation.

Motivation: In [121, 122], the dictionary is highly dependent on the source time-function $s(t)$, which means the source in the run-time phase should have the same source time-function as the one which is considered to construct the dictionary denoted as $s_0(t)$. The situation gets even worse for the multi-source case where $s_k(t) = s_0(t)\ \forall k$ (with $s_k(t)$ the source time-function of the k-th source) should hold to avoid poor results. This is difficult to achieve in practice as the sources might have a different nature and thus a different $s(t)$. This motivated us to think about a novel multi-source sparsity-aware framework, which does not rely on the knowledge of $s(t)$; or let us say it is blind to $s(t)$. Interestingly, a solution exists and can be developed by approaching the problem from the frequency domain as explained in the following.

Let us start by looking at the frequency-domain representation of (7.2). To do so, we sample the time-domain displacement traces with a sampling frequency of F_s ($F_s = 1/T_s$, with T_s the sampling interval) and take a discrete Fourier transform (DFT) of length N_t to obtain

$$\tilde{u}_n(\mathbf{x},\omega) = \sum_{pq} m_{pq}\, \frac{\partial}{\partial \zeta_q}\, \tilde{G}_{np}(\mathbf{x},\ \boldsymbol{\zeta},\ \omega)\, \tilde{s}(\omega)\, e^{j\omega\tau}, \tag{7.3}$$

where $\omega = 2\pi\, F_s\, i/N_t$ with $i = 0, 1, \cdots, N_t - 1$, and $\tilde{(.)}$ emphasizes that we deal with a frequency-domain representation. Note that the time convolution is converted to a (sample by sample) product in the frequency domain. Now, we take N_f frequencies ω_f, with $f = 1, \cdots, N_f$ from the set of N_t frequencies. This allows us to construct the matrix form for different f's given by

$$\underbrace{\begin{bmatrix} \tilde{u}_x(\mathbf{x},\omega_f) \\ \tilde{u}_y(\mathbf{x},\omega_f) \\ \tilde{u}_z(\mathbf{x},\omega_f) \end{bmatrix}}_{\tilde{\mathbf{u}}(\mathbf{x},\omega_f)} = \underbrace{\tilde{s}(\omega_f) \begin{bmatrix} \frac{\partial}{\partial \boldsymbol{\zeta}_x}\tilde{\mathbf{G}}_{xx} & \frac{\partial}{\partial \boldsymbol{\zeta}_y}\tilde{\mathbf{G}}_{xx} & \cdots & \frac{\partial}{\partial \boldsymbol{\zeta}_z}\tilde{\mathbf{G}}_{xz} \\ \frac{\partial}{\partial \boldsymbol{\zeta}_x}\tilde{\mathbf{G}}_{yx} & \frac{\partial}{\partial \boldsymbol{\zeta}_y}\tilde{\mathbf{G}}_{yx} & \cdots & \frac{\partial}{\partial \boldsymbol{\zeta}_z}\tilde{\mathbf{G}}_{yz} \\ \frac{\partial}{\partial \boldsymbol{\zeta}_x}\tilde{\mathbf{G}}_{zx} & \frac{\partial}{\partial \boldsymbol{\zeta}_y}\tilde{\mathbf{G}}_{zx} & \cdots & \frac{\partial}{\partial \boldsymbol{\zeta}_z}\tilde{\mathbf{G}}_{zz} \end{bmatrix}}_{\tilde{\boldsymbol{\Psi}}(\mathbf{x},\ \boldsymbol{\zeta},\ \omega_f)} \underbrace{\begin{bmatrix} m_{xx} \\ m_{xy} \\ \vdots \\ m_{zz} \end{bmatrix}}_{\tilde{\mathbf{m}}(\boldsymbol{\zeta},\omega_f,\tau)} e^{j\omega_f\tau}, \tag{7.4}$$

where the argument $(\mathbf{x},\ \boldsymbol{\zeta},\ \omega_f)$ is omitted for the Green's functions to simplify the notation.

Looking at the formulation in (7.4), we see an important phenomenon in the frequency domain where both the source origin-time as well as the source time-function (represented at ω_f) are translated into two (complex) constant factors. For the sake of consistency with the time-domain approach presented in [121] and [122], we also keep $\tilde{s}_0(\omega_f)$ in $\tilde{\boldsymbol{\Psi}}(\mathbf{x},\ \boldsymbol{\zeta},\ \omega_f)$ and thus in our dictionary. The contribution of the origin-time, however, can easily be accommodated in the newly defined sub-vector of interest $\tilde{\mathbf{m}}(\boldsymbol{\zeta}, \omega_f, \tau)$. Next, we expand (7.4) for L geophones located at $\mathbf{x}_1, \cdots, \mathbf{x}_L$ as well as by considering K sources located at $\boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_K$ to have

$$\begin{aligned}\tilde{\mathbf{u}}(\omega_f) &= [\tilde{\mathbf{u}}_1(\omega_f)^T, \cdots \tilde{\mathbf{u}}_L(\omega_f)^T]^T \\ &= \sum_{k=1}^{K} \underbrace{[\tilde{\boldsymbol{\Psi}}_1(\boldsymbol{\zeta}_k, \omega_f)^T, \tilde{\boldsymbol{\Psi}}_2(\boldsymbol{\zeta}_k, \omega_f)^T, \cdots, \tilde{\boldsymbol{\Psi}}_L(\boldsymbol{\zeta}_k, \omega_f)^T]^T}_{\tilde{\boldsymbol{\Psi}}(\boldsymbol{\zeta}_k, \omega_f)} \times \tilde{\mathbf{m}}(\boldsymbol{\zeta}_k, \omega_f, \tau_k),\end{aligned} \tag{7.5}$$

where $(.)^T$ denotes the transposition operator on a vector or a matrix, $\tilde{\boldsymbol{\Psi}}_l(\boldsymbol{\zeta}_k, \omega_f) = \tilde{\boldsymbol{\Psi}}(\mathbf{x}_l, \boldsymbol{\zeta}_k, \omega_f)$ with $\mathbf{x}_l$ the hypocenter of the l-th geophone, $\tilde{\mathbf{u}}_l(\omega_f) = \tilde{\mathbf{u}}(\mathbf{x}_l, \omega_f)$ is a 3×1 frequency-domain displacement vector observed at the l-th geophone; accordingly $\tilde{\mathbf{u}}(\omega_f)$ is of size $3L \times 1$. Again, similar to the time-domain approach, the next step will be discretizing the space into N grid points as the candidate points for the hypocenter of the K sources which helps us to expand (7.5) by considering all the grid points and construct a linear set of equations as

$$\tilde{\mathbf{u}}(\omega_f) = \underbrace{[\tilde{\boldsymbol{\Psi}}_1(\omega_f), \tilde{\boldsymbol{\Psi}}_2(\omega_f), \cdots, \tilde{\boldsymbol{\Psi}}_N(\omega_f)]}_{\tilde{\boldsymbol{\Psi}}(\omega_f)}\ \tilde{\mathbf{m}}(\omega_f), \tag{7.6}$$

where $\tilde{\boldsymbol{\Psi}}_n(\omega_f) = \tilde{\boldsymbol{\Psi}}(\boldsymbol{\eta}_n, \omega_f)$ with $\boldsymbol{\eta}_n$ the location of the n-th grid point and

$$\tilde{\mathbf{m}}(\omega_f) = [\tilde{\mathbf{m}}_1(\omega_f)^T, \tilde{\mathbf{m}}_2(\omega_f)^T, \cdots, \tilde{\mathbf{m}}_N(\omega_f)^T]^T, \tag{7.7}$$

is of size $6N \times 1$ where $\tilde{\mathbf{m}}_n(\omega_f) = \mathbf{0}$ unless there is a source on $\boldsymbol{\eta}_n$. Our parameters of interest can then be obtained by solving (7.6) for $\tilde{\mathbf{m}}(\omega_f)$.

Notable Remarks:

i) The fact that we accommodate the source origin-time-related constants ($e^{j\omega_f \tau}$) in $\tilde{\mathbf{m}}(\omega_f)$ describes that the dictionary is normally constructed with a zero-

origin-time source and thus the effect of the origin-time will appear in $\tilde{\mathbf{m}}(\omega_f)$ in the form of a complex constant.

ii) By looking at (7.4) and (7.6) we observe that the dictionary can be constructed with a source with even unknown $s_0(t)$ (equivalently, $\tilde{s}_0(\omega)$); whatever the source-time function in the real-time measurements is, its proportional effect in the form of a constant $\tilde{s}_k(\omega_f)/\tilde{s}_0(\omega_f)$ will appear in $\tilde{\mathbf{m}}(\omega_f)$. This is the key point of the frequency-domain approach, which allows us to design our (blind to $s(t)$) approach.

iii) This framework provides the flexibility to handle different source origin-times for different sources as well as different source time-functions. This also has the advantage that making a huge super-dictionary as the one proposed in [121] is not necessary anymore and the converted frequency-domain data will be handled more efficiently, as will be explained in the following.

iv) The down side is that the source origin-time and the source time-function effects will appear as constant factors in $\tilde{\mathbf{m}}(\omega_f)$ which makes it hard to extract them.

A simple possibility to estimate our desired parameters is to confine ourselves to one specific frequency and solve (7.6) using a G-LASSO type of estimator similar to the case of the time-domain approach; however, this approach will be naive as we do not really exploit all the information (encoded in different frequencies) available. Therefore, the important question is how to incorporate all the frequencies (the ω_f's) to make a much better estimation?

Notably, different from classical G-LASSO and other similar estimators, here we have different dictionaries $\tilde{\boldsymbol{\Psi}}(\omega_f)$ for different frequencies, which means that our different measurement vectors $\tilde{\mathbf{u}}(\omega_f)$ characterize different vectors of interest $\tilde{\mathbf{m}}(\omega_f)$. A pictorial view of the estimation problem at hand is depicted in Fig. 7.1. The key observation that should be taken into account to handle this problem is that even though the $\tilde{\mathbf{m}}(\omega_f)$'s contain different values (due to $\tilde{s}(\omega_f)$ and $e^{j\omega_f\tau}$), they share the same sparsity support, i.e., they are zero or non-zero at similar indices (groups). These groups are shown in Fig. 7.1 using similar colors within the $\tilde{\mathbf{m}}(\omega_f)$'s and across the corresponding subsections of the $\tilde{\boldsymbol{\Psi}}(\omega_f)$'s. In order to deal with this situation, we propose the following estimator (basically an extension of the estimator proposed in [127] for wideband beamforming as well as the G-LASSO employed in [121] and [122]) and we call it multi-dictionary G-LASSO (MDG-LASSO) given

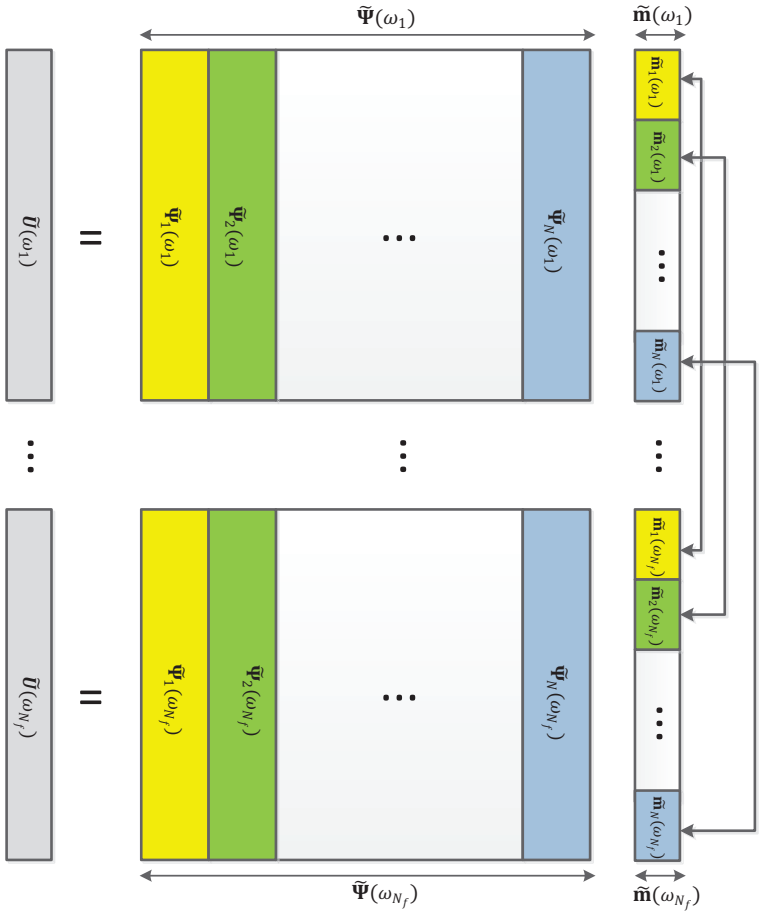


Figure 7.1: Illustration of linear sets of equations in different ω_f's. The $\tilde{\mathbf{m}}(\omega_f)$'s share a common sparsity support and also have the same group structure which helps to propose a proper estimation approach.

by

$$\hat{\mathbf{\Theta}}_{\text{MDG-LASSO}} = \arg\min_{\mathbf{\Theta}} \sum_{f=1}^{N_f} \|\tilde{\mathbf{u}}(\omega_f) - \tilde{\mathbf{\Psi}}(\omega_f)[\mathbf{\Theta}]_{:,f}\|_2^2 + \lambda \sum_{n=1}^{N} \|[\mathbf{\Theta}]_{6(n-1)+1:6n,:}\|_2, \tag{7.8}$$

Table 7.1: Velocity profile

Layers	**z margins** (m)	v_p (m/s)	v_s (m/s)	ρ
Layer 1	-2920	5326	3286	2200
Layer 2	-3125	4968	2985	2200
Layer 3	-9000	4487	2768	2200

where $\boldsymbol{\Theta} = [\tilde{\mathbf{m}}(\omega_1), \cdots, \tilde{\mathbf{m}}(\omega_{N_f})]$. The first term on the right hand side of (7.8) is the least squares part, which minimizes the error for the different frequencies and the second term enforces our specific group sparsity. It is worth pointing out that an analysis of the algorithms to solve (7.8) is outside the scope of this chapter and here we restrict ourselves to standard interior-point-based convex optimization tools such as CVX [107] to solve the problem. Based on the discussions presented in [106] for a related concept, incorporating all the frequencies within (7.8) will result in a gain in terms of identifiability compared to simply considering a single frequency, as is also corroborated by our simulation results in Section 7.4. To sum up, we would like to highlight that the proposed MDG-LASSO estimator takes into account three important features of the problem at hand, namely, the group-sparsity in the estimated vectors, the common sparsity support among them, and the fact that the model consists of different dictionaries for different measurements.

7.4 Evaluation Using Synthetic Data

In this section, we investigate the performance of the proposed algorithms in terms of positioning root mean squared error (PRMSE) and probability of detection (P_d) against signal to noise ratio (SNR), where the noise on the measured displacements is considered to be a band-limited additive white Gaussian noise occurring within the bandwidth of $\tilde{s}(\omega)$. We consider a three-layer elastic medium with different velocities in each layer. The velocity profile model can be found in Table 7.1 where primary-wave velocity, shear-wave velocity and density are respectively denoted by v_p, v_s and ρ. The synthetic data is generated using a MATLAB software package based on ray-tracing in order to compute the Green's functions for a full moment tensor source model in a multi-layer 3-D medium.

The acquisition setup is shown in Fig. 7.2. As can be seen, $L = 31$ geophones arranged in two arrays (vertical and horizontal) are employed to measure the displacement traces. This can also be done using a single array of geophones. Investigating the effect of different geophone geometries is omitted in this chapter for the sake of limited space. The area of interest is uniformly discretized into

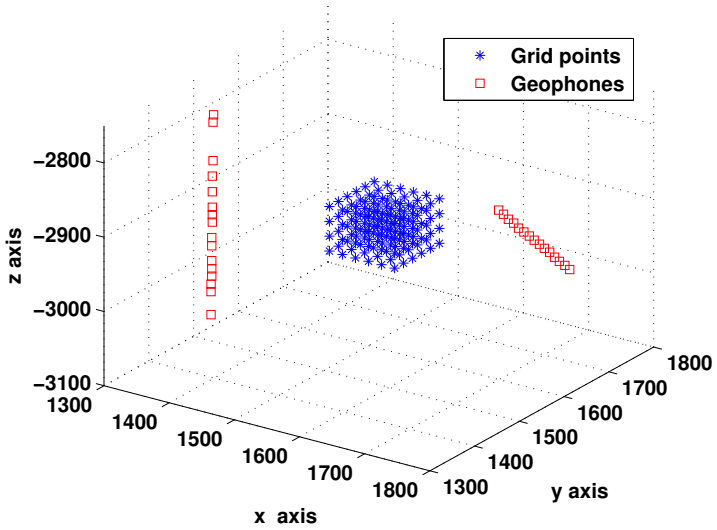


Figure 7.2: Acquisition setup

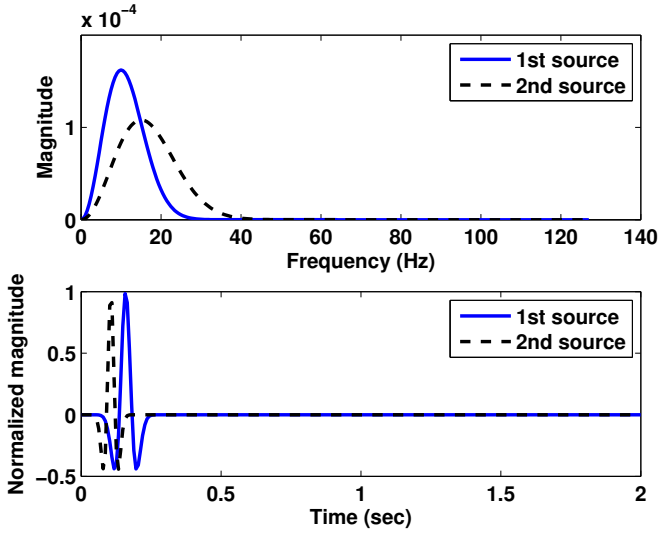


Figure 7.3: Two different $s(t)$'s

$N = 144$ grid points as is shown in Fig. 7.2 with a grid spacing of $\Delta = 20$m in three dimensions. The adopted moment tensor model is a 6-component vector (considering diagonal and upper-diagonal elements of $\mathbf{M}$) with fixed components ($\mathbf{m} = [0.7, 1, 0.5, 0.9, 0.7, 0.8]^T$) for all the sources. Note that this can be

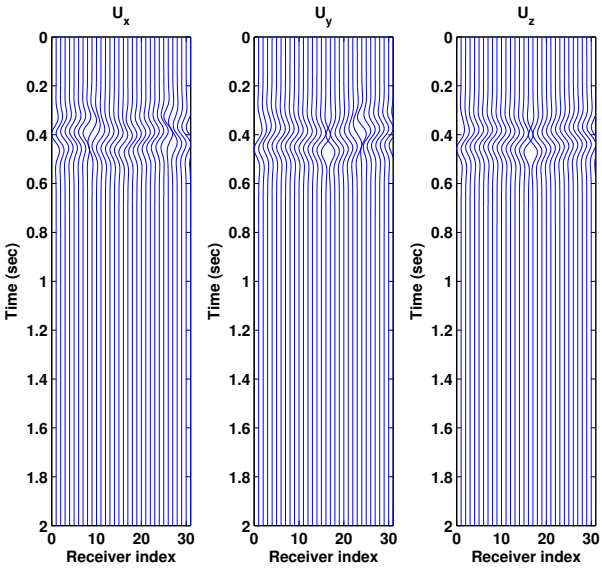


Figure 7.4: Received displacements

even different for all the sources and it will not affect our performance at all, as long as none of these components is significantly larger/smaller than the others. In order to prove that our approach is blind of $s(t)$, we consider two different $s(t)$'s as shown in Fig. 7.3, and we use the first one to construct the dictionaries ($s_0(t)$) and the second one for the real-time measurements (without loss of generality $s_k(t) = s(t) \neq s_0(t),\ \forall k$). As is depicted in Fig. 7.3, these functions are chosen to be the well-known Ricker wavelets with peak frequencies at 10Hz and 15Hz, respectively.

Another parameter which is clear from Fig. 7.3 is the measurement interval of 2s corresponding to $N_t = 256$ and thus $F_s = 256/2 = 128$Hz. This is obviously larger than twice the maximum frequency of the sources (approx. 40Hz according to Fig. 7.3) to satisfy the Nyquist criterion. Note that the source origin-times can be integer or even non-integer multiples of T_s and their values do not affect the performance. In our simulations, origin-times are chosen randomly within a range of $[0,\ 9]\,T_s$. Moreover, we consider only $N_f = 18$ frequencies ($2\pi[1, 3, 5, \cdots, 35]^T$) which means we will have 18 dictionaries $\tilde{\mathbf{\Psi}}(\omega_f)$ each of size $3L(=93) \times 6N(=864)$. Notably, another design consideration which has carefully been taken into account is that the length of the time-bin should be much larger than the rule of thumb maximum possible delay of the received displacement traces. This is to ensure that the latest displacement arrivals will be covered by measurements.

We consider up to $K = 3$ sources occurring simultaneously (during one measurement interval). Most of the simulations, whenever they do not illustrate a single snapshot, are averaged over $P = 50$ independent Monte Carlo runs where in each run the sources are deployed on different random locations (hypocenters). Increasing P will result in smoother curves. For the sake of comparison, besides the MDG-LASSO, we also consider a G-LASSO for which only an appropriate single frequency (here $f = 15$) is taken into account. Another possibility is to average the results of this G-LASSO over all the frequencies, which is not illustrated here. Averaging over different frequencies will not provide a much better result because in many single frequencies the estimations are poor, especially, for the case of multiple microseismic sources.

In order to quantify the performance we consider two different metrics:

- First, the positioning root mean squared error (PRMSE) defined by

$$\text{PRMSE} = \sqrt{\frac{1}{PK} \sum_{p=1}^{P} \sum_{k=1}^{K} e_{k,p}^2}, \tag{7.9}$$

  where $e_{k,p}$ represents the distance between the real hypocenter of the k-th source and its estimated hypocenter at the p-th Monte Carlo trial.

- Second, the probability of detection (P_d) where a source is considered to be detected if it is estimated to be within a sphere with radius $\sqrt{3} \times \Delta$ around its real hypocenter with Δ defined earlier.

Let us start with a single source located at $(1525, 1585, -2900)$ corresponding to our grid point with index 62. The displacements measured at the 31 geophones are plotted in Fig. 7.4. The SNR is set to 30dB. The result of our proposed parameter estimation algorithm is illustrated in Fig. 7.5 where as can be seen both G-LASSO and MDG-LASSO activate the correct group of indices in $\tilde{\mathbf{m}}$ (i.e., $61 \times 6 + 1 = 367, \cdots, 61 \times 6 + 6 = 372$). Note that for a better visualization, the amplitudes of the estimated moment tensors contained in $\hat{\mathbf{\Theta}}_{\text{MDG-LASSO}}$ are normalized, and we plot $\tilde{\mathbf{m}} = \sum_{f=1}^{N_f} \tilde{\mathbf{m}}(\omega_f)$ for MDG-LASSO. It is notable that, in contrast to G-LASSO, the moment tensors estimated by MDG-LASSO are just scaled versions of the real moment tensors. We would like to emphasize that according to literature [125] the normalized moment tensors contain important information about the nature of the sources, and thus, this information will be extracted using our proposed approach. Further, this is also a promising point as it motivates a further

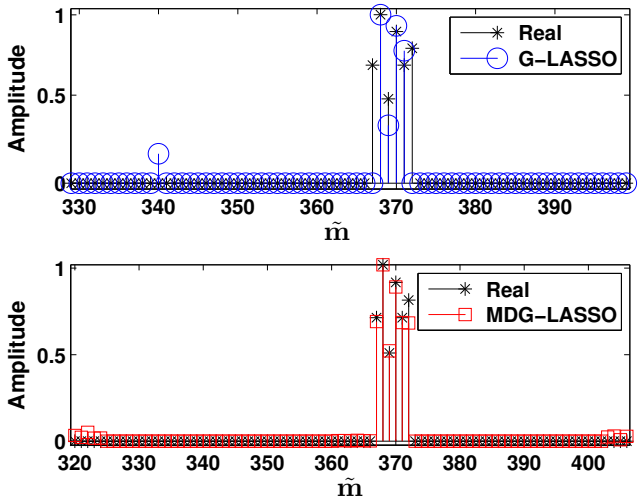


Figure 7.5: Estimation of $\tilde{\mathbf{m}}$ for single source; selected indices and corresponding amplitudes compared to their real values.

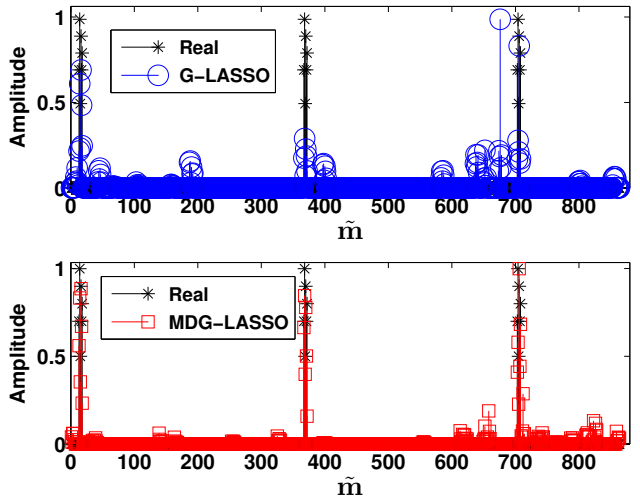


Figure 7.6: Estimation of $\tilde{\mathbf{m}}$ for multiple sources; selected indices and corresponding amplitudes compared to their real values.

post-processing step to possibly extract the exact moment tensor values as well as origin-times from the estimated amplitudes. This topic is left as future work.

Fig. 7.6 is similar to Fig. 7.5 but it presents the case of $K = 3$ sources with dif-

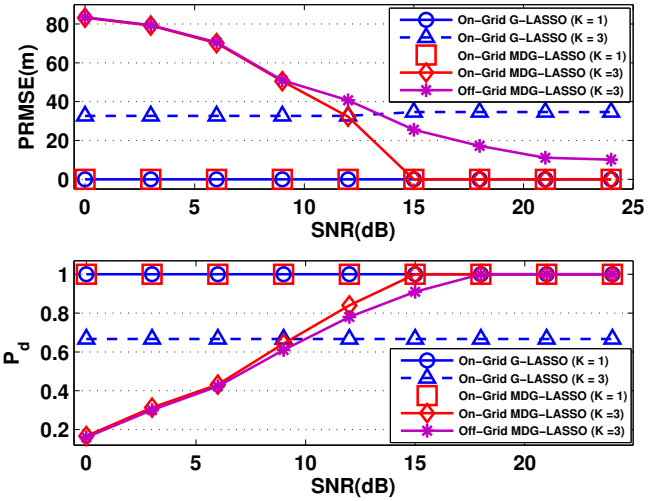


Figure 7.7: Proposed approach; localization and detection performances vs. SNR for $K = 1$ and $K = 3$ on-grid and off-grid sources.

ferent origin-times. The other two sources are located at $(1545, 1505, -2880)$ and $(1560, 1525, -2940)$ where the latter is off-grid (close to the grid point with index 118) and the former is on the grid point with index 3. As can be seen from the activated indices, while MDG-LASSO can easily handle the three sources, the (single-frequency) G-LASSO does not show an acceptable performance with a stable estimation. This is because only with 91 rows (measurements) in $\tilde{\boldsymbol{\Psi}}(\omega_{15})$ it is impossible to accurately reconstruct a sparse vector of interest with $3 \times 6 = 18$ non-zero elements corresponding to the three sources. This issue is basically related to the concept of sparse reconstruction and the interested reader is referred to [85]. The above result corroborates the idea that incorporating all the frequencies at the same time with our proposed MDG-LASSO estimator significantly improves the overall estimation performance. Note that for the third source (closest to the grid point with index 118), the effect of being off-grid appears as a few other side-groups of indices being activated with considerably smaller amplitudes compared to the correct group, i.e., the group corresponding to the grid point with index 118. This means that even in the case of off-grid sources, at least the closest grid points are usually distinguishable.

Finally, Fig. 7.7 depicts the performance of the proposed approach against SNR for $K = 1$ and $K = 3$ sources with $P = 50$ for both on-grid and off-gird sources. In the on-grid case, for a single source, both G-LASSO and MDG-LASSO (while

they are blind to $s(t)$) can attain an excellent performance in terms of both PRMSE and P_d within a reasonably large span of SNRs $[0,\ 24]$dB. However, the effect of the proposed modified framework to incorporate all the frequencies at the same time shows its effect when the number of sources is increased. Interestingly, for $K = 3$, the (single-frequency) G-LASSO cannot attain an acceptable PRMSE and P_d performance even for high SNRs. Quite the opposite, the MDG-LASSO attains a perfect detection performance and zero hypocenter estimation error for SNRs above 15dB. However, as can be seen, if we consider off-grid sources and only stick to finding the closest grid points, the performance of MDG-LASSO will be degraded in terms of both accuracy and detection performance.

7.5 Summary

In this chapter, we simply confine ourselves to finding the closest grid points to the off-grid sources whereas there might be a possibility to derive the relationship between the hypocenter of an off-grid source and its corresponding received displacements. In that case, techniques similar to the ones proposed in [103] can be employed to devise a two-step approach where in the first step the closest grid points to the off-grid sources are found and in the next step their grid mismatch is recovered to find the real hypocenters. The simpler the medium (single-layer homogenous in the best case), the easier such relationships can be discovered. Moreover, currently we only find the hypocenters of the sources and the normalized amplitudes of the moment tensors while according to our results there is a possibility to further post-process the results and estimate the exact moment tensor amplitudes as well as the corresponding origin-times.

Acknowledgment

The authors would like to thank Shell Global Solutions International B.V. for the permission to publish this chapter as well as Renat Shigapov for helpful discussions on the ray-tracing algorithm.

PART

IV

SPARSITY-AWARE SENSOR SELECTION

A mathematician is a device for turning coffee into theorems.

PAUL ERDÖS

8

SPARSITY-AWARE SENSOR SELECTION: CENTRALIZED AND DISTRIBUTED ALGORITHMS

Abstract

The selection of the minimum number of sensors within a network to satisfy a certain estimation performance metric is an interesting problem with a plethora of applications. We explore the sparsity embedded within the problem and propose a relaxed sparsity-aware sensor selection approach which is equivalent to the unrelaxed problem under certain conditions. We also present a reasonably low-complexity and elegant distributed version of the centralized problem with convergence guarantees such that each sensor can decide itself whether it should contribute to the estimation or not. Our simulation results corroborate our claims and illustrate a promising performance for the proposed centralized and distributed algorithms.

8.1 Introduction

We study the problem of selecting the minimum number of sensors among a network of sensor nodes in order to estimate a vector of interest so that a given mean squared error (MSE) is satisfied. This problem is of great interest in several practical application domains including robotics, target tracking, and energy efficient network management, to name a few (see for instance [35] and references therein). A straightforward method to solve such a problem is a combinatorial approach considering all possible combinations of all possible sizes of candidate sensors to satisfy the constraint, which is numerically intractable for a large number of sensors and thus motivates a more intelligent and structured approach. The problem becomes even more challenging when a distributed context is considered.

A related sensor selection problem has been studied in [35] where elegant convex relaxations are designed for primal and dual problems. However, instead of optimizing different performance metrics and fixing the number of sensors as in [35],

we minimize the number of sensors given a performance constraint, which is generally more practical. Interestingly, this enables us to exploit the sparsity embedded within the problem. From this angle, our look is closer to what is proposed in [38] for selecting reliable sensors, also called "robust sensing". However, we consider a different constraint than the one in [38], and we do not need the sensors to take measurements for solving the selection problem; we only need them to know their regression coefficients. Also, in both [35] and [38], a distributed approach has not been considered.

A decentralized implementation of [35] is proposed in [37]; however, the heuristic assumption of two "leader" nodes violates the classical definition of a distributed approach. Another relevant problem, but of a different nature, is considered in [39], where a distributed algorithm is designed to identify the sensors containing relevant information by a sparsity-aware decomposition of the measurement covariance matrix. Finally, in [128], two distributed implementations of [35] based on a truncated Newton algorithm are proposed. Compared to our work, first, [128] deals with a slightly different problem. Second, it considers a log-barrier and truncated Hessian approximate of the relaxed problem with no convergence (error) guarantees. Third, private sensor information has to be broadcast in this approach whereas we avoid that. Finally, our complexity is not a function of the number of sensors but of the number of sensed dimensions, and hence, it is considerably lower.

8.2 Problem Definition

We consider m sensor nodes distributed over an area of interest in $\mathbb{R}^d$, with $d \leqslant m$, which are supposed to estimate the unknown vector $\mathbf{x} \in \mathbb{R}^n$. The sensor nodes are equipped with (limited) computational and communication capabilities and each of them measures

$$y_i = \mathbf{a}_i^T \mathbf{x} + \eta_i, \; i = 1, \ldots, m, \tag{8.1}$$

where the $\mathbf{a}_i$'s $\in \mathbb{R}^n$ span $\mathbb{R}^n$ ($m \gg n$) and η_i is an additive zero-mean white measurement noise. Notably, considering the spatial distribution of the sensors, we assume that the $\mathbf{a}_i$'s are different so that we can distinguish the sensors based on their regressors. Here, we are interested in selecting *a priori* the minimum number of sensors (namely, measurements) so that the mean squared error (MSE) of estimating $\mathbf{x}$ is smaller than a desired value γ. Furthermore, we are interested in algorithms that would enable the sensors themselves to decide their own active/inactive status, without a centralized collection of the $\mathbf{a}_i$ vectors, i.e., we are interested in *distributed* algorithms.

8.3 Centralized Optimization Problem

In a centralized setup, all $\mathbf{a}_i$'s are available in a central unit which permits us to define the matrix $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_m]^T$. Now, we can construct $\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}$, where $\mathbf{y} = [y_1, \cdots, y_m]^T$, and $\boldsymbol{\eta} = [\eta_1, \cdots, \eta_m]^T$. For the linear measurement model (8.1), the MSE can be expressed as $\text{MSE} = \mathbb{E}\left[\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2\right] = \text{tr}(\mathbf{A}^T\,\mathbf{C}^{-1}\,\mathbf{A})^{-1}$, where tr(.) stands for the trace operator and $\mathbf{C}$ is the covariance matrix of the noise vector $\boldsymbol{\eta}$ [78]. Let the noise at the sensor nodes be uncorrelated, i.e., $\mathbb{E}[\eta_i \eta_j^T] = \sigma_i^2 \delta(i-j)$ with $\delta(.)$ denoting the Kronecker delta, and thus $\mathbf{C} = \text{diag}([\sigma_1^2, \cdots, \sigma_m^2])$. Based on this, the MSE can be reformulated as

$$\text{MSE} = \text{tr}\left(\Big(\sum_{i=1}^{m} \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i^T\Big)^{-1}\right), \tag{8.2}$$

where $\tilde{\mathbf{a}}_i = \mathbf{a}_i/\sigma_i = [\tilde{a}_{i,1}, \cdots, \tilde{a}_{i,n}]^T$. The associated selection constraint on the MSE can then be stated as

$$\text{tr}\left(\Big(\sum_{i=1}^{m} w_i\, \tilde{\mathbf{a}}_i\, \tilde{\mathbf{a}}_i^T\Big)^{-1}\right) \leqslant \gamma, \tag{8.3}$$

where the variable $w_i \in \{0, 1\}$ encodes whether the i-th sensor (measurement) is to be used. In practice, only a few sensors should be activated to satisfy the MSE constraint. Therefore, the problem can be cast as the following program

$$\underset{\mathbf{w}\in\{0,1\}^m, \mathbf{u}}{\text{minimize}} \quad \|\mathbf{w}\|_0 \tag{8.4a}$$

$$\text{s.t.} \quad \left[\begin{array}{c|c} \sum_{i=1}^{m} w_i\, \tilde{\mathbf{a}}_i\, \tilde{\mathbf{a}}_i^T & \mathbf{e}_j \\ \hline \mathbf{e}_j^T & u_j \end{array}\right] \succeq 0,\ j = 1, \ldots, n, \tag{8.4b}$$

$$||\mathbf{u}||_1 \leqslant \gamma,\ \ u_j \geqslant 0,\ j = 1, \ldots, n, \tag{8.4c}$$

where $\mathbf{w} = [w_1, \ldots, w_m]^T$ is the selection vector, $\mathbf{u} = [u_1, \ldots, u_n]^T$ is a vector of auxiliary variables, $\mathbf{e}_j$ is the j-th column of the $n \times n$ identity matrix $\mathbf{I}_n$, and the constraints (8.4b) and (8.4c) represented by Ω_γ are a more suitable representation of the original constraint (8.3), obtained using the Schur complement [129]. We denote the solution to (8.4) as $(\mathbf{w}^*, \mathbf{u}^*)$. Since $\|\mathbf{w}\|_0$ in the cost function of (8.4) and the finite-alphabet constraint on the w_i's are both non-convex, we consider the following relaxed version of the problem called sparsity-aware sensor selection (SparSenSe)

$$(\hat{\mathbf{w}}, \hat{\mathbf{u}}) := \underset{\mathbf{w}\in[0,1]^m, \mathbf{u}}{\text{argmin}}\ \{\|\mathbf{w}\|_1,\ \text{s.t.}\ (\mathbf{w}, \mathbf{u}) \in \Omega_\gamma\}. \tag{8.5}$$

8.4 Equivalence Theorem

In this section, we present an equivalence result, i.e., we prove that provided some conditions, the number of selected sensors is the same for both the original problem (8.4) and the relaxed version (8.5). To this aim, the following simplifying assumptions will be employed in this section.

Assumption 8.1: *Only one element of $\tilde{\mathbf{a}}_i$ is non-zero, i.e., there exists a single j for which $\tilde{a}_{i,j} \neq 0$.*

This assumption implies that the sensors can only sense one element (dimension) of the vector of interest. Let $\mathcal{I}_j,\ \forall j$ be the set containing the indices of the sensors which sense the j-th dimension, i.e., $\mathcal{I}_j = \{i \in \{1, 2, \cdots m\} \mid \tilde{a}_{i,j} \neq 0\}$ and $\mathcal{V}_j$ be the set containing the corresponding values, i.e., $\mathcal{V}_j = \{\tilde{a}_{i,j} \mid i \in \mathcal{I}_j\}$.

Assumption 8.2: *For each $\mathcal{V}_j$ there exists an $i \in \mathcal{I}_j$ such that $|\tilde{a}_{i,j}| > |\tilde{a}_{k,j}|\ \forall k \in \mathcal{I}_j, k \neq i$. We denote this $\tilde{a}_{i,j}$ as v_j^*.*

This assumption states that for each dimension j there exists a *unique dominant* sensor. Based on this, the following proposition and its proof are in place.

Proposition 8.1

Under Assumption 8.1 and Assumption 8.2, there exists a lower bound $\gamma^* = \max_j 1/|v_j^*| \sum_{j=1}^n 1/|v_j^*|$, such that if $\gamma \geqslant \gamma^*$, then $\|\hat{\mathbf{w}}\|_0 = \|\mathbf{w}^*\|_0 = n$. In addition, in this case, the solution of the relaxed version (8.5) is unique and corresponds to activating the sensors with v_j^* in the regressors.

Proof. The solution of the original non-convex problem has a cardinality of at least n, i.e., $\|\mathbf{w}^*\|_0 \geqslant n$. This is because we need to activate at least n sensors to attain a finite MSE in (8.3). Furthermore, in general, $\|\hat{\mathbf{w}}\|_0 \geqslant \|\mathbf{w}^*\|_0$. In the following, we will show that under Assumption 8.1 and Assumption 8.2, $\|\hat{\mathbf{w}}\|_0 = n$, and therefore our claim holds. The core idea is that under the aforementioned assumptions we can analytically compute $(\hat{\mathbf{w}}, \hat{\mathbf{u}})$ as explained in the following. The linear matrix inequality constraint (8.4b) can be written as

$$\mathbf{e}_j^T \left(\sum_{i=1}^m w_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i^T \right)^{-1} \mathbf{e}_j \leqslant u_j,\ j = 1, \ldots, n. \tag{8.6}$$

Under Assumption 8.1, we can write

$$\sum_{i=1}^m w_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i^T = \operatorname{diag}\left(\sum_{i \in \mathcal{I}_1} w_i a_{i,1}^2, \ldots, \sum_{i \in \mathcal{I}_n} w_i a_{i,n}^2 \right),$$

which is not singular (and therefore MSE is finite) when we select at least one sensor per dimension. This means that (8.6) yields

$$\sum_{i\in\mathcal{I}_j} w_i \tilde{a}_{i,j}^2 \geqslant u_j^{-1},\ j = 1,\ldots,n. \tag{8.7}$$

Considering (8.7) and the fact that we need to minimize $\|\mathbf{w}\|_1$, we have to maximize the u_j's w.r.t. the constraints $\|\mathbf{u}\|_1 \leqslant \gamma$ and $u_j \geqslant 0\ \forall j$, which leads to $\|\hat{\mathbf{u}}\|_1 = \gamma$. Given any $\mathbf{u}$, we can compute $w_i(\mathbf{u})$ analytically since the relaxed problem can now be written as the following linear program (LP)

$$\begin{aligned} \underset{\mathbf{w}\in[0,1]^m}{\text{minimize}} \quad & ||\mathbf{w}||_1 && \text{(8.8a)} \\ \text{s.t.} \quad & \sum_{i\in\mathcal{I}_j} w_i \tilde{a}_{i,j}^2 \geqslant u_j^{-1},\ j = 1,\ldots,n. && \text{(8.8b)} \end{aligned}$$

The solution of this LP lies on the vertices of the polytope defining the constraints (following Assumption 8.2) and for $u_j \geqslant 1/v_j^{*2}$ it is given by

$$w_i(\mathbf{u}) = \begin{cases} (u_j v_j^{*2})^{-1} & \text{if } \tilde{a}_{i,j} = v_j^*, \\ 0 & \text{otherwise.} \end{cases} \tag{8.9}$$

This helps us to rewrite (8.5) as

$$\hat{\mathbf{u}} = \underset{\mathbf{u},\ u_j \geqslant 1/v_j^{*2}}{\operatorname{argmin}} \left\{ \sum_{j=1}^{n} (u_j v_j^{*2})^{-1},\ \text{s.t.}\ \|\mathbf{u}\|_1 = \gamma \right\}, \tag{8.10}$$

which is convex for $u_j \geqslant 0$. The optimal $\hat{\mathbf{u}}$ has to satisfy the KKT conditions given by

$$\begin{aligned} \hat{u}_j^{-2} &= \lambda v_j^{*2},\ j = 1,\ldots,n, && \text{(8.11a)} \\ \|\hat{\mathbf{u}}\|_1 &= \gamma, && \text{(8.11b)} \end{aligned}$$

where λ is the Lagrange multiplier associated with $\|\mathbf{u}\|_1 = \gamma$. From (8.11a), $\lambda \geqslant 0$; solving for $\hat{u}_j$ and substituting it into (8.11b), after some simplifications, leads to

$$\hat{u}_j = \frac{\gamma}{|v_j^*| \sum_{j=1}^{n} 1/|v_j^*|},\ j = 1,\ldots,n, \tag{8.12}$$

which due to the convexity of (8.10) is the unique optimizer of (8.10). Substituting

(8.12) back into (8.9) yields

$$\hat{w}_i = \begin{cases} \frac{\sum_{j=1}^{n} 1/|v_j^*|}{\gamma |v_j^*|} & \text{if } \tilde{a}_{i,j} = v_j^*, \\ 0 & \text{otherwise}, \end{cases} \tag{8.13}$$

for $\hat{u}_j \geqslant 1/v_j^{*2}$, i.e., $\gamma \geqslant \max_j 1/|v_j^*| \sum_{j=1}^{n} 1/|v_j^*| = \gamma^*$. Thus, for $\gamma \geqslant \gamma^*$, $\hat{\mathbf{w}}$ is unique and has cardinality n. ■

8.5 Distributed algorithm

Triggered by the localized nature of many phenomena of interest in practical applications, in this section, we develop a distributed version of the centralized approach proposed earlier. Let us start with some notations. We call $\mathcal{N}_i$ the neighborhood set of the i-th sensor including i itself, with cardinality $|\mathcal{N}_i| = N_i$ (either given or to be estimated). We also define the following convex sets:

$$\mathcal{W}_i = \{w_i \mid 0 \leqslant w_i \leqslant 1\}, \tag{8.14}$$

$$\mathcal{U} = \{\mathbf{u} \mid u_j \geqslant 0, \sum_{j=1}^{n} u_j \leqslant \gamma\}, \tag{8.15}$$

and form the Lagrangian of the problem (8.5) given by

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^{m} w_i - \sum_{j=1}^{n} \text{tr}\left(\left[\begin{array}{c|c} \sum_{i=1}^{m} w_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i^T & \mathbf{e}_j \\ \hline \mathbf{e}_j^T & u_j \end{array} \right] \mathbf{G}_j \right) \\ &= \sum_{i=1}^{m} w_i - \sum_{j=1}^{n} \sum_{i=1}^{m} \text{tr}\left(\left[\begin{array}{c|c} w_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i^T & \mathbf{e}_j/m \\ \hline \mathbf{e}_j^T/m & u_j/m \end{array} \right] \mathbf{G}_j \right) \\ &= \sum_{i=1}^{m} \left(w_i - \sum_{j=1}^{n} \text{tr}\left(\left[\begin{array}{c|c} w_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i^T & \mathbf{e}_j/m \\ \hline \mathbf{e}_j^T/m & u_j/m \end{array} \right] \mathbf{G}_j \right) \right) = \sum_{i=1}^{m} \mathcal{L}_i(w_i, \mathbf{u}, \mathbf{G}), \end{aligned} \tag{8.16}$$

where $\mathbf{G}_j \succeq 0, \ \forall j$ are appropriately sized dual variables, and $\mathbf{G} = [\mathbf{G}_1, \ldots, \mathbf{G}_n]$. The dual function defined on $\mathcal{L}$ can be given by

$$\begin{aligned} q(\mathbf{G}) &= \min_{w_i \in \mathcal{W}_i, \mathbf{u} \in \mathcal{U}} \sum_{i=1}^{m} \mathcal{L}_i(w_i, \mathbf{u}, \mathbf{G}) \\ &= \sum_{i=1}^{m} \left(\min_{w_i \in \mathcal{W}_i, \mathbf{u} \in \mathcal{U}} \mathcal{L}_i(w_i, \mathbf{u}, \mathbf{G}) \right) = \sum_{i=1}^{m} q_i(\mathbf{G}). \end{aligned} \tag{8.17}$$

Notably, since both $\mathcal{W}_i$ and $\mathcal{U}$ are convex and compact sets, given a certain value of $\mathbf{G}$, the functions $q_i(\mathbf{G})$ and their subgradient w.r.t. $\mathbf{G}$, called $\mathbf{Q}$ and defined later on, can be computed locally (for example using SeDuMi to solve the resulting LPs) at each sensor [130].

Whenever γ is large enough so that we expect sparse solutions in terms of $\hat{\mathbf{w}}$, Slater condition holds for (8.4). In fact, in this case, we can always find a pair $(\hat{\mathbf{w}},\ \hat{\mathbf{u}})$ that satisfies (8.4b) and (8.4c) strictly. Therefore, the original ℓ_1-regularization (8.5) leads to the differentiable dual optimization problem

$$\underset{\mathbf{G}_1\succeq 0,\ldots,\mathbf{G}_n\succeq 0}{\text{maximize}} \sum_{i=1}^{m} q_i(\mathbf{G}), \tag{8.18}$$

with *zero* duality gap. This convex optimization program can be solved iteratively in a distributed fashion using a variety of algorithms. For instance, we can use gradient-based methods, such as the dual averaging scheme of [131] with a variable stepsize, or the simpler dual subgradient of [130] with a fixed stepsize. The latter method has the advantage of providing a recovery mechanism for the primal solution (i.e., we recover $\hat{\mathbf{w}}$ as a by-product of the optimal $\mathbf{G}$, which is in fact our goal). Furthermore, the subgradient method of [130] has the benefit to employ a fixed stepsize giving explicit trade-offs in terms of accuracy and feasibility of the solution and the number of iterations. In particular, given the number of iterations t, and the stepsize α, we can prove that (see [130, Proposition 1])

$$\operatorname{tr}\Big(\sum_{i=1}^{m} \hat{w}_i^t\, \tilde{\mathbf{a}}_i\, \tilde{\mathbf{a}}_i^T\Big)^{-1} \leqslant \gamma + \frac{c^2}{t\alpha}, \tag{8.19}$$

where $\hat{w}_i^t$ is the recovered approximate primal solution for sensor i at iteration t, and c^2 is a positive constant that depends on the problem at hand. This equation tells us *a priori* how many iterations we need to run before we reach a given infeasibility level; or provides us with a bound on how much we should tighten the constraint on γ to guarantee feasibility w.r.t. the MSE constraint for finite t.

In order to implement the dual subgradient of [130], each node requires a copy of $\mathbf{Q} = \sum_{i=1}^{m} \nabla_{\mathbf{G}^i} q_i(\mathbf{G}^i)$. This can be circumvented by using the method of [132] where the local sensor nodes have different local copies of $\mathbf{Q}$, say $\mathbf{Q}^i$, and they run an inexact consensus procedure for φ times (where $\varphi \in \mathbb{N}^+$). If $\varphi \to \infty$, we recover the procedure of [130], while if φ is limited we introduce an additional error in the distributed optimization procedure. Our proposed distributed sparsity-aware sensor selection (DiSparSenSe) algorithm can be summarized in Algorithm 8.1.

We would like to highlight that DiSparSenSe will converge to the solution of SparSenSe with an error floor dependent on α and φ. This can be proven using an ϵ-subgradient argument as discussed in [130] and [132].

8.6 Numerical Results

In this section, we investigate the performance of the proposed algorithms to see if SparSenSe actually selects a few sensors to satisfy the MSE constraint as well as to illustrate that DiSparSenSe selects the same sensors as SparSenSe. To this aim, we consider $m = 50$ sensors to estimate a parameter of interest $\mathbf{x}$ of dimension $n = 2$.

Algorithm 8.1 DiSparSenSe

Initialization: We call the i-th sensor version of $\mathbf{G}$ at iteration t, $\mathbf{G}^{i,t}$. Let an initial value for $\mathbf{G}^{i,t}$ be given at each sensor node (cold start $\mathbf{G}_j^{i,0} = \mathbf{I}$). Initialize the $\hat{w}_i^t$'s with $\hat{w}_i^0 = 0$.

Input: $\mathbf{G}^{i,t}$, $\hat{w}_i^t$, $\forall i,\ j$.

1- Dual optimization (LP): Compute, in parallel at each sensor i, the value of $q_i(\mathbf{G}^{i,t})$, its derivative $\mathbf{Q}^{i,t} = \nabla_{\mathbf{G}^{i,t}} q_i(\mathbf{G}^{i,t})$, and the related optimal primal variable $\bar{w}_i^t$. The dimension of $\mathbf{Q}^{i,t}$ is the same as that of $\mathbf{G}^{i,t}$. This step requires the solution of an LP problem whose computational complexity is $O(n^3)$.

2- Primal recovery: Following the primal recovery method of [130], compute

$$\hat{w}_i^{t+1} = t\,\hat{w}_i^t/(t+1) + \bar{w}_i^t/(t+1).$$

3- Consensus:

For $\tau = 1$ to φ do

- ◇ Send $\mathbf{Q}^{i,t}$ to the neighboring sensor nodes. The communication cost involved is of $O(N_i n^3)$;
- ◇ Perform, in parallel, one consensus step as

$$\mathbf{Q}^{i,t} \leftarrow \sum_{l \in \mathcal{N}_i} \mathbf{Q}^{l,t}/N_i.$$

End

4- Dual recovery: Update each sensor node's dual variable and store it in local variables $\mathbf{G}^i$ as

$$\mathbf{G}^{i,t+1} = \mathcal{P}_{\geq 0}\left[\mathbf{G}^{i,t} + m\alpha\mathbf{Q}^{i,t}\right],$$

where $\mathcal{P}_{\geq 0}[\cdot]$ is the projection operator onto the cone of positive semidefinite matrices. This step requires n singular value decompositions (SVDs), each of which has a computational complexity $O(n^3)$.

Output: $\mathbf{G}^{i,t+1}$, $\hat{w}_i^{t+1}$, $\forall i,\ j$.

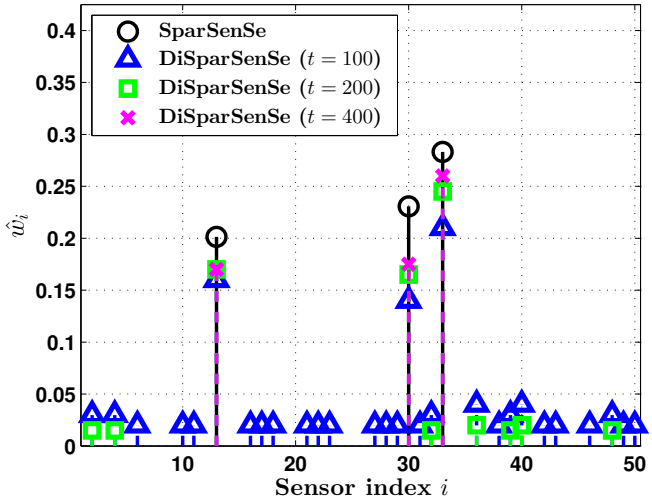


Figure 8.1: Centralized vs. distributed; selected sensors.

The measurement (regression) matrix $\mathbf{A} \in \mathbb{R}^{50\times 2}$ is drawn from a zero-mean unit-variance Gaussian distribution $\mathcal{N}(0,1)$. The noise experienced at different sensors has the same $\sigma = 1/\sqrt{\text{SNR}}$. For DiSparSenSe we assume that the sensors are connected based on a random connectivity graph $\mathcal{G}$ with average node degree of 9. Further, we set the number of consensus steps to $\varphi = 5$ or 8, the step-size to $\alpha = 0.01$ and the SNR to 10dB. Notably, for SparSenSe, we consider a sensor as active if $w_i > 0$, whereas for DiSparSenSe, due to the fixed step-size error floor, we consider a sensor as active if $w_i > \alpha$.

In the first simulation, depicted in Fig. 8.1, we plot $\hat{\mathbf{w}}$ estimated by SparSenSe and DiSparSenSe for $\gamma = 1$ and $\varphi = 5$. As can be seen, only 3 sensors (out of 50) are activated by SparSenSe to satisfy our MSE constraint which corroborates the fact that $\hat{\mathbf{w}}$ is sparse. Note that for $t = 100$ many different sensors are activated by DiSparSenSe. However, as expected, by increasing the number of iterations (from $t = 100$ to $t = 400$), the same sensors as for SparSenSe are activated by DiSparSenSe and the magnitude of the related $\hat{w}_i$'s gets closer to the values estimated by SparSenSe. This illustrates the fact that our distributed implementation (as expected) converges to the centralized algorithm.

In order to be able to quantitatively assess the performance, we also define $\mathcal{C}$ as the set of indices of the selected sensors by SparSenSe and $\mathcal{D}$ as the corresponding set for DiSparSenSe. This helps us to define an equivalence metric between the distributed and centralized algorithms as $\xi = 1 - |\mathcal{C} \cap \mathcal{D}|/\max\{|\mathcal{C}|, |\mathcal{D}|\}$ (i.e., if

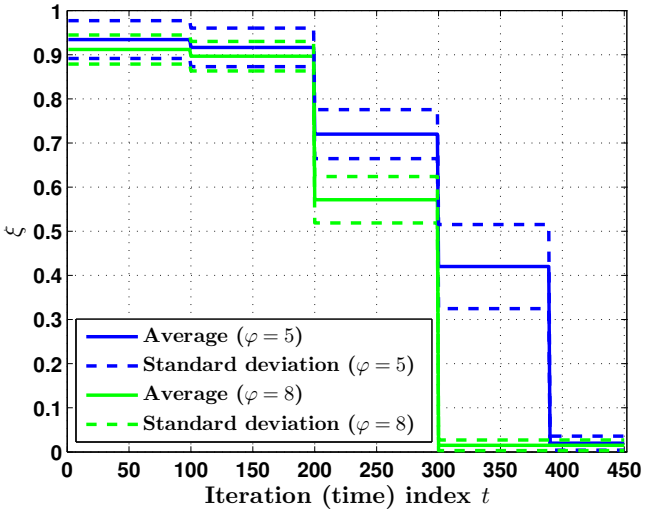


Figure 8.2: Equivalence metric ξ vs. t.

$\xi = 0$ then $\mathcal{D} \equiv \mathcal{C}$). Again, $\gamma = 1$, and we run 50 independent Monte Carlo trials. The result is shown in Fig. 8.2, where we clearly observe from the average of the Monte Carlo trials (the solid line) that with increasing t an equivalence is acquired as ξ goes to zero. Finally, the convergence is faster in the case of $\varphi = 8$ compared to $\varphi = 5$.

8.7 Discussion

We would like to conclude this letter by emphasizing the following points. First, note that based on (8.3) even after rounding the w_i's to 1 our MSE metric is certainly satisfied. Second, in our distributed algorithm, each sensor itself decides about its status of being active or inactive. More importantly, the "*private*" information contained in w_i is not broadcast, but instead an "*encoded*" version $\mathbf{Q}^{i,t}$ is communicated to reach convergence. Furthermore, based upon our earlier explanations, the total computational complexity of DiSparSenSe is $O(n^4)$ per node per iteration which is considerably lower compared to the computational complexity of SparSenSe $O(m^3)$ ($m \gg n$). The communication cost of DiSparSenSe is $O(\varphi N_i n^3)$ per node per iteration which is reasonably low as it is independent of m. Quite a few interesting topics such as a more elaborate equivalence proof and developing centralized and distributed algorithms for the case of correlated noise are left for future work.

9

DISTRIBUTED SPARSITY-AWARE SENSOR SELECTION

Abstract

The selection of the minimum number of sensors within a network to satisfy a certain estimation performance metric is an interesting problem with a plethora of applications. The problem becomes even more interesting in a distributed configuration when each sensor has to decide itself whether it should contribute to the estimation or not. In this chapter, we explore the sparsity embedded within the problem and propose a *sparsity-aware* sensor selection paradigm for both uncorrelated and correlated noise experienced at different sensors. We also present reasonably low-complexity and elegant *distributed* versions of the centralized problems with convergence guarantees. Moreover, we theoretically prove the convergence of our proposed distributed algorithms as well as analytically quantify their complexity compared to the centralized algorithms. Our simulation results corroborate our claims and illustrate a promising performance for the proposed centralized and distributed algorithms.

9.1 Introduction

We consider a typical sensor network estimation problem, where the sensors are supposed to estimate a vector of interest in a linear measurement model. For such a network, we study the problem of selecting the minimum number of sensors within the network, so that a given mean squared error (MSE) estimation performance is satisfied. This generic problem is of great interest in several practical application domains including radar and target tracking [133], event detection [36], and energy-efficient network management [134], to name a few. A straightforward solution to this problem is a combinatorial approach considering all possible combinations of all possible sizes of candidate sensors to satisfy the constraint, which is numerically intractable for a large number of sensors and thus motivates a more intelligent

and structured approach. The problem becomes even more challenging when a distributed context is considered, where each sensor should itself decide about its state of being selected (active) or not (inactive).

A related sensor selection problem has been studied in [35] where elegant convex relaxations are designed for primal and dual problems. Also, in [36] the same problem with a different optimality (selection) constraint is considered for event detection in sensor networks. However, instead of optimizing different performance metrics and fixing the number of sensors as in [36] and [35], we minimize the number of sensors given a performance constraint, which is generally more practical from a design perspective. Interestingly, this enables us to exploit the sparsity embedded within the problem and propose sparsity-aware solutions. From this angle, our approach is closer to what is proposed in [38] for selecting reliable sensors, also called "robust sensing". However, we consider a different constraint from the one in [38], and we do not need the sensors to be activated and take measurements for solving the selection problem; we only need them to know their regression coefficients. Worthy of being mentioned, is the work of [133], wherein both ideas (minimizing the number of sensors and minimizing the performance constraint) are considered for a multiple-radar localization architecture. The problem is formulated as a knapsack problem, but the sparsity is not taken into account. Also, in all the aforementioned studies, a distributed approach has not been considered.

The problem of distributed sensor selection is of crucial importance because in many practical network configurations, it is impossible to establish a central processing unit to gather all the information and make centralized decisions. Even if possible, this centralized process may drain significantly on the communication and energy resources [71], [135]. The alternative approach is to make decisions using in-network distributed processing [71]. A decentralized implementation of [35] is proposed in [37]; however, the heuristic assumption of two "leader" nodes violates the classical definition of a distributed approach. In [128], two distributed implementations of [35] based on a truncated Newton algorithm are proposed. In [136], we have explored the sparsity embedded within the problem and have proposed a relaxed sparsity-aware sensor selection approach. We have also presented a reasonably low-complexity distributed implementation of the centralized algorithm such that each sensor can decide itself whether it should contribute to the estimation or not. Compared to [136], the work of [128] deals with a slightly different problem and also requires the private sensor information to be broadcast whereas the proposed approach in [136] avoids that. Moreover, the distributed approach of [136] is considerably more efficient in terms of complexity compared to [128]. Finally, another relevant problem, but of a different nature, is considered in [39], where a

distributed algorithm is designed to identify the sensors containing relevant information by a sparsity-aware decomposition of the measurement covariance matrix.

In [136], we have only considered the case where the noise experienced by the sensors is *uncorrelated*. This might be a justifiable assumption in some cases, but in general, the noise experienced by the sensors can be correlated. Particularly, as it is pointed out in [137] and [138], since the measurement noises of different sensors may depend on a common "estimatee" (as is the case in our problem formulation), the sensors can observe correlated noise. Another example occurs when the estimatee is observed by sensors in a common noisy environment, such as noise generated by a jammer. In such cases, the measurement noises of the sensors are often correlated. This motivates us to extend our previously proposed algorithms to be able to operate in a more practical (and more general) framework of *correlated* noise.

A particular case where we can handle correlated noise is when we consider clusters of sensors with correlated noise and assume that the inter-cluster noise correlation is negligible, as we have proposed in [139]. This intuitive approach necessitates considering some sensors as "cluster heads" with higher processing power. In practice, such clusters (with zero inter-cluster correlations) can not always be defined. Furthermore, cluster heads impose extra constraints and violate the homogeneity of the sensor network. Thus, we would like to develop a generalized approach by dropping the cluster assumption. In this chapter, we extend our basic idea in [136] by presenting the following main contributions.

i) First, we modify the proposed distributed approach for uncorrelated noise in [136] by introducing a novel consensus weighting and conducting a double-consensus, which results in a smother convergence and robustness against the choice of regressors.

ii) Second, we formulate the centralized problem for the case of correlated noise, as well as propose an elegant low-complexity distributed implementation of the problem, where we have no clusters and cluster heads.

iii) Further, we analyze and quantify the convergence behavior of all of our proposed distributed algorithms and prove that we have convergence guarantees to the centralized algorithms.

iv) Finally, we investigate the computational and communication complexities involved in the proposed centralized and distributed algorithms, and promote that it is wise to employ the proposed distributed approaches.

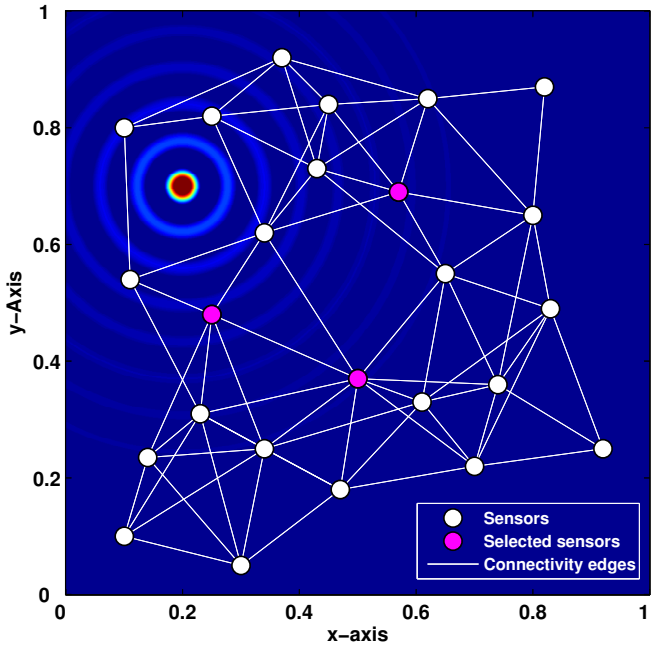


Figure 9.1: Schematic view of 2-D sensor selection

The rest of this chapter is organized as follows. In Section 9.2, we define the problem under consideration. Section 9.3 describes our proposed centralized and distributed algorithms for the case of uncorrelated noise. Section 9.4 is devoted to our proposed algorithms in order to handle correlated noise. In Section 9.5, the computational and communication costs involved in the proposed algorithms are investigated and compared. Numerical results are illustrated in Section 9.6, and this chapter is concluded in Section 9.7.

9.2 Problem Definition

We consider a network estimation problem where m sensor nodes distributed over an area of interest in $\mathbb{R}^d$ ($d \leqslant m$) are supposed to estimate an unknown vector $\mathbf{x} \in \mathbb{R}^n$. The elements of $\mathbf{x}$ can for instance represent the contribution of a static physical phenomenon in different dimensions within the area of interest. A schematic view of such a network deployed in order to estimate a static wave field in a 2-D area is shown in Fig. 9.1. The sensor nodes are equipped with (limited) computational and communication capabilities and each of them measures $y_i = \mathbf{a}_i^T \mathbf{x} + \eta_i,\ i = 1, \ldots, m$, where the regressors $\mathbf{a}_i$'s $\in \mathbb{R}^n$ are assumed known (or measured) and they should span $\mathbb{R}^n$ ($m \gg n$). The η_i's are the additive noise

experienced by different sensors, for which we need to know (or estimate) their second-order statistics. Note that, given the spatial distribution of the sensors, it practically makes sense that the $\mathbf{a}_i$'s are different so that we can distinguish the sensors based on their regressors. Here, we are interested in selecting *a priori* (without measuring y_i's) the minimum number of sensors so that the mean squared error (MSE) of estimating $\mathbf{x}$ is smaller than a desired value γ. Furthermore, we are interested in algorithms that would enable the sensors themselves to decide their own active/inactive status, without a centralized collection of the $\mathbf{a}_i$ vectors, i.e., we are interested in *distributed* algorithms. The next two sections of this chapter, which explain our proposed algorithms, are respectively derived based on the assumptions that the noise experience by the sensors is uncorrelated or correlated.

9.3 Sensor Selection for Uncorrelated Noise

In this section, we develop a sparsity-aware sensor selection paradigm, by considering uncorrelated noise. This is normally the case when the sensors are placed far apart. We derive centralized and distributed algorithms and investigate the convergence properties of the distributed algorithm.

9.3.1 Centralized Optimization Problem

In a centralized setup, all $\mathbf{a}_i$'s are transmitted to a central processing unit which allows us to define the matrix $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_m]^T$. Now, we can construct

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}, \tag{9.1}$$

where $\mathbf{y} = [y_1, \cdots, y_m]^T$, and $\boldsymbol{\eta} = [\eta_1, \cdots, \eta_m]^T$. We consider $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0},\ \mathbf{C})$, where the covariance matrix of the measurement noise $\mathbf{C}$ is by definition a symmetric and positive semidefinite (PSD) matrix [78]. For the centralized linear measurement model (9.1) and the maximum likelihood (ML) estimator, the MSE can be expressed as

$$\text{MSE} = \mathbb{E}\left[\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2\right] = \text{tr}\left((\mathbf{A}^T\,\mathbf{C}^{-1}\,\mathbf{A})^{-1}\right), \tag{9.2}$$

where $\hat{\mathbf{x}}$ is the ML estimate and tr(.) stands for the trace operator. Given uncorrelated noise, we have $\mathbb{E}[\eta_i\,\eta_j^T] = \sigma_i^2\delta(i-j)$ with $\delta(.)$ denoting the Kronecker delta, and thus $\mathbf{C} = \text{diag}([\sigma_1^2, \cdots, \sigma_m^2])$. Note that diag($\mathbf{x}$) returns a diagonal matrix with elements of $\mathbf{x}$ on its diagonal. Based on this assumption, the MSE can be reformulated as

$$\text{MSE} = \text{tr}\left((\sum_{i=1}^{m} \tilde{\mathbf{a}}_i\tilde{\mathbf{a}}_i^T)^{-1}\right), \tag{9.3}$$

where $\tilde{\mathbf{a}}_i = \mathbf{a}_i/\sigma_i = [\tilde{a}_{i,1}, \cdots, \tilde{a}_{i,n}]^T$. The associated selection constraint on the MSE can then be stated as

$$\mathrm{tr}\left((\sum_{i=1}^{m} w_i \, \tilde{\mathbf{a}}_i \, \tilde{\mathbf{a}}_i^T)^{-1} \right) \leqslant \gamma, \tag{9.4}$$

where the variable $w_i \in \{0, 1\}$ encodes whether the i-th sensor has to be activated. In practice, only a *few sensors* should be activated to satisfy the MSE constraint, which triggers the idea of exploiting the sparsity embedded within the problem. Therefore, we cast the problem as the following program

$$\underset{\mathbf{w}\in\{0,1\}^m, \mathbf{u}}{\text{minimize}} \quad \|\mathbf{w}\|_0 \tag{9.5a}$$

$$\text{s.t.} \quad \left[\begin{array}{c|c} \sum_{i=1}^{m} w_i \, \tilde{\mathbf{a}}_i \, \tilde{\mathbf{a}}_i^T & \mathbf{e}_j \\ \hline \mathbf{e}_j^T & u_j \end{array}\right] \geq 0, \ \forall j, \tag{9.5b}$$

$$\|\mathbf{u}\|_1 \leqslant \gamma, \ \ u_j \geqslant 0, \ j = 1, \ldots, n, \tag{9.5c}$$

where $\mathbf{w} = [w_1, \ldots, w_m]^T$ is the selection vector, $\mathbf{u} = [u_1, \ldots, u_n]^T$ is a vector of auxiliary variables, $\mathbf{e}_j$ is the j-th column of the $n \times n$ identity matrix $\mathbf{I}_n$, and the constraints (9.5b) and (9.5c) are a linear matrix inequality (LMI) representation of the original constraint (9.4), obtained by using the Schur complement [129]. We denote the global optimizers of (9.5) as $(\mathbf{w}^*, \mathbf{u}^*)$. Since both the cost $\|\mathbf{w}\|_0$ in (9.5) and the finite-alphabet constraint on the w_i's are non-convex, we consider the following relaxed version of the problem labeled as sparsity-aware sensor selection (SparSenSe)

$$\underset{\mathbf{w}\in[0,1]^m, \mathbf{u}}{\text{minimize}} \quad \|\mathbf{w}\|_1 \tag{9.6a}$$

$$\text{s.t.} \quad \left[\begin{array}{c|c} \sum_{i=1}^{m} w_i \, \tilde{\mathbf{a}}_i \, \tilde{\mathbf{a}}_i^T & \mathbf{e}_j \\ \hline \mathbf{e}_j^T & u_j \end{array}\right] \geq 0, \ \forall j, \tag{9.6b}$$

$$\|\mathbf{u}\|_1 \leqslant \gamma, \ \ u_j \geqslant 0, \ j = 1, \ldots, n, \tag{9.6c}$$

and we denote its optimizers as $(\hat{\mathbf{w}}, \hat{\mathbf{u}})$. In [136], we present a detailed equivalence result, i.e., we prove that provided some conditions, the number of selected sensors is the same for both the original (9.5) and the relaxed problem (9.6). Note that these conditions in [136] only help to obtain closed-form expressions in the equivalence proof, and for all of our simulations they do not necessarily hold.

9.3.2 Distributed Algorithm

In this subsection, we develop a distributed version of SparSenSe. Let us start with some notations. We call $\mathcal{N}_i$ the neighborhood set of the i-th sensor, with cardinality $|\mathcal{N}_i| = N_i$ (either given or to be estimated). Similarly, we define $\bar{\mathcal{N}}_i = \mathcal{N}_i \cup i$ with cardinality $|\bar{\mathcal{N}}_i| = \bar{N}_i = N_i + 1$. We also define the following convex sets to simplify our notations

$$\mathcal{W}_i = \{w_i \mid 0 \leqslant w_i \leqslant 1\}, \tag{9.7}$$

$$\mathcal{U} = \{\mathbf{u} \mid u_j \geqslant 0,\ \|\mathbf{u}\|_1 \leqslant \gamma\}, \tag{9.8}$$

and form the Lagrangian of (9.6a)-(9.6b) given by

$$\begin{aligned}
\mathcal{L}(\mathbf{w}, \mathbf{u}, \mathbf{G}) &= \sum_{i=1}^{m} w_i - \sum_{j=1}^{n} \mathrm{tr}\left(\left[\begin{array}{c|c} \sum_{i=1}^{m} w_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i^T & \mathbf{e}_j \\ \hline \mathbf{e}_j^T & u_j \end{array}\right] \mathbf{G}_j \right) \\
&= \sum_{i=1}^{m} w_i - \sum_{j=1}^{n} \sum_{i=1}^{m} \mathrm{tr}\left(\left[\begin{array}{c|c} w_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i^T & \mathbf{e}_j/m \\ \hline \mathbf{e}_j^T/m & u_j/m \end{array}\right] \mathbf{G}_j \right) \\
&= \sum_{i=1}^{m} \left(w_i - \sum_{j=1}^{n} \mathrm{tr}\left(\left[\begin{array}{c|c} w_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i^T & \mathbf{e}_j/m \\ \hline \mathbf{e}_j^T/m & u_j/m \end{array}\right] \mathbf{G}_j \right) \right) \\
&= \sum_{i=1}^{m} \mathcal{L}_i(w_i, \mathbf{u}, \mathbf{G}),
\end{aligned} \tag{9.9}$$

where $\mathbf{G}_j \succeq 0,\ \forall j$, are appropriately sized dual variables, and $\mathbf{G} = [\mathbf{G}_1, \ldots, \mathbf{G}_n]$. The dual function of $\mathcal{L}$ can be given by

$$\begin{aligned}
q(\mathbf{G}) &= \min_{w_i \in \mathcal{W}_i, \mathbf{u} \in \mathcal{U}} \sum_{i=1}^{m} \mathcal{L}_i(w_i, \mathbf{u}, \mathbf{G}) \\
&= \sum_{i=1}^{m} \left(\min_{w_i \in \mathcal{W}_i, \mathbf{u} \in \mathcal{U}} \mathcal{L}_i(w_i, \mathbf{u}, \mathbf{G}) \right) = \sum_{i=1}^{m} q_i(\mathbf{G}).
\end{aligned} \tag{9.10}$$

Note that in (9.9) and (9.10), we try to decompose the global problem into local problems, and to this aim, we reformulate the Lagrangian and corresponding dual function as the summation of local Lagrangians $\mathcal{L}_i(w_i, \mathbf{u}, \mathbf{G})$ and dual functions $q_i(\mathbf{G})$. Given a certain value of $\mathbf{G}$, the functions $q_i(\mathbf{G})$ and their subgradient w.r.t. $\mathbf{G}$, called $\mathbf{Q}$ and defined later on, can be computed locally (for example, using CVX [107] to solve the resulting linear programs (LPs)) at each sensor [130]. The mathematical steps to model the local dual optimization problems as equivalent LP

ones are omitted here for the sake of space limitation.

Whenever γ is large enough so that we expect sparse solutions in terms of $\hat{\mathbf{w}}$, Slater condition holds for (9.6), which can be formulated as the following proposition.

Proposition 9.1

Slater condition holds for (9.6), for sufficiently large γ.

Proof. For sufficiently large γ, we can always find a pair $(\mathbf{w},\ \mathbf{u})$ that strictly satisfies (9.6b) - (9.6c). ■

Therefore, the original ℓ_1-regularization (9.6) leads to the dual optimization problem

$$\underset{\mathbf{G}_1\succeq 0,\ldots,\mathbf{G}_n\succeq 0}{\text{maximize}} \sum_{i=1}^{m} q_i(\mathbf{G}), \tag{9.11}$$

with *zero* duality gap. This convex optimization program can be solved iteratively in a distributed fashion using a few possible algorithms. For instance, we can use proximal-based methods, such as the dual averaging scheme of [131] with a variable step-size, or the simpler dual subgradient of [130] with a fixed step-size. The latter method has the advantage of providing a recovery mechanism for the primal solution, i.e., we recover $\hat{\mathbf{w}}$ as a by-product of the optimal $\mathbf{G}$, which is in fact our goal. Furthermore, the subgradient method of [130] has the benefit of employing a fixed step-size which yields a simpler implementation. That is why we opt to employ the dual subgradient method of [130]. In order to implement the dual subgradient of [130], each sensor requires a copy of the subgradient of $q(\mathbf{G})$ w.r.t. $\mathbf{G}_j,\ \forall j$, i.e., each sensor requires a copy of $\mathbf{Q} = [\mathbf{Q}_1, \ldots, \mathbf{Q}_n]$. Given that $\mathcal{W}_i$ and $\mathcal{U}$ are compact and convex, we can define such a subgradient as

$$\mathbf{Q}_j = \nabla_{\mathbf{G}_j} q_i(\mathbf{G}_j) = -\left[\begin{array}{c|c} \sum_{i=1}^{m} \bar{w}_i \tilde{\mathbf{a}}_i \tilde{\mathbf{a}}_i^T & \mathbf{e}_j \\ \hline \mathbf{e}_j^T & \bar{u}_j \end{array}\right]. \tag{9.12}$$

where the $\bar{w}_i$'s and the $\bar{u}_j$'s are optimizers of

$$q(\mathbf{G}) = \min_{w_i\in\mathcal{W}_i, \mathbf{u}\in\mathcal{U}} \sum_{i=1}^{m} \mathcal{L}_i(w_i, \mathbf{u}, \mathbf{G}). \tag{9.13}$$

Note that the dimension of $\mathbf{Q}$ is the same as that of $\mathbf{G}$. The need for this *global* parameter can be circumvented by using the method of [132] where the sensors have different local copies of both $\mathbf{G}$ and $\mathbf{Q}$, say $\mathbf{G}^i$ and $\mathbf{Q}^i$, and they run an inexact consensus procedure for φ times ($\varphi \in \mathbb{N}_+$). In particular, to solve (9.11), we will consider the following inexact subgradient update. We call the i-th sensor

Algorithm 9.1 DiSparSenSe

Initialization: $\mathbf{G}_j^{i,0} = \mathbf{I}_{n+1}, \;\; w_i^0 = 0, \forall i,\, j.$

Input: $\mathbf{G}_j^{i,t}$, $\hat{w}_i^t$, $\forall i,\, j.$

1- Dual optimization (LP): Compute, in parallel at each sensor i, the value of $q_i(\mathbf{G}^{i,t})$, its derivative $\mathbf{Q}^{i,t}$ using (9.12), and the related optimal primal variables $\bar{w}_i^t$.

2- Primal recovery: Update method of [130]:

$$\hat{w}_i^{t+1} = t\,\hat{w}_i^t/(t+1) + \bar{w}_i^t/(t+1).$$

3- Consensus:

For $\tau = 1$ to φ

- $\diamond$ Send $\mathbf{G}^{i,t}$ and $\mathbf{Q}^{i,t}$ to the neighboring sensor nodes;
- $\diamond$ Perform, in parallel, one consensus step as

$$\mathbf{V}_j^{i,\tau,t} = \sum_{p=1}^{m} [\mathbf{Z}]_{ip} \mathbf{V}_j^{p,\tau-1,t},$$

which is initialized as in (9.14).

End

4- Dual recovery: Update each sensor's dual variable as

$$\mathbf{G}_j^{i,t+1} = \mathcal{P}_{\geq 0}\left[\mathbf{V}_j^{i,\varphi,t}\right], \forall j.$$

Output: $\mathbf{G}_j^{i,t+1}$, $\hat{w}_i^{t+1}$, $\forall i,\, j.$

version of $\mathbf{G}$ at iteration t, $\mathbf{G}^{i,t}$. We start with a given initial condition $\mathbf{G}_j^{i,0}$ for each sensor, and then $\forall\, t \geqslant 0$ we have

$$\mathbf{V}_j^{i,\tau=0,t} = \mathbf{G}_j^{i,t} + \alpha\, m\, \mathbf{Q}_j^{i,t}, \text{ for } j = 1, \cdots, n, \tag{9.14}$$

where α is the step-size. Next, we run φ times a consensus procedure as

$$\mathbf{V}_j^{i,\tau,t} = \sum_{p=1}^{m} [\mathbf{Z}]_{ip} \mathbf{V}_j^{p,\tau-1,t}, \tag{9.15}$$

and finally a projection over the cone of PSD matrices as

$$\mathbf{G}_j^{i,t+1} = \mathcal{P}_{\geq 0}\left[\mathbf{V}_j^{i,\varphi,t}\right], \text{ for } j = 1, \cdots, n. \tag{9.16}$$

In (9.15), $\mathbf{Z} \in \mathbb{R}^{m \times m}$ indicates a proper sensor-wise consensus matrix whose

weights have been defined using a Metropolis weighting, i.e.,

$$[\mathbf{Z}]_{ip} = \begin{cases} 1/(\max\{\bar{N}_i, \bar{N}_p\}) & \text{if } p \in \bar{\mathcal{N}}_i \\ 0 & \text{if } p \notin \bar{\mathcal{N}}_i, p \neq i \\ 1 - \sum_{l=1}^{m} [\mathbf{Z}]_{il} & \text{if } i = p. \end{cases} \tag{9.17}$$

If we execute (9.15) for $\varphi \to \infty$, we recover the procedure of [130], whereas if φ is limited we introduce an additional error in the distributed optimization procedure. Our proposed distributed SparSenSe (called DiSparSenSe) algorithm is summarized in Algorithm 9.1, where we denote the primal optimizer of DiSparSenSe at iteration t as $\hat{\mathbf{w}}^t$.

Remark 9.1

It is worth highlighting that in this chapter we have modified our previously proposed distributed algorithm in [136] in the consensus averaging step from two aspects. First, here we apply a double-consensus *on both* $\mathbf{G}$ *and* $\mathbf{Q}$ *instead of only a consensus on* $\mathbf{Q}$ *in [136]. Second, instead of a simple consensus averaging in [136], here we propose a symmetric consensus matrix* $\mathbf{Z}$*. We illustrate in Subsection 9.6.1 that these refinements lead to a smoother and faster convergence of DiSparSenSe.*

9.3.3 Convergence Properties of DiSparSenSe

We would like to highlight that DiSparSenSe will converge to the solution of SparSenSe with an error floor dependent on α and φ. This can be proven extending the ϵ-subgradient argument discussed in [130] and [132], as is briefly summarized in this subsection and is detailed in Appendix 9.A. We investigate both primal and dual convergence problems in Appendix 9.A. For the latter, we prove that there exists a finite $\bar{\varphi} > 0$ such that if $\varphi \geqslant \bar{\varphi}$ the sequence of dual functions $\{q(\mathbf{G}^{i,t})\}$ generated by DiSparSenSe converges to its optimal value within a bounded error floor. Based on this, we prove that the convergence of the running average primal sequence $\{\hat{\mathbf{w}}^t\}$ (as defined in step 2 of the algorithm) can be formulated in terms of a constraint violation, and an upper and lower bound on the primal function. The results in Appendix 9.A, (9.34)-(9.35), show that the running average primal function is upper bounded as

$$\|\hat{\mathbf{w}}^t\|_1 \leqslant \|\hat{\mathbf{w}}\|_1 + \frac{nG^2}{2t\alpha/m} + \frac{\alpha m \left(\sqrt{n}\, Q + \tau\right)^2}{2} + \tau m \sqrt{n}\, G + m\psi_2(\alpha, Q, \varphi).$$

and it is lower bounded as

$$\|\hat{\mathbf{w}}^t\|_1 \geqslant \|\hat{\mathbf{w}}\|_1 - \frac{9nG^2}{2t\alpha/m} - \frac{\alpha m\,(\sqrt{n}\,Q+\tau)^2}{2} - \tau m\sqrt{n}\,G - m\psi_2(\alpha, Q, \varphi),$$

where Q is an upper bound on the norm of the dual subgradient $\mathbf{Q}_j^{i,t}$, ψ_2 is a non-negative functions monotonically increasing with α and decreasing with φ, and τ is a non-negative scalar depending on ϖ. To sum up, these lower and upper bounds on the primal function indicate a convergence rate of $O(1/t)$ for the running average primal sequence to a bounded region around the optimal primal cost $\|\hat{\mathbf{w}}\|_1$ (the solution to SparSenSe). The width of this region depends on α and φ.

9.4 Sensor Selection for Correlated Noise

In this section, we develop a sparsity-aware sensor selection paradigm, by considering a correlated noise. This normally happens for neighboring sensors in a dense network. We derive centralized and distributed algorithms and investigate the convergence properties of the distributed algorithm.

9.4.1 Centralized Algorithm

Similar to the uncorrelated case in Subsection 9.3.1, we can construct (9.1) and compute the MSE of the ML estimator as in (9.2). However, given correlated noise, different from the case of uncorrelated noise, $\mathbf{C}$ is not diagonal and can even be a full matrix if all the sensors experience correlated noise. Thus, the non-diagonal elements $[\mathbf{C}]_{ij}$, $i \neq j$, should also be incorporated within our selection procedure. In order to handle these non-diagonal elements, we define a symmetric PSD selection matrix $\mathbf{W} = \mathbf{w}\mathbf{w}^T$, where $\mathbf{w}$ is our selection vector as defined earlier. Notice that based on this new definition of $\mathbf{W}$, $[\mathbf{C}]_{ij}$ will only be incorporated if both w_i and w_j are non-zero at the same time. The associated selection constraint on the MSE can then be stated as

$$\text{tr}\left((\mathbf{A}^T[\mathbf{W}\odot\mathbf{C}^{-1}]\mathbf{A})^{-1}\right) \leqslant \gamma, \tag{9.18}$$

where $\odot$ stands for the Hadamard product. Note that since $w_i \in \{0,1\}$, we have $\text{diag}(\mathbf{W}) = \mathbf{w}$, where $\text{diag}(\mathbf{X})$ returns a vector containing the diagonal elements of $\mathbf{X}$. Again, by exploring the sparsity embedded within the problem, it can be cast

as the following optimization program

$$\underset{\mathbf{W},\mathbf{u}}{\text{minimize}} \quad \|\text{diag}(\mathbf{W})\|_0 \tag{9.19a}$$

$$\text{s.t.} \quad \left[\begin{array}{c|c} \mathbf{A}^T[\mathbf{W}\odot\mathbf{C}^{-1}]\mathbf{A} & \mathbf{e}_j \\ \hline \mathbf{e}_j^T & u_j \end{array}\right] \succeq 0, \ \forall j, \tag{9.19b}$$

$$\|\mathbf{u}\|_1 \leqslant \gamma, \ u_j \geqslant 0, \ j=1,\ldots,n, \tag{9.19c}$$

$$[\mathbf{W}]_{i,j} \in \{0,1\}, \ \mathbf{W} \succeq 0, \ \text{rank}(\mathbf{W}) = 1. \tag{9.19d}$$

Similar to the derivations in Subsection 9.3.1, the constraints (9.19b) and (9.19c) are a more suitable representation of the original constraint (9.18). We denote the global optimizers of (9.19) as $(\mathbf{W}^*, \mathbf{u}^*)$. Clearly, the problem in (9.19) is non-convex due to its objective (ℓ_0 norm), and the first and the third terms in (9.19d) (finite-alphabet constraint on the elements of $\mathbf{W}$ and rank-1 constraint, respectively). Delving deeper in (9.19) reveals a problem on our way to distribute it in the next subsection, and that is the positive semi-definiteness constraint on $\mathbf{W}$ in (9.19d). Positive semi-definiteness is a global constraint and cannot be decomposed into corresponding sub-constraints, as we desire in the next subsection. That is why we use the following lemma and replace the first two terms in (9.19d) with the following sufficient condition

$$[\mathbf{W}]_{i,j} \in \{0,1\}, \ \mathbf{W} \in \mathbb{D}^m, \ \mathbf{W}^T = \mathbf{W}, \tag{9.20}$$

where $\mathbb{D} = \{\mathbf{X} \mid [\mathbf{X}]_{ii} \geqslant \sum_{j\neq i}[\mathbf{X}]_{i,j}, \forall i\}$ denotes the set of diagonally dominant matrices.

Lemma 9.1

A symmetric diagonally dominant matrix with real non-negative entries is PSD.

Proof. The proof follows from Greshgorin's circle theorem [52]. ■

Finally, we relax the three non-convex terms to obtain

$$\underset{\mathbf{W},\mathbf{u}}{\text{minimize}} \quad \text{tr}(\mathbf{W}) \tag{9.21a}$$

$$\text{s.t.} \quad \left[\begin{array}{c|c} \mathbf{A}^T[\mathbf{W}\odot\mathbf{C}^{-1}]\mathbf{A} & \mathbf{e}_j \\ \hline \mathbf{e}_j^T & u_j \end{array}\right] \succeq 0, \ \forall j, \tag{9.21b}$$

$$\|\mathbf{u}\|_1 \leqslant \gamma, \ u_j \geqslant 0, \ j=1,\ldots,n, \tag{9.21c}$$

$$0 \leqslant [\mathbf{W}]_{i,j} \leqslant 1, \ \mathbf{W} \in \mathbb{D}^m, \ \mathbf{W}^T = \mathbf{W}. \tag{9.21d}$$

We call this algorithm SparSenSe-C to distinguish it from SparSenSe, and we denote its optimizer as $(\hat{\mathbf{W}}, \hat{\mathbf{u}})$. A final step to recover $\hat{\mathbf{w}}$ from $\hat{\mathbf{W}}$ is to apply a Choleskey decomposition and a possible randomization to compensate for the relaxed rank-1 constraint. Alternatively, we can simply consider $\hat{\mathbf{w}} = \text{diag}(\hat{\mathbf{W}})$, which is what we do in this chapter.

9.4.2 Distributed Algorithm

In this subsection, we develop a distributed version of SparSenSe-C. Our approach towards the problem is to decompose it so that the i-th sensor can estimate the i-th row of $\mathbf{W}$. Looking at SparSenSe-C, we clearly observe that the non-diagonal elements of $\mathbf{C}^{-1}$ complicate the derivation of a distributed algorithm compared to the case of DiSparSenSe. The more non-diagonal elements, the more coupling terms are introduced, and thus, the more computational and communication steps are required.

Triggered by the localized nature of many phenomena of interest in practical applications, we define the following set of noise covariance matrices

$$\mathcal{C} = \{\mathbf{C} \mid \mathbf{C} \succeq 0, [\mathbf{C}]_{i,j} = 0, \text{ if } [\mathbf{I}_m + \mathcal{A}]_{i,j} = 0\}, \tag{9.22}$$

where $\mathcal{A}$ is the adjacency matrix associated with the network connectivity graph with zero diagonal elements. This means that we consider the nodes to experience correlated noise with their immediate neighbors. In practice, $\mathbf{C} \in \mathcal{C}$ is a sparse matrix if the network is not highly connected. We can reorder $\mathbf{C}$ using the Cuthill-McKee algorithm [140] to end up with a banded matrix. Note that we need to distribute $\mathbf{C}^{-1}$ as in (9.21). One solution is to compute the inverse of such a banded matrix in a distributed fashion using only local computations at different sensors via algorithms such as "DICI" in [141]. The alternative solution is to approximate it. Nonetheless, the inverse would not necessarily be a banded matrix. In general, $\mathbf{C} = \mathbf{C}_d + \bar{\mathbf{C}}_d$, where $\mathbf{C}_d$ and $\bar{\mathbf{C}}_d$ respectively stand for the matrices containing the diagonal and non-diagonal elements of $\mathbf{C}$. We can rewrite

$$\mathbf{C}^{-1} = \mathbf{C}_d^{-1} \left(\mathbf{I}_m + \mathbf{C}_d^{-1}\bar{\mathbf{C}}_d\right)^{-1}. \tag{9.23}$$

Now, for the specific case where the autocorrelation of the noise experienced at each sensor is much larger than the cross-correlation with its neighbors, we have $\|\mathbf{I}_m\|_F \gg \|\mathbf{C}_d^{-1}\bar{\mathbf{C}}_d\|_F$. For such a case, we can use Taylor's expansion as

$$\mathbf{C}^{-1} = \mathbf{C}_d^{-1} \left(\mathbf{I}_m - \mathbf{C}_d^{-1}\bar{\mathbf{C}}_d + 1/2(\mathbf{C}_d^{-1}\bar{\mathbf{C}}_d)^T\mathbf{C}_d^{-1}\bar{\mathbf{C}}_d + \cdots\right),$$

$$= \mathbf{C}_d^{-1}\left(\mathbf{I}_m - \mathbf{C}_d^{-1}\bar{\mathbf{C}}_d + 1/2\,\mathbf{C}_d^{-1}\bar{\mathbf{C}}_d^2\mathbf{C}_d^{-1} + \cdots\right),$$

which due to $\bar{\mathbf{C}}_d$, $\bar{\mathbf{C}}_d^2$, and the next powers of $\bar{\mathbf{C}}_d$ mandates single-hop, two-hop, and multi-hop communications. To simplify our next derivations, and without loss of generality (see Remark 2), we can confine ourselves to a first-order approximation as

$$\mathbf{C}^{-1} \approx \mathbf{C}_d^{-1}\left(\mathbf{I}_m - \mathbf{C}_d^{-1}\bar{\mathbf{C}}_d\right), \tag{9.24}$$

which after reordering is again a banded matrix and easier to be distributed. To simplify our subsequent notations, let us denote the (i,j)-th element of $\mathbf{C}$, $\mathbf{C}^{-1}$ and $\mathbf{W}$ by c_{ij}, c_{ij}^{-1} and w_{ij}, respectively. We also denote the i-th row of $\mathbf{W}$ by $\mathbf{w}_i$. Next, we define the following convex set

$$\mathcal{W}_i^c = \{\mathbf{w}_i \mid 0 \leqslant w_{ik} \leqslant 1, w_{ii} \geqslant \sum_{j\neq i} w_{ij},\ w_{ik} = w_{ki}, \forall k \in \bar{\mathcal{N}}_i,\}. \tag{9.25}$$

Note that sensor i only estimates $\bar{N}_i$ elements out of m in $\mathbf{w}_i$ because the rest are known to be zeros. The banded property of our newly defined $\mathbf{C}^{-1}$ in (9.24) helps us to expand (9.18) as

$$\text{tr}\left(\Big(\sum_{i=1}^{m}\sum_{k\in\bar{\mathcal{N}}_i} w_{ik}\, c_{ik}^{-1}\,\mathbf{a}_i\,\mathbf{a}_k^T\Big)^{-1}\right) \leqslant \gamma,$$

and construct the Lagrangian of (9.21a)-(9.21b) as

$$\begin{aligned}
\mathcal{L}(\mathbf{W},\mathbf{u},\mathbf{G}) &= \sum_{i=1}^{m} w_{ii} - \sum_{j=1}^{n}\text{tr}\left(\left[\begin{array}{c|c} \sum_{i=1}^{m}\sum_{k\in\bar{\mathcal{N}}_i} w_{ik}\, c_{ik}^{-1}\,\mathbf{a}_i\,\mathbf{a}_k^T & \mathbf{e}_j \\ \hline \mathbf{e}_j^T & u_j \end{array}\right]\mathbf{G}_j\right) \\
&= \sum_{i=1}^{m}\left(w_{ii} - \sum_{j=1}^{n}\text{tr}\left(\left[\begin{array}{c|c} \sum_{k\in\bar{\mathcal{N}}_i} w_{ik}\, c_{ik}^{-1}\,\mathbf{a}_i\,\mathbf{a}_k^T & \mathbf{e}_j/m \\ \hline \mathbf{e}_j^T/m & u_j/m \end{array}\right]\mathbf{G}_j\right)\right) \\
&= \sum_{i=1}^{m}\mathcal{L}_i(\mathbf{w}_i,\mathbf{u},\mathbf{G}).
\end{aligned} \tag{9.26}$$

Both $\mathbf{G}_j$ and $\mathbf{G}$ are defined earlier. Now, the dual function of $\mathcal{L}$ can be given by

$$\begin{aligned}
q(\mathbf{G}) &= \min_{\mathbf{w}_i\in\mathcal{W}_i^c,\mathbf{u}\in\mathcal{U}} \sum_{i=1}^{m}\mathcal{L}_i(\mathbf{w}_i,\mathbf{u},\mathbf{G}) \\
&= \sum_{i=1}^{m}\left(\min_{\mathbf{w}_i\in\mathcal{W}_i^c,\mathbf{u}\in\mathcal{U}}\mathcal{L}_i(\mathbf{w}_i,\mathbf{u},\mathbf{G})\right) = \sum_{i=1}^{m} q_i(\mathbf{G}).
\end{aligned} \tag{9.27}$$

Algorithm 9.2 ADMM

Input: ρ, $\mathbf{G}$, $\lambda_{i,k}$, $w_{i,k}$, $\forall i,\ k$.
For $s = 0$ to $s_{\max} - 1$
1- In parallel at each sensor, solve (9.28).
2- Each sensor transmits its own estimation $\hat{\mathbf{w}}_i^{s+1}$ to its neighbors.
3- Update $\lambda_{i,k}$'s as

$$\lambda_{i,k}^{s+1} = \lambda_{i,k}^{s} + \rho \frac{(w_{i,k}^{s+1} - w_{k,i}^{s+1})}{2}$$

End
Output: $\bar{\mathbf{w}}_i = \hat{\mathbf{w}}_i^{s_{\max}}$ and $\bar{\mathbf{u}} = \hat{\mathbf{u}}^{s_{\max}}$.

Given a certain value of $\mathbf{G}$, the functions $q_i(\mathbf{G})$ and their subgradient w.r.t. $\mathbf{G}$, called $\mathbf{Q}$ and defined later on, can be computed as follows. By taking a deeper look into (9.25), we detect another issue on our way towards a fully distributed implementation. The problem with $\mathcal{W}_i^c$ is that the row-wise symmetry constraint, $w_{ik} = w_{ki}, \forall k \in \bar{\mathcal{N}}_i$, cannot be handled only based on local information available at sensor i because we also need to know the w_{ki}'s. That is why we propose to modify $\mathcal{W}_i^c$ as

$$\overline{\mathcal{W}}_i^c = \{\mathbf{w}_i \mid 0 \leqslant w_{ik} \leqslant 1, w_{ii} \geqslant \sum_{j \neq i} w_{ij}, \forall k \in \bar{\mathcal{N}}_i\},$$

and instead of (9.27), optimize

$$\begin{aligned} \underset{\mathbf{w}_i \in \overline{\mathcal{W}}_i^c, \mathbf{u} \in \mathcal{U}}{\text{minimize}} \quad & \sum_{i=1}^{m} \mathcal{L}_i(\mathbf{w}_i, \mathbf{u}, \mathbf{G}) \\ \text{s.t.} \quad & w_{ik} = w_{ki}, \forall k \in \bar{\mathcal{N}}_i, \end{aligned}$$

which can be solved using the alternating direction method of multipliers (ADMM) [70] for a fixed $\mathbf{G}$. This is shown in Algorithm 9.2, where the symmetry is enforced by the second and third terms of

$$\begin{aligned} (\hat{\mathbf{w}}_i^{s+1}, \hat{\mathbf{u}}^{s+1}) = \underset{\mathbf{w}_i \in \overline{\mathcal{W}}_i^c, \mathbf{u} \in \mathcal{U}}{\operatorname{argmin}} \ & \mathcal{L}_i(\mathbf{w}_i, \mathbf{u}, \mathbf{G}) + \sum_{k \in \mathcal{N}_i} \lambda_{i,k}^s (w_{i,k} - \frac{w_{i,k}^s + w_{k,i}^s}{2}) \\ & + \sum_{k \in \mathcal{N}_i} \frac{\rho}{2} \left\| w_{i,k} - \frac{w_{i,k}^s + w_{k,i}^s}{2} \right\|_2^2, \end{aligned} \tag{9.28}$$

in step 1 of the algorithm.

Notice that (9.28) can be modeled as a disjoint quadratic program (QP) on $\mathbf{w}_i$ and

an LP on $\mathbf{u}$, and can be solved using the corresponding MATLAB functions (*linprog*(.), *quadprog*(.)), which is very efficient in terms of speed and computational complexity compared to solving it in its current form using CVX. The mathematical modeling details are omitted here for the sake of limited space. In practice, as we also discuss in the next subsection, we only need a few iterations to converge ($s_{\max} < 10$). Similar to our analysis in Subsection 9.3.2, the original ℓ_1-regularization (9.21) leads to the dual optimization problem

$$\underset{\mathbf{G}_1 \succeq 0, \ldots, \mathbf{G}_n \succeq 0}{\text{maximize}} \sum_{i=1}^{m} q_i(\mathbf{G}), \tag{9.30}$$

with *zero* duality gap. Again, we solve (9.30) using the dual subgradient method of [130], where each sensor requires a copy of the subgradient of $q(\mathbf{G})$ w.r.t. $\mathbf{G}_j$ as before

$$\mathbf{Q}_j = \nabla_{\mathbf{G}_j} q_i(\mathbf{G}_j) = -\left[\begin{array}{c|c} \sum_{i=1}^{m} \sum_{k \in \overline{\mathcal{N}}_i} \bar{w}_{ik} c_{ik}^{-1} \mathbf{a}_i \mathbf{a}_k^T & \mathbf{e}_j \\ \hline \mathbf{e}_j^T & \bar{u}_j \end{array}\right], \tag{9.31}$$

where $\bar{w}_{ik}$'s and $\bar{u}_j$'s are the outputs of the ADMM iterations in Algorithm 9.2. Similar to Subsection 9.3.2, the need for a global knowledge of $\mathbf{Q}$ is circumvented using an inexact consensus procedure. The rest of the steps follow the same trend as in DiSparSenSe except that instead of an LP to solve the dual optimization problem, here we have an extra inner-loop for ADMM. Our proposed algorithm for distributed SparSenSe in case of correlated noise (we call it DiSparSenSe-C) is summarized in Algorithm 9.3, where we denote the primal optimizer of DiSparSenSe-C at iteration t as $\hat{\mathbf{w}}^t$.

Remark 9.2

We would like to highlight that our assumption on the structure of $\mathbf{C}^{-1}$ does not limit the generality of the proposed solution, i.e., DiSparSenSe-C. In the most generic case where $\mathbf{C}^{-1}$ is a full matrix, each sensor has to estimate a full row instead of only a few elements (corresponding to its neighbors) in each row of $\mathbf{W}$. This calls for rounds of multi-hop communications if the network is connected. Nonetheless, our proposed approach immediately applies.

Remark 9.3

DiSparSenSe-C can readily be applied to the case of uncorrelated noise. However, if some knowledge about the nature of the experienced noise is available it makes sense to employ the corresponding algorithm, especially from a complexity perspective.

Algorithm 9.3 DiSparSenSe-C

Initialization: $\mathbf{G}_j^{i,0} = \mathbf{I}_{n+1}$, $\mathbf{w}_i^0 = \mathbf{0}$, $\forall i,\ j$.

Input: $\mathbf{G}_j^{i,t}$, $\hat{\mathbf{w}}^t$, $\forall i,\ j$.

1- Dual Optimization (ADMM):

- ◇ Initialize Algorithm 9.2 with ρ, $\lambda_{i,k} = 0$, $\forall i$ and $k \in \bar{\mathcal{N}}_i$, $\mathbf{G}^{i,t}$ and $\hat{\mathbf{w}}_i$'s.
- ◇ Use outputs $\bar{\mathbf{w}}_i^t = \bar{\mathbf{w}}_i$ and $\bar{\mathbf{u}}^t = \bar{\mathbf{u}}$ to compute $\mathbf{Q}^{i,t}$ using (9.31).

2- Primal recovery: Update method of [130]:

$$\hat{\mathbf{w}}_i^{t+1} = t\,\hat{\mathbf{w}}_i^t/(t+1) + \bar{\mathbf{w}}_i^t/(t+1).$$

3- Consensus:

For $\tau = 1$ to φ

- ◇ Send $\mathbf{G}^{i,t}$ and $\mathbf{Q}^{i,t}$ to the neighboring sensor nodes;
- ◇ Perform, in parallel, one consensus step as

$$\mathbf{V}_j^{i,\tau,t} = \sum_{p=1}^{m} [\mathbf{Z}]_{ip}\mathbf{V}_j^{p,\tau-1,t},$$

which is initialized as in (9.14).

End

4- Dual recovery: Update each sensor's dual variable as

$$\mathbf{G}_j^{i,t+1} = \mathcal{P}_{\succeq 0}\left[\mathbf{V}_j^{i,\varphi,t}\right], \forall j.$$

5- Selection: Estimate the selection vector

$$\hat{\mathbf{w}}^{t+1} = \text{diag}\left([(\hat{\mathbf{w}}_i^t)^T, \cdots, (\hat{\mathbf{w}}_i^t)^T]^T\right)$$

Output: $\mathbf{G}_j^{i,t+1}$, $\hat{\mathbf{w}}^{t+1}$, $\forall i,\ j$.

9.4.3 Convergence Properties of DiSparSenSe-C

In this subsection, we first investigate the convergence of the ADMM iterations. To this aim, we compare the solution of ADMM with the corresponding centralized problem at time iteration t for a fixed $\mathbf{G}^{i,t}$. We show that in practice, the proposed ADMM iterations converge to the result of the centralized problem with a modest accuracy, sufficient for our application, within only a few iterations. The related centralized problem is

$$\begin{aligned}
\underset{\mathbf{W},\mathbf{u}}{\text{minimize}} \quad & \text{tr}(\mathbf{W}) - \\
& \sum_{i=1}^{m}\sum_{j=1}^{n} \text{tr}\left(\left[\begin{array}{c|c} \sum_{k\in\overline{\mathcal{N}}_i} w_{ik}\, c_{ik}^{-1}\, \mathbf{a}_i\, \mathbf{a}_k^T & \frac{\mathbf{e}_j}{m} \\ \hline \frac{\mathbf{e}_j^T}{m} & \frac{u_j}{m} \end{array}\right] \mathbf{G}_j^{i,t} \right) \\
\text{s.t.} \quad & \|\mathbf{u}\|_1 \leqslant \gamma,\ u_j \geqslant 0,\ j = 1,\ldots,n, \\
& 0 \leqslant w_{i,j} \leqslant 1, \forall i,j,\ \mathbf{W} \in \mathbb{D}^m,\ \mathbf{W}^T = \mathbf{W},
\end{aligned}$$

where we denote the solution to the aforementioned problem with $(\hat{\mathbf{W}}^t_{\text{cent.}}, \hat{\mathbf{u}}^t_{\text{cent.}})$. This convergence is illustrated in Subsection 9.6.2 for our simulation setup. As a result of this fast dual optimization convergence, given that the major difference between DiSparSenSe and DiSparSenSe-C is the dual optimization part, the convergence proof of DiSparSenSe-C follows the same path as the one of DiSparSenSe. Therefore, we can prove similar expressions as (9.34)-(9.35), for DiSparSenSe-C. The formal proof is almost identical to the one of DiSparSenSe, and thus, we omit it in this chapter in favor of space limitation.

9.5 Complexity Analysis

Let us investigate the computational and communication complexities of the proposed distributed algorithms (DiSparSenSe and DiSparSenSe-C) compared to the centralized ones (SparSenSe and SparSenSe-C). A deeper look into the steps of Algorithm 9.1 reveals that step 2 requires the solution of an LP problem whose computational complexity is $O(n^3)$, where $O(.)$ denotes the order of complexity. Besides, the communication cost involved in step 3 is $O(\varphi N_i n^3)$ because n square matrices of size $n+1$ are broadcast to N_i neighbors for φ times. Furthermore, step 4 requires n singular value decompositions, each of which requires a computational complexity of $O(n^3)$. Thus, the total computational complexity of DiSparSenSe is $O(n^4)$ per sensor per iteration which is considerably lower compared to the computational complexity of SparSenSe which is $O(m^3)$ ($m \gg n$). The communication

Table 9.1: Complexity Order Comparison

Algorithm	Comp. complexity	Comm. complexity
SparSenSe	$O(m^3)$	–
SparSenSe-C	$O(m^3)$	–
DiSparSenSe	$O(n^4)$	$O(\varphi N_i n^3)$
DiSparSenSe-C	$O(n^4 + s_{\max}(N_i^3 + n^3))$	$O(N_i(\varphi\, n^3 + s_{\max}))$

cost of DiSparSenSe is $O(\varphi N_i n^3)$ per sensor per iteration which is reasonably low as it is independent of m.

DiSparSenSe-C involves almost the same computational and communication costs as compared to DiSparSenSe. The main difference is step 1 of Algorithm 9.3, i.e., ADMM, which requires the solution of $s_{\max}$ local QP problems with dimension N_i and local LP problems with dimension n, resulting in a total complexity of $O(s_{\max}(N_i^3 + n^3))$. ADMM also calls for an extra communication cost of $O(s_{\max} N_i)$ because of step 2 of Algorithm 9.2. Thus, the total computational cost of DiSparSenSe-C is $O(n^4 + s_{\max}(N_i^3 + n^3))$ and its communication cost is $O(N_i(\varphi\, n^3 + s_{\max}))$, both per sensor per iteration. Table 9.1 summarizes the discussed complexities of both centralized and distributed algorithms. From the table, we observe that the computational and communication complexities of DiSparSenSe-C are relatively larger than those of DiSparSenSe due to replacing a simple LP with ADMM iterations in order to handle the correlated noise.

9.6 Numerical Results

In this section, we investigate the performance of the proposed algorithms. First, we would like to see whether SparSenSe and SparSenSe-C actually select a few sensors (i.e., a sparse solution) which satisfies the MSE constraint. Then, we consider these centralized algorithms as our selection *performance metric* beyond which we cannot perform, and investigate whether their corresponding distributed algorithms (namely, DiSparSenSe and DiSparSenSe-C) select the same sensors or not.

To this objective, we consider a medium-scale network with $m = 50$ sensors to estimate a parameter of interest $\mathbf{x}$ of dimension $n = 2$. The regression matrix $\mathbf{A} \in \mathbb{R}^{50\times 2}$ is drawn from a zero-mean unit-variance Gaussian distribution $\mathcal{N}(0, 1)$. For DiSparSenSe and DiSparSenSe-C we assume that the sensors are connected based on a random connectivity graph $\mathcal{G}$ with average node degree of 5. Further, we set the SNR to 10dB and $\gamma = 0.1$. We could consider a sensor as active by defining

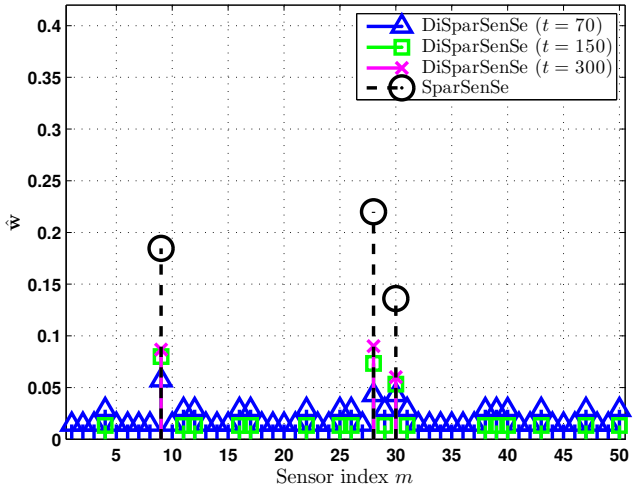


Figure 9.2: Selected sensors for the uncorrelated case

thresholds based on our estimation error floor (a complicated function of α and φ) coming from our convergence analysis in Appendix 9.A. A simpler alternative, a rule of thumb, would be to consider a sensor as active if $w_i > \alpha/10$, and this is what we consider in our simulations.

In order to quantitatively assess the performance of the distributed algorithms, we define an equivalence metric to investigate the normalized level of similarity between the selected sensor sets by the centralized and distributed algorithms. To this aim, we define $\mathcal{S}_c$ as the set of indices of the selected sensors by the centralized algorithms and $\mathcal{S}_d$ as the set for the corresponding distributed algorithms. This helps us to define an equivalence metric between the distributed and centralized algorithms as

$$\xi = 1 - |\mathcal{S}_c \cap \mathcal{S}_d| / \max\{|\mathcal{S}_c|, |\mathcal{S}_d|\},$$

which means that if $\mathcal{S}_c \equiv \mathcal{S}_d$, then $\xi = 0$.

9.6.1 Case of Uncorrelated Noise

In case of uncorrelated noise, for the sake of simplicity of our simulations, we assume that the noise experienced at different sensors has the same $\sigma = 1/\sqrt{\text{SNR}}$.

In the first simulation, depicted in Fig. 9.2, we plot $\hat{\mathbf{w}}$ estimated by SparSenSe and $\hat{\mathbf{w}}^t$ estimated by DiSparSenSe for $\varphi = 5$. As can be seen, only 3 sensors (out of 50) are activated by SparSenSe to satisfy our MSE constraint, which verifies the

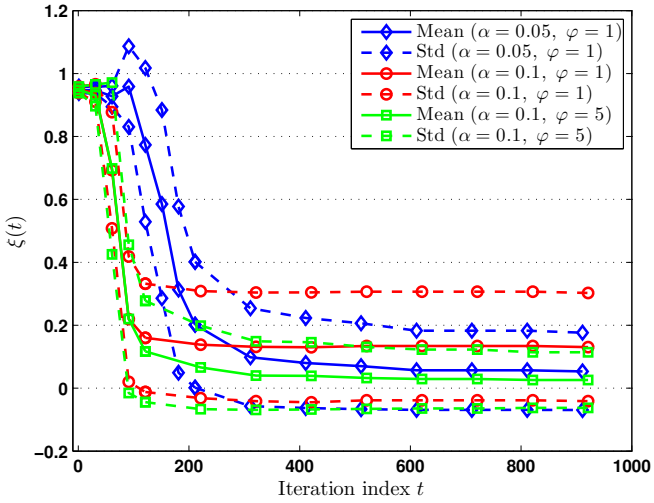


Figure 9.3: Equivalence metric $\xi(t)$

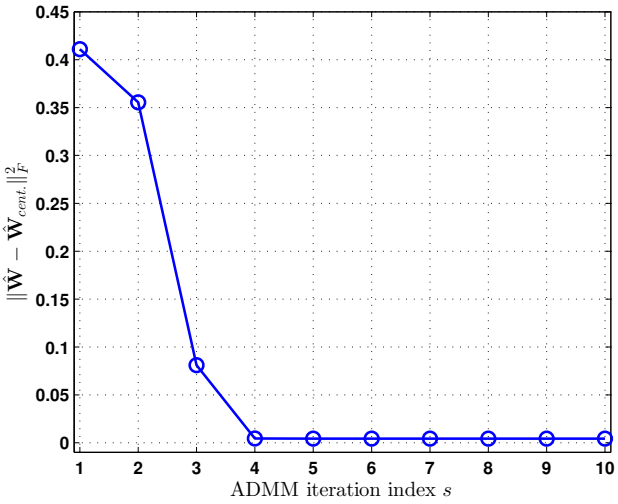


Figure 9.4: Convergence of ADMM iterations

fact that $\hat{\mathbf{w}}$ is sparse. Note that for $t = 70$ many different sensors are activated by DiSparSenSe. However, as expected, by increasing the number of iterations (from $t = 70$ to $t = 300$), the same sensors as for SparSenSe are activated by DiSparSenSe and the magnitude of the related $\hat{w}_i^t$'s gradually gets closer to the values estimated by SparSenSe. However, as is clear from the figure, it is not necessary to

attain the magnitudes estimated by SparSenSe to be able to make a decision about the selected sensors. This result illustrates the fact that our distributed implementation (as expected from our convergence analysis) converges to the solution of the centralized algorithm.

The next simulation result which is illustrated in Fig. 9.3, investigates the convergence of DiSparSenSe over 100 independent Monte Carlo realizations of $\mathbf{A}$ (leading to 100 different subsets of sensors to be selected) for $\varphi = 5$, and $\alpha = 0.1$ and 0.05. We also plot the standard deviation (std) of our estimates with dashed lines. As can be seen, for both values of α we converge to the correct solution with an error floor. The convergence is faster for $\alpha = 0.1$ as is expected from our convergence analysis, (9.34)-(9.35), because the second terms on the right-hand-side of both expressions (the ones $\propto 1/t$) vanish faster with a larger α. Fig. 9.3 also illustrates the effect of varying φ for $\alpha = 0.1$, where reducing φ from 5 down to 1 leads to a larger error floor. This can also be justified using our explanations in Appendix 9.A on Theorem 9.2.

Notice that Fig. 9.3 depicts a smoother convergence compared to our initial results in [136]. As we discussed earlier in Subsection 9.3.2 (Remark 1), this is due to our modified consensus weighting and the double-consensus. We also observe in our simulations that these modifications bring about a more robust performance against the choice of $\mathbf{A}$.

9.6.2 Case of Correlated Noise

In case of correlated noise, similar to the previous subsection, we assume that the noise experienced at different sensors has the same $\sigma = 1/\sqrt{\text{SNR}}$ and on top of that 5% correlation with the neighbors per sensor (we set $\|\mathbf{I}_m\|_F/\|\mathbf{C}_d^{-1}\bar{\mathbf{C}}_d\|_F = 0.05$). For the ADMM algorithm, we set ρ to 0.1 and initialize the $\lambda_{i,k}$'s with zeros.

Let us start by investigating the convergence of ADMM, based on our explanations in Subsection 9.4.3. The result is illustrated in Fig. 9.4 where we plot the primal convergence norm $\|\hat{\mathbf{W}}^t - \hat{\mathbf{W}}^t_{\text{cent.}}\|_F^2$ vs. s (the number of ADMM iterations) averaged over 50 iterations t. Clearly, $\|\cdot\|_F$ stands for the Frobenius norm. As can be seen from the figure, in practice, ADMM converges relatively fast within only a few ($s_{\max} < 10$) iterations. Note that this is partly due to the fact that the solution of DiSparSenSe-C is actually sparse, and hence, for many i and $k \in \bar{\mathcal{N}}_i$, $w_{ik} = w_{ki} = 0$. This means $\hat{\mathbf{W}}^t$ is almost automatically symmetric and only a few ADMM iterations would suffice to converge to a feasible solution.

In the next simulation, we plot $\hat{\mathbf{w}}$ and $\hat{\mathbf{w}}^t$ respectively estimated by SparSenSe-C and DiSparSenSe-C for $\varphi = 10$ in Fig. 9.5. As can be seen, only 3 sensors

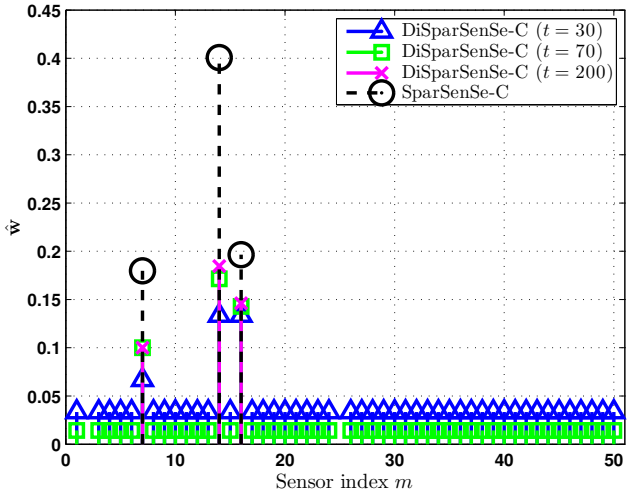


Figure 9.5: Selected sensors for the correlated case

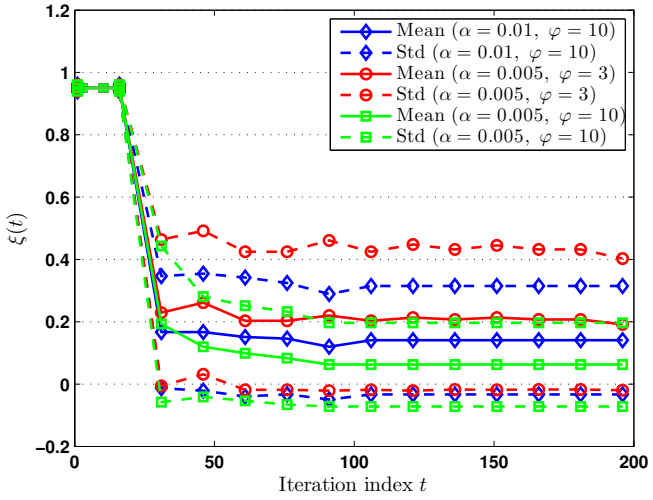


Figure 9.6: Equivalence metric $\xi(t)$

(out of 50) are activated by SparSenSe-C to satisfy our MSE constraint. Similar to the case of DiSparSenSe, by increasing the number of iterations from $t = 30$ to $t = 200$, the same sensors as for SparSenSe-C are activated by DiSparSenSe-C and the magnitude of the related $\hat{w}_i^t$'s gradually gets closer to the values estimated by SparSenSe-C. This result clarifies the fact that our distributed implementation (as

expected from our convergence analysis) converges to the solution of the centralized algorithm.

Finally, the simulation results depicted in Fig. 9.6 investigate the convergence of DiSparSenSe-C over 100 independent Monte Carlo trials for $\varphi = 10$, and $\alpha = 0.01$ and 0.005. As can be seen from the figure, for both values of α we converge with an error floor. Similar to the case of DiSparSenSe in the previous subsection, the convergence is faster for the larger $\alpha = 0.01$, as is also expected from our convergence analysis. However, we observe here that with $\alpha = 0.005$ we also get a better equivalence performance compared to $\alpha = 0.01$. Fig. 9.6 also illustrates the effect of varying φ for $\alpha = 0.005$ where reducing φ to 3 from 10 leads to a larger error floor. This can be justified using our convergence results, similar to our explanations for DiSparSenSe in the previous subsection.

9.7 Conclusions

We have proposed a framework for sparsity-aware sensor selection in centralized and distributed fashions for cases where the noise experienced by different sensors is either uncorrelated or correlated. In favor of the limited space, we have omitted the possibility of imposing different budget constraints (such as power budget) on the sensors. Our initial results show that involving such constraints into our optimization problems would lead to the selection of different subsets of sensors. Another direction to be investigated is the case of time-varying regressors. We are currently considering dynamic sparse reconstruction algorithms to handle this problem.

Appendix

9.A Convergence Analysis of DiSparSenSe

In this appendix, we analyze both primal and dual convergence properties of DiSparSenSe. First of all, since the sets $\mathcal{W}_i$ and $\mathcal{U}$ in (9.7) and (9.8) are compact, the subgradient $\mathbf{Q}_j^{i,t}$ is bounded by a certain finite bound Q [132] as

$$\|\mathbf{Q}_j^{i,t}\|_F \leqslant Q,\ j = 1, \cdots, n,\ i = 1, \cdots, m,\ t \geqslant 0.$$

For the consensus matrix $\mathbf{Z}$ in (9.17), it is true that

$$\mathbf{Z} = \mathbf{Z}^T,\ \mathbf{Z}\mathbf{1}_m = \mathbf{1}_m,\ \rho\Big(\mathbf{Z} - \frac{\mathbf{1}_m\mathbf{1}_m^T}{m}\Big) \leqslant \nu < 1,$$

where $\rho(\cdot)$ returns the spectral radius and ν is an upper bound on the value of the spectral radius. In the following, we assume that the dual variable estimates $\mathbf{G}_j^{i,t}$'s are bounded by a convex compact set (comprising the zero element) as

$$\|\mathbf{G}_j^{i,t}\|_F \leqslant G,\ j = 1, \cdots, n,\ i = 1, \cdots, m,\ t \geqslant 0,$$

for a certain finite positive constant G. Nonetheless, if this is not the case, we can always project them into such a bounded set, which will not considerably affect our subsequent convergence analysis [130].

9.A.1 Dual Objective Convergence

Let us start our convergence analysis in the dual sense by the following theorem.

Theorem 9.1

Let $\hat{q}$ be the optimal dual value of SparSenSe (9.10), i.e.,

$$\hat{q} = \max_{\mathbf{G}_1 \geq 0, \ldots, \mathbf{G}_n \geq 0} \sum_{l=1}^{m} q_l(\mathbf{G}).$$

Then, there exists a finite $\bar{\varphi} > 0$ such that if $\varphi \geqslant \bar{\varphi}$ the sequence of dual functions $\{q(\mathbf{G}^{i,t})\}$ generated by DiSparSenSe converges as

$$\limsup_{t \to \infty} q(\mathbf{G}^{i,t}) \geqslant \hat{q} - m\psi_1(\alpha, Q, \varphi),\ \ i = 1, \cdots, m,$$

where ψ_1 is a non-negative function of φ, α, and Q.

Proof. The proof is based on Theorem 2 of [132]. The trick is to first rewrite the steps of DiSparSenSe in a more compact way. To this aim, we define the vectors $\mathbf{g}_j^{i,t} = \text{vec}(\mathbf{G}_j^{i,t})$ and $\mathbf{h}_j^{i,t} = \text{vec}(\mathbf{Q}_j^{i,t})$. Next, we define the convex set $\mathcal{G}$ as

$$\mathcal{G} = \{\mathbf{g}^{i,t} | \mathbf{G}_j^{i,t} \succeq 0,\ j = 1, \cdots, n\}.$$

This helps us to rewrite the updates in DiSparSenSe (9.16) as

$$\mathbf{g}^{t+1} = \mathcal{P}_{\mathcal{G}} \left[(\mathbf{Z}\mathbf{I}_{n(n+1)})^{\varphi} (\mathbf{g}^t + \alpha m \mathbf{h}^t) \right], \text{ where} \tag{9.31}$$

$$\mathbf{g}^t = [(\mathbf{g}_1^{1,t})^T, \cdots, (\mathbf{g}_n^{1,t})^T, \cdots, (\mathbf{g}_1^{m,t})^T, \cdots, (\mathbf{g}_n^{m,t})^T]^T,$$
$$\mathbf{h}^t = [(\mathbf{h}_1^{1,t})^T, \cdots, (\mathbf{h}_n^{1,t})^T, \cdots, (\mathbf{h}_1^{m,t})^T, \cdots, (\mathbf{h}_n^{m,t})^T]^T,$$

where $\otimes$ stands for the Kronecker product. Note that now we can see DiSparSenSe as a subgradient method to minimize the function $-q(\mathbf{g})$, exactly as the recursion (8) in [132], and apply Theorem 2 of [132]. To do so, we first have to make sure their main assumptions hold. Assumptions 1, 2, and 4 of [132] hold in our case since the subgradient is bounded, the consensus matrix $\mathbf{Z}$ verifies the properties of Assumption 2, and Assumption 4 holds given that $\|\mathbf{G}_j^{i,t}\|_F \leqslant G,\ \forall i,\ j$. Let us also define $\mathbf{v}_j^{i,t} = \text{vec}(\mathbf{V}_j^{i,\varphi,t})$. Now, the term

$$\left\| \mathbf{v}_j^{i,0} - \frac{1}{m} \sum_{p=1}^{m} \mathbf{v}_j^{p,0} \right\| \tag{9.32}$$

is bounded since we initialize the algorithm with a fixed $\mathbf{G}_j^{i,0} = \mathbf{G}^0$, $\forall i,\ j$, and also because the subgradient is bounded. Given this, Theorem 2 of [132] yields the claim. ■

Notice that due to optimality, $q(\mathbf{G}^{i,t}) = \sum_{l=1}^{m} q_l(\mathbf{G}^{i,t})$ cannot be greater than $\hat{q}$, and therefore $\liminf_{t\to\infty} |q(\mathbf{G}^{i,t}) - \hat{q}| \leqslant m\psi_1(\alpha, Q, \varphi),\ \ i = 1, \cdots, m$, which guarantees convergence of the dual function to a bounded error floor around its optimal value. The requirement $\varphi \geqslant \bar{\varphi}$ is not too restrictive, as explained in [132]. Based on the definition of $\mathbf{g}_j^{i,t}$ and the definition of $\mathbf{v}_j^{i,t}$, let us define the two average vectors $\bar{\mathbf{g}}^t$ and $\bar{\mathbf{v}}^t$, defined as

$$\bar{\mathbf{g}}^t = \left[\frac{1}{m} \sum_{i=1}^{m} (\mathbf{g}_1^{i,t})^T, \ldots, \frac{1}{m} \sum_{i=1}^{m} (\mathbf{g}_n^{i,t})^T \right]^T,$$

$$\bar{\mathbf{v}}^t = \left[\frac{1}{m} \sum_{i=1}^{m} (\mathbf{v}_1^{i,t})^T, \ldots, \frac{1}{m} \sum_{i=1}^{m} (\mathbf{v}_n^{i,t})^T \right]^T,$$

as well as the following two supporting sequences

$$\mathbf{y}^t = \mathcal{P}_{\mathcal{G}}[\bar{\mathbf{v}}^{t-1}], \quad \mathbf{d}^t = \bar{\mathbf{g}}^t - \mathbf{y}^t. \tag{9.33}$$

For the supporting sequence $\mathbf{y}^t$ the following lemmas hold.

Lemma 9.2

The sequence $\mathbf{y}^t$ is updated with an ϵ-subgradient method [142] to maximize $q(\mathbf{y})$, that is

$$\mathbf{y}^{t+1} = \mathcal{P}_{\mathcal{G}}\Big[\mathbf{y}^t + \frac{\alpha}{m}\tilde{\mathbf{h}}^t\Big],$$

where the vector

$$\tilde{\mathbf{h}}^t = \sum_{i=1}^{m}(\mathbf{h}^{i,t} + \mathbf{d}^t/\alpha), \quad \textit{with } \mathbf{h}^{i,t} = [(\mathbf{h}_1^{i,t})^T, \ldots, (\mathbf{h}_n^{i,t})^T]^T,$$

is an ϵ-subgradient of $q(\mathbf{y})$ and $\epsilon = m\psi_2(\alpha, Q, \varphi)$. Notably, $\mathcal{P}_{\mathcal{G}}[.]$ stands for projection onto the convex set $\mathcal{G}$ and ψ_2 is a positive function of φ, α, and Q. Furthermore, $\mathbf{d}^t/\alpha$ is bounded, i.e., $\|\mathbf{d}^t/\alpha\| \leqslant \tau$, for a certain non-negative scalar τ, and

$$q_i(\mathbf{y}) \leqslant q(\mathbf{y}^t) + (\tilde{\mathbf{h}}^t)^T(\mathbf{y} - \mathbf{y}^t) + \psi_2(\alpha, Q, \varphi), \quad \forall \mathbf{y} \in \mathcal{G}.$$

Proof. The proof follows from the definition of the supporting sequences $\mathbf{y}^t$ and $\mathbf{d}^k$ in (9.33) and, in particular, directly from [132, Lemma 5 and Theorem 2] applied to our update sequence (9.31). ■

Lemma 9.3

For the supporting sequence $\mathbf{y}^t$ the followings hold.

(a)

$$-\sum_{k=1}^{t}(\tilde{\mathbf{h}}^k)^T\mathbf{y}^k \leqslant \frac{\|\mathbf{y}^1\|^2}{2\alpha/m} + t\,\frac{\alpha m\,(\sqrt{n}\,Q + \tau)^2}{2};$$

(b)

$$+\sum_{k=1}^{t}(\tilde{\mathbf{h}}^k)^T\hat{\mathbf{y}} \leqslant \frac{\|\mathbf{y}^1 - 2\hat{\mathbf{y}}\|^2}{2\alpha/m} + t\,\frac{\alpha m\,(\sqrt{n}\,Q + \tau)^2}{2} + t\,\psi_2(\alpha, Q, \varphi),$$

where $\hat{\mathbf{y}}$ is the optimal dual variable.

Proof. The result is rather standard and applies to any ϵ-subgradient method. A concise proof for the case $\epsilon = 0$, can be found in [130, Proposition 3-(a)]; extending it to any $\epsilon \geqslant 0$ is straightforward. ■

9.A.2 Primal Objective Convergence

In this subsection, we investigate the convergence of the running average cost $\|\hat{\mathbf{w}}^t\|_1$ to the optimal value of the primal cost $\|\hat{\mathbf{w}}\|_1$. Our analysis is formulated in the following theorem.

Theorem 9.2

Convergence of the primal running average sequence $\{\hat{\mathbf{w}}^t, \hat{\mathbf{u}}^t\}$ can be formulated as follows.

(a) The running average cost is upper bounded as

$$\|\hat{\mathbf{w}}^t\|_1 \leqslant \|\hat{\mathbf{w}}\|_1 + \frac{nG^2}{2t\alpha/m} + \frac{\alpha m\,(\sqrt{n}\,Q+\tau)^2}{2} + \tau m\sqrt{n}\,G + m\psi_2(\alpha, Q, \varphi). \quad (9.34)$$

(b) The running average cost is lower bounded as

$$\|\hat{\mathbf{w}}^t\|_1 \geqslant \|\hat{\mathbf{w}}\|_1 - \frac{9nG^2}{2t\alpha/m} - \frac{\alpha m\,(\sqrt{n}\,Q+\tau)^2}{2} - \tau m\sqrt{n}\,G - m\psi_2(\alpha, Q, \varphi). \quad (9.35)$$

where $\psi_2(\alpha, Q, \nu, \varphi)$ is a positive function, monotonically increasing with α and decreasing with φ. The non-negative scalar τ is defined in Lemma 9.2.

Proof. The proof is an adaptation of Proposition 3 in [130]. We start by the claim (a). By convexity of the primal cost $\|\cdot\|_1$ and the definition of $\bar{w}_i^t$ as a minimizer of the local Lagrangian functions, we have for $t \geqslant 1$,

$$\|\hat{\mathbf{w}}^t\|_1 \leqslant \frac{1}{t}\sum_{k=1}^{t}\|\bar{\mathbf{w}}^k\|_1 = \frac{1}{t}\sum_{k=1}^{t}\sum_{i=1}^{m}\Big(q_i(\mathbf{g}^{i,k}) - (\mathbf{g}^{i,k})^T\mathbf{h}^{i,k}\Big), \quad (9.36)$$

where

$$\mathbf{g}^{i,k} = [(\mathbf{g}_1^{i,k})^T, \ldots, (\mathbf{g}_n^{i,k})^T]^T.$$

By Lemma 9.2, since $\mathbf{g}^{i,k} \in \mathcal{G}$

$$q_i(\mathbf{g}^{i,k}) - q_i(\mathbf{y}^k) \leqslant (\mathbf{h}^{i,k})^T\mathbf{g}^{i,k} + (\mathbf{d}^k/\alpha)^T\mathbf{g}^{i,k} - (\mathbf{h}^{i,k} + \mathbf{d}^k/\alpha)^T\mathbf{y}^k + \psi_2(\alpha, Q, \varphi).$$

Next, by summing over i we have

$$\sum_{i=1}^{m} q_i(\mathbf{g}^{i,k}) \leqslant q(\mathbf{y}^k) + \sum_{i=1}^{m}(\mathbf{h}^{i,k})^T\mathbf{g}^{i,k} + \sum_{i=1}^{m}(\mathbf{d}^k/\alpha)^T\mathbf{g}^{i,k} - (\tilde{\mathbf{h}}^k)^T\mathbf{y}^k + m\psi_2(\alpha, Q, \varphi),$$

and thus,

$$\|\hat{\mathbf{w}}^t\|_1 \leqslant \frac{1}{t}\sum_{k=1}^{t}\Big(q(\mathbf{y}^k) + \sum_{i=1}^{m}(\mathbf{d}^k/\alpha)^T\mathbf{g}^{i,k} - (\tilde{\mathbf{h}}^k)^T\mathbf{y}^k + m\psi_2(\alpha, Q, \varphi)\Big). \quad (9.37)$$

We can use Lemma 9.3-(a) to find an upper bound for the term $-(\tilde{\mathbf{h}}^k)^T\mathbf{y}^k$. Besides, since $\|(\mathbf{d}^k/\alpha)^T\mathbf{g}^{i,k}\| \leqslant \|\mathbf{d}^k/\alpha\|\|\mathbf{g}^{i,k}\|$ and we known from Lemma 9.2 that $\|\mathbf{d}^k/\alpha\| \leqslant \tau$, together with $\|\mathbf{g}^{i,k}\| \leqslant \sqrt{n}\,G$ from our earlier assumption on bounded dual variable estimates, we obtain $\|(\mathbf{d}^k/\alpha)^T\mathbf{g}^{i,k}\| \leqslant \tau\sqrt{n}\,G$. With this in place, we can rewrite (9.37) as

$$\|\hat{\mathbf{w}}^t\|_1 \leqslant \frac{1}{t}\sum_{k=1}^{t} q(\mathbf{y}^k) + \frac{\|\mathbf{y}^1\|^2}{2t\alpha/m} + \frac{\alpha m\,(\sqrt{n}\,Q + \tau)^2}{2} + m\tau\sqrt{n}\,G + m\psi_2(\alpha, Q, \varphi).$$

We known that by optimality $q(\mathbf{y}^k) \leqslant \hat{q}$, by strong duality $\hat{q} = \|\hat{\mathbf{w}}\|_1$, and $\|\mathbf{y}^1\|^2 \leqslant nG^2$. Therefore, the claim (a) follows.

As for claim (b), given any optimal dual solution $\hat{\mathbf{y}}$, we have

$$\|\hat{\mathbf{w}}^t\|_1 = \underbrace{\|\hat{\mathbf{w}}^t\|_1 + (\hat{\mathbf{y}})^T\Big(\frac{1}{t}\sum_{k=1}^{t}\tilde{\mathbf{h}}^k\Big)}_{\omega} - (\hat{\mathbf{y}})^T\Big(\frac{1}{t}\sum_{k=1}^{t}\tilde{\mathbf{h}}^k\Big). \quad (9.38)$$

We also know that,

$$\omega = \|\hat{\mathbf{w}}^t\|_1 + (\hat{\mathbf{y}})^T\Big(\frac{1}{t}\sum_{k=1}^{t}\sum_{i=1}^{m}\mathbf{h}^{i,k}\Big) + m(\hat{\mathbf{y}})^T\Big(\frac{1}{t}\sum_{k=1}^{t}\mathbf{d}^k/\alpha\Big)$$

$$\geqslant \|\hat{\mathbf{w}}^t\|_1 + (\hat{\mathbf{y}})^T \Big(\sum_{i=1}^{m} \mathbf{h}^{i,k}(\hat{\mathbf{w}}^k) \Big) - m\sqrt{n}\, G\tau. \tag{9.39}$$

Furthermore, by the saddle point property of the Lagrangian function, i.e.,

$$\mathcal{L}(\hat{\mathbf{w}}^k, \hat{\mathbf{u}}^k, \hat{\mathbf{y}}) \geqslant \mathcal{L}(\hat{\mathbf{w}}, \hat{\mathbf{u}}, \hat{\mathbf{y}}) = \hat{q} = \|\hat{\mathbf{w}}\|_1,$$

we can write

$$\|\hat{\mathbf{w}}^t\|_1 + (\hat{\mathbf{y}})^T \Big(\sum_{i=1}^{m} \mathbf{h}^{i,k}(\hat{\mathbf{w}}^k) \Big) - m\sqrt{n}\, G\tau = \mathcal{L}(\hat{\mathbf{w}}^k, \hat{\mathbf{u}}^k, \hat{\mathbf{y}}) - m\sqrt{n}\, G\tau \geqslant \|\hat{\mathbf{w}}\|_1 - m\sqrt{n}\, G\tau. \tag{9.40}$$

We can now find an upper bound for the term $(\hat{\mathbf{y}})^T \Big(\frac{1}{t} \sum_{k=1}^{t} \tilde{\mathbf{h}}^k \Big)$ in (9.38) as in Lemma 9.3-(b). By substituting this bound in (9.38) and by combining it with (9.39)-(9.40), we obtain

$$\|\hat{\mathbf{w}}^t\|_1 \geqslant \|\hat{\mathbf{w}}\|_1 - m\tau\sqrt{n}G - \frac{\|\mathbf{y}^1 - 2\hat{\mathbf{y}}\|^2}{2t\alpha/m} - \frac{\alpha m(\sqrt{n}G + \tau)^2}{2} - m\psi_2(\alpha, Q, \varphi),$$

and since $\|\mathbf{y}^1 - 2\hat{\mathbf{y}}\|^2 \leqslant 9nG^2$, the claim follows. ■

PART

V

EPILOGUE

Those who cannot remember the past, are condemned to repeat it.

GEORGE SANTAYANA

10

Conclusions and Future Works

In this chapter, we provide concluding remarks on the main contributions of this thesis, and also highlight some of the possible future research directions.

10.1 Concluding Remarks

In this thesis, we have tried to revisit some of the main issues and challenges of wireless (sensor) networks (WSNs) from a different standpoint. What has distinguished our contributions from exsiting works is the concept of sparse reconstruction and compressive sensing (CS) which is somehow the main flavor of this thesis as well. We have shown that sparsity (in different domains, such as spatial sparsity) inherently exists in the structure of many of these problems. Throughout this thesis, we have attempted to explore such sparsity embedded within our problem structure and have come up with sparsity-aware solutions.

With regards to the organization of this thesis, except for this final part on conclusions and the first part giving an introduction and mathematical preliminaries, it is comprised of three main parts (Parts II-IV) containing our major contributions as is outlined in the following.

Part II, which contains two chapters, is devoted to our contributions to the context of mobile network localization. To be more specific, in Chapter 3, we have studied the problem of mobile network localization using only pairwise distance measurements. To do so, we have proposed to combine multidimensional scaling (MDS) with subspace perturbation expansion (SPE) in order to derive a model-independent dynamic MDS paradigm which could track a network of mobile nodes. In order to circumvent the need for a fully connected network of nodes, in both Chapters 3 and 4, we have proposed extensions which broaden the applicability of our proposed approach to partially connected networks where up to 50% of the pairwise distances can be missing.

Part III, which is comprised of three chapters, is dedicated to our contributions to multi-source localization. In particular, in Chapter 5, we have proposed a sparsity-aware multi-source time-difference-of-arrival (TDOA) localization paradigm which uses an innovative trick to attain a significant source identifiablity gain and also can localize multiple off-grid sources. In Chapter 6, we have proposed a sparsity-aware multi-source received-signal-strength (RSS) localization paradigm which exploits the information present in the cross-correlations of the received signal as well as in the different time lags of the correlation functions. In Chapters 6 and 7 (respectively in wireless channels and an underground medium), we have presented ideas which are blind to the source signal information and can even handle fully non-cooperative sources.

Part IV, which consists of two chapters, is devoted to our contributions to the context of sensor selection. More specifically, in Chapters 8 and 9, we have proposed a sparsity-aware sensor selection paradigm in order to a priori select the minimum number of sensors within a network to satisfy a mean squared error (MSE) estimation performance metric. Our ideas are developed in both centralized and distributed fashions for both uncorrelated and correlated noise experienced at different sensors.

10.2 Recommendations for Future Directions

Particular questions arise from the research results we have presented in this thesis. These questions mark some challenges for future research to broaden the applicability of our proposed methods. The major challenges are as follows.

- ◇ Performance investigation in real testbeds: A very general recommendation for future work is to implement and assess the performance of our proposed sparsity-aware network localization and sensor selection ideas in real wireless (sensor) network testbeds. Several unforeseen practical challenges will show up during establishing the real testbeds for implementing our ideas where each of which can introduce a previously unsolved problem.

- ◇ *Distributed dynamic MDS*: An interesting step forward in line with our contributions on dynamic MDS in Chapter 3, is to see how our subspace update rules can be distributed all over the network to devise a distributed dynamic MDS paradigm. To be more specific, one has to figure out a way to keep updating D (number of embedding dimensions) eigenvalues and eigenvectors in a distributed fashion. Such a distributed approach can tolerate even a higher level of partial connectivity (much higher than 50%) and it is scalable as well making it a more suitable choice for large-scale WSNs.

- ◇ *Multi-source TDOA localization for indoor scenarios*: Our sparsity-aware TDOA localization algorithm in Chapter 5 is mainly aimed at outdoor scenarios where harsh multipath effects do not exist. Therefore, extensions to handle multipath and non-line-of-sight (NLOS) effects which can complicate or degrade our approach are highly desirable. A possible workaround which is already mentioned in Chapter 5 is to use multipath disambiguation techniques. However, a general extension which can possibly benefit from multipath reflections as in our RSS-based approach in Chapter 6 is an interesting research direction.

- ◇ *Distributed sparsity-aware multi-source localization*: The algorithms we have proposed in Part III are centralized by definition. It means the sensors or access points (APs), depending upon the scenario, are assumed to be connected to a fusion center (FC). This FC constructs the overall linear set of equations $\mathbf{y} = \boldsymbol{\Psi}\boldsymbol{\theta}$ and recovers the sparse $\boldsymbol{\theta}$. An interesting extension is to eliminate the FC and conduct this with distributed sensors/APs. There are two possible cases to study. First, the case where sensors can only communicate with their neighboring sensors; however, they can sense the full $\boldsymbol{\theta}$. Second, which is even more interesting, is the case where the sensors only sense a part of $\boldsymbol{\theta}$ and these parts can be overlapping. A possible solution to these problems is to exploit a modified version of the distributed least absolute shrinkage and selection operator (LASSO) for linear regression within networks such as the one proposed in [143].

- ◇ *More informative microseismic monitoring*: In Chapter 7, we have confined ourselves to finding the closest grid points to the off-grid sources. However, the possibility to derive the relationship between the hypocenter of an off-grid source and its corresponding received displacement traces should be investigated. Evidently, the simpler the medium (single-layer homogenous in the best case), the easier such relationships can be discovered. In such a case, our ideas in Chapter 5 for off-grid source localization can be employed to localize off-grid seismic sources with higher accuracy. Moreover, in Chapter 7, we have only estimated the hypocenters of the sources and the normalized amplitudes of the moment tensors. Even though these two parameters are the most important ones we would like to extract, according to our results there is a possibility to further process the results and estimate the exact moment tensor amplitudes as well as the corresponding origin-times. This is another interesting extension which is worthy of giving some thought.

- ◇ *Budget constraints for distributed sensor selection*: In Chapter 9, we have

omitted adding other constraints to our subjective performance metric, such as power budget constraints on the sensors or logical constraints on their status of being active/inactive. Adding such constraints yields a more realistic WSN performance model. We expect that involving such constraints into our optimization problems would lead to the selection of different subsets of sensors. As an example, a total budget constraint can be enforced by adding a weighted sum inequality as $\sum_i c_i w_i \leqslant B_{\text{tot.}}$ to our subjective part, where the c_i's help to model budget limitations of the sensors.

- ◇ *Dynamic distributed sensor selection*: Our proposed distributed algorithms in Part IV fail to operate in highly varying environments where regressors can alter quickly (e.g., less than a second). In such a case, the regressors will acquire new values before our algorithms converge and thus the selected sensors might be invalid. This motivates extending our proposed algorithms to dynamic ones in order to operate in highly time-varying environments. To do so, some information about this time-varying process is required. Putting this together with the measurement equations we already have, one should devise appropriate distributed Kalman filtering algorithms to dynamically estimate the selected sensors.

Alongside these specific research directions directly arising from our contributions in this thesis, there exist a number of related research questions that can be considered for possible future work. These problems which are typically more high-level are enumerated in the following.

- ◇ *Dynamic MDS in other contexts*: The low-complexity dynamic MDS we have proposed and its extensions to handle missing connections in Part II can, in principle, be exploited for applications other than mobile network localization. Even though we have adapted our approach to fit into a network localization framework, there are several other application domains where MDS is known to be a solution (see Chapter 1 for examples such as machine learning) and in some of those the dissimilarities are dynamically changing. For such cases, our proposed dynamic MDS can be applied with minimum modifications to track the variations of the configuration of comprising components.

- ◇ *Sparsity-aware feature assignment*: We believe that our sparsity-aware approach to solve the multi-source TDOA localization problem in Chapter 5 can be generalized to cast a sparsity-aware feature assignment problem. Suppose that there is a large dictionary of fingerprints/features or a large basis

comprised of elements and one has to find out the contributing features/elements based on multiple measurements which are weighted versions of those features. In cases where the number of contributing features is much smaller than the cardinality of the dictionary/basis, this sort of problems can be solved using our ideas in Chapter 5. Deriving performance bounds in order to quantify how much we gain by applying nonlinear functions and creating new sets of measurements is another interesting problem but of a formidable nature!

◇ *Seismic geophone selection*: An interesting domain where our sensor selection ideas can be applied is seismic monitoring. The underground medium changes extremely slow compared to wireless channels or similar media, and thus the regressors observed by geophones change slowly. In practice, significant variations that will affect the observed regressors might happen every ten to hundred years. This motivates applying our sensor selection ideas in Part IV to microseismic monitoring problems in general. More specifically though, if one would like to apply our sensor selection ideas to *sparsity-aware* microseismic localization as in Chapter 7, then sparsity appears in two domains: first, the spatial sparsity of sources, and second, the sparsity of sensors to be activated. Therefore, the performance constraint to be satisfied is no longer the MSE of estimation, and a proper metric related to the quality of sparse reconstruction (possibly related to the restricted isometry property (RIP)) should be considered. Nonetheless, our ideas immediately apply to traditional inversion-based approaches which are based on using classical least squares (LS).

BIBLIOGRAPHY

[1] I. Akyildiz, W. Su, Y. Sankarasubramanian, and Y. Cayirci, "Wireless sensor networks: A survey," *Computer Networks Journal*, vol. 38, pp. 393–422, Mar. 2002.

[2] N. B. Priyantha, A. Miu, and H. Balakrishnan, "The cricket compass for contextaware applications," *Proceeding of ACM Intl. Conf. on Mobile Computing and Networking (MOBICOM)*, pp. 1–4, Jul. 2001.

[3] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[4] A. Mainwaring, D. Culler, J. Polastre, and R. S. J. Anderson, "Wireless sensor networks for habitat monitoring," *Proc. of ACM Int. Workshop Wireless Sensor Netw. Applicat. (WSNA)*, pp. 88–97, 2002.

[5] R. Fontana, E. Richley, and J. Barney, "Commercialization of an ultra wideband precision asset location system," *Proc. of Int. Conf. Ultra-Wideband (ICUWB)*, pp. 369–373, Nov. 2003.

[6] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Sig. Proc. Magazine*, vol. 22, pp. 54 – 69, Jul. 2005.

[7] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. of the IEEE*, vol. 97, pp. 427–450, Apr. 2009.

[8] T. T. Cai and L. Wang, "Mobile positioning using wireless networks: Possibilities and fundamental limitations based on available wireless network measurements," *IEEE Trans. on Info. Theory*, vol. 22, pp. 41–53, Jul. 2005.

[9] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," *Proc. of Intl. Conf. on Mobile Computing and Networking (ACM MOBICOM)*, pp. 81–95, Sep. 2003.

[10] Y. Shang, W. Ruml, Y.Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Trans. on Parallel and Dist. Systems*, vol. 15, pp. 961–974, Nov. 2004.

[11] D. B. Jourdan, D. Dardari, and M. Z. Win, "Position error bound for UWB localization in dense cluttered environments," *IEEE Trans. Aerospace and Elec. Syst.*, vol. 44, pp. 613–628, Apr. 2008.

[12] B. L. J. S. A. G. Dempster and C. Rizos, "Indoor positioning techniques based on wireless LAN," *Proc. of IEEE Intl. Conf. on Wireless Broadband and Ultra Wideband Comm.*, Mar. 2006.

[13] D. Madigan, E. Elnahrawy, R.R.Matrin, W. Ju, P. Krishnan, and A. Krishnakumar, "Bayesian indoor positioning systems," *Proc. of IEEE Intl. Conf. on Computer Comm. (INFOCOM)*, pp. 1217 – 1227, Mar. 2005.

[14] K. W. Cheung and H. C. So, "A multidimensional scaling framework for mobile location using time-of-arrival measurements," *IEEE Trans. on Sig. Proc.*, vol. 53, pp. 460–470, Feb. 2005.

[15] D. Macagnano and G. T. F. de Abreu, "Tracking multiple targets with multidimensional scaling," *Proc. of Intl. Symp. on Wireless Personal Multimedia Comms. (WPMC)*, pp. 1–5, Sep. 2006.

[16] F. K. W. Chan and H. C. So, "Efficient weighted multidimensional scaling for wireless sensor networks," *IEEE Trans. on Sig. Proc.*, vol. 57, pp. 4548–4553, Nov. 2009.

[17] Z. Wang and S. A. Zekavat, "A new TOA-DOA node localization for mobile ad-hoc networks: Achieving high performance and low complexity," *Proc. of Intl. Conf. on Telecomm. (ICT)*, pp. 836–842, 2010.

[18] U. Ferner, H. Wymeersch, and M. Z. Win, "Cooperative anchor-less localization for large dynamic networks," *Proc. of Intl. Conf. on Ultra-Wideband (ICUWB)*, pp. 181–185, Sep. 2008.

[19] M. A. Caceres, F. Sottile, R. Gallero, and M. A. Spirito, "Hybrid GNSS-ToA localization and tracking via cooperative unscented kalman filter," *Proc. of Intl. Symp. on Personal, Indoor and Mobile Radio Comms. (PIMRC)*, pp. 272–276, Sep. 2010.

[20] L. Dong, "Cooperative network localization via node velocity estimation," *Proc. of IEEE Wireless Comm. and Networking Conf. (WCNC)*, pp. 1–6, Apr. 2009.

[21] H. Wymeersch, U. Ferner, and M. Z. Win, "Cooperative bayesian self-tracking for wireless networks," *IEEE Comms. Letters*, vol. 12, pp. 505–507, Jul. 2008.

[22] K. Cheung, H. C. So, W.-K.Ma, and Y. T. Chan, "A constrained least squares approach to mobile positioning: Algorithms and optimality," *EURASIP Journal on Advances in Sig. Proc.*, vol. 2006, Dec. 2006.

[23] H. C. So, Y. T. Chan, and F. K. W. Chan, "Closed form formulae for time difference of arrival estimation," *IEEE Trans. on Sig. Proc.*, vol. 56, pp. 2614–2620, Jun. 2008.

[24] D. Malioutov, M. Çetin, and A. S. Wilsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. on Sig. Proc.*, vol. 53, pp. 3010–3022, Aug. 2005.

[25] C. R. Comsa, A. M. Haimovich, S. C. Schwartz, Y. H. Dobyns, and J. A. Dabin, "Source localization using time difference of arrival within a sparse representation framework," *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Sig. Proc. (ICASSP)*, pp. 2872 – 2875, May 2011.

[26] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Received signal strength based indoor positioning using compressive sensing," *IEEE Trans. on Mobile Computing*, vol. 11, pp. 1983 – 1993, Dec. 2011.

[27] S. Nikitaki and P. Tsakalides, "Localization in wireless networks via spatial sparsity," *Proc. of Asilomar Conf. on Signals, Systems and Computers*, pp. 236 – 239, Nov. 2010.

[28] D. M. G. Tzagkarikis, P. Jacquet, and P. Tsakalides, "Indoor positioning in wireless LANs using compressive sensing signal-strenghth fingerprinting," *Proc. of European Siganl Proc. Conf. (EUSIPCO)*, pp. 1776 – 1780, Aug.-Sep. 2011.

[29] B. Zhang, X. Cheng, N. Zhang, Y. Cui, Y. Li, and Q. Liang, "Sparse target counting and localization in sensor networks based on compressive sensing," *Proc. of IEEE Intl. Conf. on Computer Comm. (INFOCOM)*, pp. 2255 – 2263, Apr. 2011.

[30] A. Lombard, T. Rosenkranz, H. Buchner, and W. Kellermann, "Multidimensional localization of multiple sound sources using averaged directivity patterns of blind source separation systems," *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Sig. Proc. (ICASSP)*, pp. 233–236, Apr. 2009.

[31] J. Scheuing and B. Yang, "Disambiguation of TDOA estimation for multiple sources in reverberant environtments," *IEEE Trans. on Audio, Speech, and Language Proc.*, vol. 16, pp. 1479–1489, Nov. 2008.

[32] J. J. Xiao, A. Ribeiro, Z.-Q. Lou, and G. B. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Sig. Proc. Mag.*, vol. 23, pp. 27–41, Jul. 2006.

[33] S. Kumar, F. Zhao, and D. Shepherd, "Special issue on collaborative information processing in microsensor networks," *IEEE Sig. Proc. Mag.*, vol. 19, pp. 13–14, Mar. 2002.

[34] A. Bertrand, *Signal Processing Algorithms For Wireless Acoustic Sensor Networks*. PhD thesis, Faculty of Engineering, Katholieke Universiteit Leuven, May 2011.

[35] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. on Sig. Proc.*, vol. 57, no. 2, pp. 451–462, 2009.

[36] D. Bajović and J. X. B. Sinopoli, "Sensor selection for event detection in wireless sensor networks," *IEEE Trans. on Sig. Proc.*, vol. 59, no. 10, pp. 4938–4953, 2011.

[37] F. Altenbach, S. Corroy, G. Böcherer, and R. Mathar, "Strategies for distributed sensor selection using convex optimization," *Proc. of IEEE Global Comms. Conf. (GLOBECOM)*, pp. 2367–2372, Dec. 2012.

[38] V. Kekatos and G. Giannakis, "From sparse signals to sparse residuals for robust sensing," *IEEE Trans. on Sig. Proc.*, vol. 59, pp. 3355–3368, Jul. 2011.

[39] I. Schizas, "Distributed informative-sensor identification via sparsity-aware matrix decomposition," *IEEE Trans. on Sig. Proc.*, vol. 61, pp. 4610–4624, Sep. 2013.

[40] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1 – 27, 1964.

[41] J. A. Costa, N. Patwari, and A. O. Hero, "Distributed weighted MDS for node localization in sensor networks," *ACM Trans. on Sensor Networks*, vol. 2, pp. 39–64, Feb. 2006.

[42] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*. Boca Raton, FL: Chapman Hall/CRC, 2001.

[43] J. B. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 22, pp. 2319–2323, 2000.

[44] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Newyork: Springer, 1997.

[45] J. C. Gower, "Properties of euclidean and non-euclidean distance matrices," *Linear Algebra and its Applications*, vol. 67, pp. 81–97, 1985.

[46] J. B. Kruskal, "Nonmetric multidimensional scaling: A numerical method," *Psychometrika*, vol. 29, no. 2, pp. 115 – 129, 1964.

[47] J. de Leeuw, "Convergence of the majorization method for multidimensional scaling," *Journal of Classification*, vol. 5, no. 2, pp. 163 – 180, 1964.

[48] J. de Leeuw, "Algorithms in multidimensional scaling," *Conceptual and Numerical Analysis of Data*, pp. 159 – 177, 1989.

[49] Y. Takane, F. Young, and J. D. Leeuw, "Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling feature," *Psychometrika*, vol. 42, no. 1, pp. 7 – 67, 1977.

[50] R. J. Vaccaro, "Weighted subspaces fitting using subspace perturbation expansions," EAST-SISTA/TR 1997-45, Katholieke Universiteit Leuven, Dept. of Elec. Eng, Jun. 1997.

[51] R. J. Vaccaro, "Weighted subspaces fitting using subspace perturbation expansions," *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Sig. Proc. (ICASSP)*, pp. 1973–1976, May 1998.

[52] G. H. Golub and C. V. Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 3rd ed., 1996.

[53] R. G. Baraniuk, "More is less: Signal processing and the data deluge," *Science*, vol. 331, no. 6018, pp. 717–719, 2011.

[54] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Sig. Proc. Magazine*, vol. 25, pp. 21–30, Mar. 2008.

[55] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok, "Introduction to compressed sensing," in *Compressed Sensing: Theory and Applications*, Cambridge University Press, 2012.

[56] M. Davenport, "The fundamentals of compressive sensing, part 2." Video tutorial. Available at IEEE SigView.

[57] E. J. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2005.

[58] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. on Info. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[59] J. D. Blanchard, C. Cartis, and J. Tanner, "Compressed sensing: How sharp is the restricted isometry property?," *SIAM Rev.*, vol. 53, no. 1, pp. 105–125, 2011.

[60] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. on Info. Theory*, vol. 52, pp. 5406–5425, Dec. 2006.

[61] M. Rudelson and R. Vershynin, "On sparse reconstruction from fourier and gaussian measurements," *Comm. on Pure and Applied Math.*, vol. 61, pp. 1025–1045, Aug. 2008.

[62] E. Candes, J. R. T., and Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Info. Theory*, vol. 52, pp. 489–509, Feb. 2006.

[63] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. on Sig. Proc.*, vol. 41, pp. 3397–3415, Dec. 1993.

[64] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. on Info. Theory*, vol. 53, pp. 4655–4666, Dec. 2007.

[65] D. Needell and J. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, pp. 301–321, Dec. 2009.

[66] S. S. Chen, D. L. Donoho, and M. A. Sunders, "Atomic decomposition by basis pursuit," *SIAM Journal of Scientific Computing*, vol. 20, pp. 33–61, Jan. 1998.

[67] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2009.

[68] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[69] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B*, vol. 68, pp. 49–67, Feb. 2006.

[70] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1 – 122, 2010.

[71] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 1997.

[72] Y. Shang, W. Rumi, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," *Proc. of ACM Intl. Symp. on Mobile Ad Hoc Networking and Computing*, pp. 201–212, Apr. 2003.

[73] F. Caballero, L. Merrino, I. Maza, and A. Ollero, "A particle filtering method for wireless sensor network localization with an aerial robot beacon," *Proc. of Intl. Conf. on Rob. and Aut. (ICRA)*, pp. 596– 601, May 2008.

[74] H. Jamali Rad, T. van Waterschoot, and G. Leus, "Cooperative localization using efficient Kalman filtering for mobile wireless sensor networks," *Proc. of European Sig. Proc. Conf. (EUSIPCO)*, pp. 1984–1988, Aug.-Sep. 2011.

[75] H. Jamali Rad, A. Amar, and G. Leus, "Cooperative mobile network localization via subspace tracking," *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Sig. Proc. (ICASSP)*, pp. 2612–2615, May 2011.

[76] H. Jamali Rad, T. van Waterschoot, and G. Leus, "Anchorless cooperative localization for mobile wireless sensor networks," *Proc. of the Joint WIC/IEEE SP Symposium on Information Theory and Signal Processing in the Benelux*, pp. 1–8, May 2011.

[77] Y. Hua, "Asymptotical orthonormalization of subspace matrices without square root," *IEEE Sig. Proc. Magazine*, vol. 21, pp. 56–61, Jul. 2004.

[78] S. M. Kay, *Fundamentals of Statistical Sig. Proc.: Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[79] P. Tichavski, H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. on Sig. Proc.*, vol. 46, pp. 1386–1396, May 1998.

[80] P. Drineas, A. Javed, M. Magdon-Ismail, G. Pandurangan, R. Virrankoski, and A. Savvides, "Distance matrix reconstruction from incomplete distance information for sensor network localization," *Proc. of IEEE Intl. Conf. on Sensing, Comm. and Networking (SECON)*, pp. 536–544, Sep. 2006.

[81] P. A. Regalia and J. Wang, "On distance reconstruction for sensor network localization," *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Sig. Proc. (ICASSP)*, pp. 2866–2869, 2010.

[82] J. C. Platt, "FastMap, MetricMap and Landmark MDS are all Nyström algorithms," *Proc. of Intl. Workshop on Artificial Intelligence and Statistics*, pp. 261–268, 2005.

[83] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free postioning in mobile ad-hoc networks," *Proc. of the Intl. Conf. on System Sciences*, pp. 1 – 10, Jan. 2001.

[84] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2009.

[85] D. L. Donoho, "Compressive sensing," *IEEE Trans. on Info. Theory*, vol. 52, pp. 1289–1306, Sep. 2006.

[86] V. Cevher and R. G. Baraniuk, "Compressive sensing for sensor calibration," *Proc. of IEEE Sensor Array and Multichannel Sig. Proc. Workshop (SAM)*, Jul. 2008.

[87] V. Cevher, M. F. Durate, and R. G. Baraniuk, "Distributed target localization via spatial sparsity," *Proc. of European Siganl Proc. Conf. (EUSIPCO)*, Aug. 2009.

[88] S. Nikitaki and P. Tsakalides, "Localization in wireless networks based on jointly comressed sensing," *Proc. of European Siganl Proc. Conf. (EUSIPCO)*, pp. 1809 – 1813, Aug.-Sep. 2011.

[89] H. Jamali-Rad, H. Ramezani, and G. Leus, "Sparse multi-target localization using cooperative access points," *Proc. of IEEE Sensor Array Multichannel Proc. Workshop (SAM)*, pp. 353 – 356, Jun. 2012.

[90] L. Yang and K. C. Ho, "An approximately efficient TDOA localization algorithm in closed-form for locating multiple disjoint sources with erroneous sensor positions," *IEEE Trans. on Sig. Proc.*, vol. 57, pp. 4598–4615, Dec. 2009.

[91] A. Brutti, M. Omologo, and P. Svaizer, "Multiple source localization based on acoustic map de-emphasis," *EURASIP Journal on Audio, Speech, and Music Proc.*, Nov. 2010.

[92] C. Blandin, A. Ozerov, and E. Vincent, "Multi-source TDOA estimation in reverberant audio using angular spectra and clustering," *Sig. Proc.*, vol. 92, pp. 1950–1960, Aug. 2012.

[93] C. R. Comsa, A. M. Haimovich, S. C. Schwartz, Y. H. Dobyns, and J. A. Dabin, "Time difference of arrival based source localization within a sparse representation framework," *Proc. of Conf. on Info. Sciences and Systems (CISS)*, pp. 1–6, Mar. 2011.

[94] H. Jamali-Rad and G. Leus, "Sparsity-aware tdoa localization of multiple sources," *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Sig. Proc. (ICASSP)*, pp. 4021 – 4025, May 2013.

[95] H. Zhu, G. Leus, and G. B. Giannakis, "Sparsity-cognizant total least-squares for purturbed compressive sampling," *IEEE Trans. on Sig. Proc.*, vol. 59, pp. 2002–2016, May 2011.

[96] K. C. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. on Acoustic, Speech, Sig. Proc.*, vol. 24, pp. 320–327, Aug. 1976.

[97] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Trans. on Info. Theory*, vol. 57, pp. 4680–4688, Jul. 2011.

[98] G. B. Giannakis, G. Mateo, S. Farahmand, V. Kekatos, and H. Zhu, "USPACOR: Universal sparsity-controlling outlier rejection," *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Sig. Proc. (ICASSP)*, pp. 1952– 1955, May 2011.

[99] Y. Wang and G. Leus, "Reference-free time-based localization for an asynchronous target," *EURASIP Journal on Advances in Sig. Proc.*, vol. 2012, Jan. 2012.

[100] B. Friedlander, "A passive localization algorithm and its accuracy analysis," *IEEE Journal of Oceanic Engineering*, vol. 12, pp. 234–245, Jan. 1987.

[101] C. Feng, S. Valaee, and Z. Tan, "Multiple target localization using compressive sensing," *Proc. of IEEE Global Comms. Conf. (GLOBECOM)*, pp. 1 – 6, Nov.-Dec. 2009.

[102] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Compressive sensing based positioning using RSS of WLAN access points," *Proc. of IEEE Intl. Conf. on Computer Comm. (INFOCOM)*, pp. 1 – 9, Mar. 2010.

[103] H. Jamali-Rad and G. Leus, "Sparsity-aware multi-source TDOA localization," *IEEE Trans. on Sig. Proc.*, vol. 61, no. 19, pp. 4874 – 4887, 2013.

[104] E. A. P. Habets, "Room impulse response generator," 2008. available at `http://home.tiscali.nl/ehabets/rir_generator.html`.

[105] C. Bingham, M. D. Godfrey, and J. W. Tukey, "Modem techniques of power spectrum estimation," *IEEE Trans. on Audio and Electroacoustics*, vol. 15, pp. 56–66, Jun. 1967.

[106] J. Chen and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Trans. Sig. Proc.*, vol. 54, pp. 4634–4643, Dec. 2006.

[107] CVX Research Inc., "CVX: Matlab software for disciplined convex programming, version 2.0 beta." `http://cvxr.com/cvx`, Sep. 2012.

[108] Z. Tian, G. Leus, and V. Lottici, "Detection of sparse signals under finite-alphabet constraints.," *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Sig. Proc. (ICASSP)*, pp. 2349–2352, May 2009.

[109] H. Zhu and G. Giannakis, "Exploiting sparse user activity in multiuser detection," *IEEE Trans. on Comm.*, vol. 59, pp. 454–456, Feb. 2011.

[110] T. N. D. W.-K. Ma and, K. M. Wong, Z.-Q. Luo, and P.-C. Ching, "Quasi-ML multiuser detection using semi-definite relaxation with application to synchronous cdma," *IEEE Trans. on Sig. Proc.*, vol. 50, pp. 912–922, Apr. 2002.

[111] R. R. Picard and R. D. Cook, "Cross-validation of regression models,," *J. Amer. Statist. Assoc.*, vol. 79, pp. 575–583, Sep. 1984.

[112] S. M. Kay, *Fundamentals of Statistical Sig. Proc.: Detection Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[113] P. R. Gill, A. Wang, and A. Molnar, "The in-crowd algorithm for fast basis pursuit denoising," *IEEE Trans. on Sig. Proc.*, vol. 59, pp. 4595–4605, Oct. 2011.

[114] A. Baig and T. Urbancic, "Microseismic moment tensors: A path to understanding fracture growth," *The Leading Edge*, vol. 29, pp. 320–324, 2010.

[115] L. Scognamiglio, E. Tinti, and A. Michelini, "Real-time determination of seismic moment tensor for the italian region," *Bull. of Seismic Soc. of Ame.*, vol. 99, pp. 2223–2242, 2009.

[116] Y. Gibowicz, "Seismicity induced by mining: Recent research," *Adv. Geophys.*, vol. 51, no. 6, pp. 1–53, 2009.

[117] A. Droujinine, J. Winsor, and K. Slauenwhite, "Microseismic elastic full waveform inversion for hydraulic fracture monitoring," *Proc. of EAGE Conf. and Exhibition, Extended Abstracts*, pp. 2312–2316, Jun. 2012.

[118] J. Li, H. Z. H. Kuleli, and M.N.Toksöz, "Focal mechanism determination using high frequency, full waveform information," *SEG Expanded Abstracts*, vol. 28, pp. 2312–2316, 2009.

[119] S.-J. Lee, B.-S. Huang, W.-T. Liang, and K.-C. Chen, "Grid-based moment tensor inversion technique by using 3-D Green's functions database: A demonstration of the 23 October 2004 taipei earthquake," *Terr. Atmos. Ocean. Sci.*, vol. 21, pp. 503–514, Jun. 2010.

[120] H. Jamali-Rad, H. Ramezani, and G. Leus, "Blind sparsity-aware multi-source localization," *Proc. of European Sig. Proc. Conf. (EUSIPCO)*, pp. 1 – 5, Sep. 2013.

[121] I. V. Rodriguez, M. Sacchi, and Y. J. Gu, "Simultaneous recovery of origin time, hypocentre location and seismic moment tensor using sparse representation theory," *Geophysical Journal Intl.*, vol. 188, pp. 1188–1202, 2012.

[122] I. V. Rodriguez, M. Sacchi, and Y. J. Gu, "A compressive sensing framework for seismic source parameter," *Geophysical Journal Intl.*, vol. 191, pp. 1226–1236, 2012.

[123] I. V. Rodrigues and M. D. Sacchi, "Microseismic source characterization combining compressive sensing with a migration-based methodology," *Proc. of EAGE Conf. and Exhibition*, Jun. 2013.

[124] R. Madariaga, "Seismic source theory," *Treatise on Geophysics*, pp. 59 – 82, 2007.

[125] K. Aki and P. Richards, *Quantitative Seismology*. Sausalito, CA: University Science Books, 2nd ed., 2002.

[126] Y. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Sig. Proc.*, vol. 58, pp. 3042–3054, Jun. 2010.

[127] Z. Tang, G. Blacquière, and G. Leus, "Aliasing-free wideband beamforming using sparse signal representation," *IEEE Trans. on Sig. Proc.*, vol. 59, pp. 3464–3469, Jul. 2011.

[128] D. Bickson and D. Dolev, "Distributed sensor selection using a truncated Newton method." submitted for publication, available on arXiv:0907.0931v2 [cs.IT].

[129] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[130] A. Nedić and A. Ozdaglar, "Approximate primal solutions and rate analysis for dual subgradient methods," *SIAM J. on Opt.*, vol. 19, no. 4, pp. 1757–1780, 2009.

[131] J. C. Duchi, A. Agarwal, and M. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. on Auto. Cont.*, vol. 57, no. 3, pp. 592–606, 2012.

[132] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, "Subgradient methods and consensus algorithms for solving convex optimization problems," *Proc. of IEEE Conf. on Decision and Cont. (CDC)*, pp. 4185–4190, Dec. 2008.

[133] H. Godrich, A. P. Petropulu, and H. V. Poor, "Sensor selection in distributed multiple-radar architectures for localization: A knapsack problem formulation," *IEEE Trans. on Sig. Proc.*, vol. 60, no. 1, pp. 6033–6043, 2012.

[134] E. Pollakis, R. L. G. Renato, and S. Stańczak, "Base station selection for energy efficient network operation with the majorization-minimization algorithm," *Proc. of IEEE Intl. Workshop on Sig. Proc. Advances in Wireless Comm. (SPAWC)*, pp. 219–223, Jun. 2012.

[135] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," *Proc. of Info. Proc. in Sensor Networks (IPSN)*, pp. 20–27, Apr. 2004.

[136] H. Jamali-Rad, A. Simonetto, and G. Leus, "Sparsity-aware sensor selection: Centralized and distributed algorithms," *IEEE Sig. Proc. Letters*, vol. 21, no. 2, pp. 217 – 220, 2014.

[137] S. Roy and R. A. Iltis, "Decentralized Linear Estimation in Correlated Measurement Noise," *IEEE Trans. on Aerospace and Elec. Sys.*, vol. 27, no. 6, pp. 939 – 941, 1991.

[138] X. R. Li, Y. Zhu, J. Wang, and C. Han, "Optimal Linear Estimation Fusion-Part I: Unified Fusion Rules," *IEEE Trans. on Info. Theory*, vol. 49, no. 9, pp. 2192 – 2208, 1991.

[139] H. Jamali-Rad, A. Simonetto, X. Ma, and G. Leus, "Sparsity-aware sensor selection for correlated noise," *Proc. of Intl. Conf. on Info. Fusion (FUSION)*, Jul. 2014.

[140] E. Cuthill and J. McKee, "Reducing the bandwidth of sparse symmetric matrices," *In Proc. of 24th National Confernce ACM*, pp. 157 – 172, 1969.

[141] U. A. Khan and J. F. Moura, "Distributed iterative-collapse inversion (DICI) algorithm for L-banded matrices," *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Sig. Proc. (ICASSP)*, pp. 2529 – 2532, Apr. 2008.

[142] K. Kiwiel, "Convergence of approximate and incremental subgradient methods for convex optimization," *SIAM Journal of Optimization*, vol. 14, no. 3, pp. 807–840, 2004.

[143] J. A. Bazerque, G. Mateos, and G. B. Giannakis, "Distributed LASSO for in-network linear regression," *Proc. of IEEE Intl. Conf. on Acoustics, Speech, and Sig. Proc. (ICASSP)*, pp. 2978–2981, Mar. 2010.

GLOSSARY

Abbreviations

WSN	Wireless sensor networks
RSS	Received signal strength
TOA/F	Time-of-arrival/flight
TDOA	Time-difference-of-arrival
AOA	Angle-of-arrival
DOA	Direction-of-arrival
SNR	Signal to noise ratio
AWGN	Additive white Gaussian
LOS	Line-of-sight
NLOS	Non-line-of-sight
RIR	Room impulse response
CPU	central processing unit
FC	Fusion center
GPS	Global positioning system
UWB	Ultrawide bandwidth
WLAN	Wireless local area network
CDMA	Code division multiple access CDMA
AP	Access point
MS	Mobile station
MDS	Multidimensional scaling
WMDS	Weighted MDS
dwMDS	Distributed weighted MDS
SPE	Subspace perturbation expansion

i.i.d.	Independent identically distributed
RIP	Restricted isometry property
CVD	Cross-validation
BPDN	Basis pursuit denoising
LASSO	Least absolute shrinkage and selection operator
G-LASSO	Group LASSO
ADMM	Alternating direction method of multipliers
DD	Dual decomposition
MM	Method of multipliers
KF	Kalman filter
EKF	Extended KF
UKF	Unscented KF
PF	Particle filter
FIM	Fisher information matrix
CRB	Cramér-Rao bound
PCRB	Posterior CRB
EVD	Eigenvalue decomposition
GS	Gram-Schmidt
PEST	Perturbation expansion-based subspace tracking
PIST	Power iteration-based subspace tracking
FLOPS	Floating point operations
MSE	Mean squared error
RMSE	Root MSE
PRMSE	Positioning RMSE
MC	Monte Carlo
IMNC	Imperfect measurement noise covariance
IPNC	Imperfect process noise covariance
GLR	Geometric link reconstruction
MGLR	Modified GLR
LAR	Linear algebraic reconstruction
MLAR	Modified LAR
SQRT	square root

SN	Source nodes
BSS	Blind source separation
CS	Compressive sampling
KNN	K-nearest neighbors
SMTL	Sparsity-aware multi-source TDOA localization
ESMTL	Enhanced SMTL
LS	Least squares
STLS	Sparse total least squares
CWLS	Constrained weighted least squares
WSSTLS	Weighted structured STLS
JDCS	Joint distributed CS
SRL	Sparsity-aware RSS localization
SRLC	Sparsity-aware RSS localization via cooperative APs
MMV	Multiple measurement vectors
MUD	Multiuser detection
FA	Finite alphabet
MP	Matching pursuit
OMP	Orthogonal MP
BOMP	Block OMP
MDG-LASSO	Multi-dictionary G-LASSO
ML	Maximum likelihood
SparSenSe	Sparsity-aware sensor selection
DiSparSenSe	Distributed SparSenSe
PSD	Positive semidefinite
SVD	Singular value decomposition
SDR	Semi-definite relaxation
LMI	Linear matrix inequality
KKT	Karush–Kuhn–Tucker

General Notations

$\approx$	Approximately equal to
$\propto$	Proportional to
x	Scalar x
$\mathbf{x}$	Vector $\mathbf{x}$
$\hat{\mathbf{x}}$	Estimate of $\mathbf{x}$
$[\mathbf{x}]_i$	i-th element of vector $\mathbf{x}$
$\mathbf{X}$	Matrix $\mathbf{X}$
$[\mathbf{X}]_{i,j}$	(i,j)-th element of $\mathbf{X}$
$\mathbf{X}^{-1}$	Matrix inverse
$\mathbf{X}^{\dagger}$	Matrix Penrose-Moore pseudo-inverse
$(.)^T$	Transpose operator
$(.)^H$	Hermitian operator
$\mathbf{I}_N$	Identity matrix of size $N \times N$
$\mathbf{0}_{M\times N}$	$M \times N$ matrix of all zeros
$\mathbf{1}_N$	$N \times 1$ vector of all ones
$\text{diag}(\mathbf{x})$	Diagonal matrix with $\mathbf{x}$ on its diagonal
$\text{diag}(\mathbf{X})$	Vector containing diagonal elements of $\mathbf{X}$
$\text{rank}(.)$	Rank operator
$\text{tr}(.)$	Trace operator
$\text{vec}(.)$	Vectorziation operation
$\text{ivec}(.)$	Inverse vectorziation operation
$\otimes$	Kronecker product
$\odot$	Element-wise Hadamard product
$*$	Convolution
$\|.\|_p$	ℓ_p norm
$\|.\|_F$	Frobenius norm
$\mathbb{E}(.)$	Statistical expectation
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^N$	Vector space of size N containing real numbers
$\mathbb{R}^{M\times N}$	Matrix space of size $M \times N$ containing real numbers
$\mathbb{N}$	Set of natural numbers

$\mathbb{N}_+$	Set of non-negative natural numbers
$\nabla_a f(a)$	(Sub)gradient of $f(.)$ w.r.t. a
$\mathcal{N}(m, \mathbf{C})$	Gaussian distribution with mean m and covariance $\mathbf{C}$
$\delta(t)$	Dirac impulse function
δ_n	Kronecker (unit) impulse function
$\lambda_{\text{max/min}}(\mathbf{X})$	Maximum/Minimum eigenvalue of $\mathbf{X}$
$\rho(\mathbf{X})$	Spectral radius of $\mathbf{X}$
$d(\mathbf{A}, \mathbf{B})$	Eucledian distance between points $\mathbf{A}$ and $\mathbf{B}$
max/min$(.)$	Maximum/minimum operator
$\mathcal{P}_{\geq 0}$	Projection over the cone of PSD matrices
$O(.)$	Order of complexity
$\mathcal{A} \cap \mathcal{B}$	Intersection of sets $\mathcal{A}$ and $\mathcal{B}$
$\lvert\mathcal{A}\rvert$	Cardinality of set $\mathcal{A}$

INDEX

SAMENVATTING

Draadloze netwerken hebben de hedendaagse wereld gerevolutionaliseerd door snelle verbindingen en goedkope diensten aan te bieden. Maar zelfs deze nooit geziene voorzieningen kunnen de drang naar meer geavanceerde technologieën niet vervullen. Daarom wordt er momenteel heel wat aandacht geschonken aan (mobiele) draadloze sensornetwerken bestaande uit goedkope sensoren die op een gedistribueerde manier taken kunnen uitvoeren in extreme omstandigheden. De unieke eigenschappen van de draadloze omgeving, de verhoogde complexiteit omwille van de mobiliteit, de gedistribueerde aard van de taakuitvoering, en de strikte performantie- en energiebeperkingen vormen een grote uitdaging voor onderzoekers om systemen te bedenken die een goede balans slaan tussen performantie en energieverbruik.

Wij bestuderen enkele fundamentele uitdagingen voor draadloze (sensor)netwerken zoals een efficiënt energieverbruik, schaalbaarheid, en besef van plaats. Wat ons onderzoek van de beschikbare literatuur onderscheidt is dat wij in onze probleemstellingen en systeemontwerpen gebruik maken van concepten gerelateerd aan de reconstructie van schaarse signalen en gecomprimeerde data-acquisitie. Wij buiten de schaarse structuren uit die aanwezig zijn in de bestudeerde modellen. Als de eerder vermelde uitdagingen vanuit dit perspectief worden bekeken, dan geeft dit niet alleen aanleiding tot een kostreductie omdat minder metingen nodig zijn, maar ook tot een aanvaardbare nauwkeurigheid als het systeem op de juiste manier ontworpen wordt.

We kijken in deze thesis eerst naar lokalisatie in mobiele draadloze netwerken. Gegeven de elegantie en eenvoud van de multidimensionele schaleringstechniek (MDS) voor netwerklokalisatie, combineren we deelruimteperturbatietheorie met klassieke MDS om zo een modelonafhankelijke dynamische versie van MDS te ontwikkelen waarmee een netwerk van mobiele nodes kan gelokaliseerd worden, enkel gebruik makende van paarsgewijze afstandsmetingen. Verder breiden we ons goedkoop dynamisch MDS paradigma op twee manieren uit naar netwerken die enkel gedeeltelijk verbonden zijn en waar sommige afstandsmetingen ontbreken. We bestuderen ook een modelafhankelijke versie van MDS waarbij het bewegings-

process van de nodes bekend is. In dat geval lineariseren we de nietlineaire afstandsmetingen tot de ankernodes en volgen we de positie van de mobiele nodes met behulp van een Kalman filter (KF) in plaats van een uitgebreid Kalman filter (EKF). Voor beide onderzoeksrichtingen laten we veelbelovende resultaten zien die de efficiëntie van onze voorgestelde methodes aantoont.

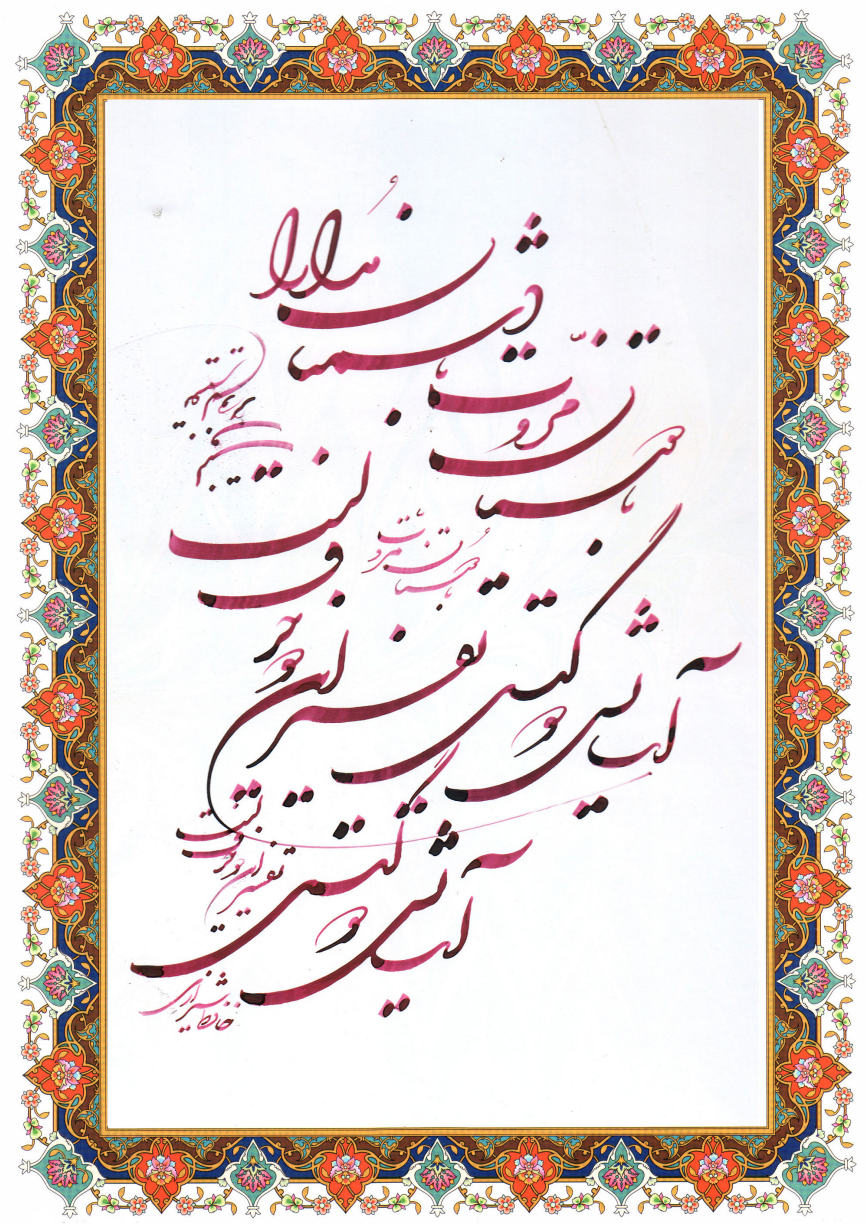
Daarna onderzoeken we een gerelateerd multi-bron lokalisatieprobleem waarbij sommige nodes in het netwerk zendbronnen zijn. Het feit dat deze bronnen niet van mekaar kunnen worden onderscheiden verhoogt de complexiteit van dit probleem. Het introduceert een complex toekenningsprobleem om de ontvangen signalen (typisch de som van de uitgezonden signalen) op te splitsen in de verschillende uitgezonden signalen en om die dan te lokaliseren. Wij stellen innovatieve ideeën voor om dit probleem op te lossen gebruik makende van tijdsverschil- (TDOA) en signaalsterktemetingen (RSS). De algemene aanpak die we voorstellen is gebaseerd op het herkennen van vingerafdrukken waarbij het desbetreffende spatiale domein wordt gediscretiseerd en iedere discrete positie nu een vingerafdruk heeft. Verder gebruiken we de schaarsheid van de bronnen in dit spatiale discrete domein en stellen oplossingen voor die een ongezien aantal bronnen kunnen lokaliseren. Ook breiden we onze TDOA-gebaseerde techniek uit om bronnen te kunnen lokaliseren die tussen de discrete posities in liggen. De RSS-gebaseerde methode, die zowel kan gebruikt worden voor binnenlokalisatie als voor het lokaliseren van ondergrondse microseismische activiteiten, wordt daarentegen uitgebreid naar een volledig blind scenario waarbij de statistiek van de bronsignalen niet gekend is. We presenteren uitvoerige simulatieresultaten die onze beweringen bevestigen.

Tenslotte vestigen we onze aandacht op het sensorselectieprobleem in draadloze netwerken. In lijn met het thema van deze thesis verkennen we de schaarsheid van de geselecteerde sensoren in vergelijking met het totaal aantal sensoren en stellen we oplossingen voor die gebaseerd zijn op deze schaarsheid. We doen dit voor gevallen waarbij de ruis ontvangen door de sensoren al dan niet gecorreleerd is. Om een centrale aanpak in zeer grote netwerken te vermijden, breiden we onze algoritmes ook uit naar gedistribueerde versies waarbij iedere sensor enkel communiceert met zijn buren en zelf beslist of hij deelneemt aan de probleemoplossing of niet. Gedetailleerde convergentiebewijzen, gekwantificeerde bewerkings- en communicatiekosten, alsook onze simulatieresultaten bevestigen de bruikbaarheid en efficiëntie van ons innovatief sensorselectieparadigma.

Zoektermen: Mobiele netwerklokalisatie, multi-bron lokalisatie, sensorselectie, reconstructie van schaarse signalen, gedistribueerde optimisatie.

About the Author

Hadi Jamali-Rad was born in Lahijan, a major tourist hub in northern Iran. He earned the B.Sc. degree in Electrical Engineering from the Iran University of Science and Technology (IUST), Tehran, Iran, in 2007 and the M.Sc. degree (with honors) in Electrical Engineering from the IUST, in 2010. In the meantime, from 2009 till 2010, he worked for industry as a part-time design engineer in telecommunications sector. In 2010, he joined the Circuits and Systems (CAS) group at the Delft University of Technology (TU Delft), Delft, The Netherlands, where he has been working ever since towards his Ph.D. degree under supervision of Prof. Geert Leus. In 2012, he visited the SISTA division of Katholieke Universiteit Leuven (KU Leuven) for which he won an Erasmus grant. In 2013, he did an internship with Shell Global Solutions B.V., The Netherlands. In 2014, he visited the Center for Signal and Information Processing (CSIP) at the Georgia Institute of Technology (Georgia Tech). He has been actively serving as the technical reviewer for several IEEE journals and major conferences. His general research interests lie at the confluence of signal processing, communications, and optimization. In particular, he is interested in sparse reconstruction and compressive sensing, cooperative estimation, and distributed optimization.



Persian (Nastaleeq) calligraphy of a Hāfez poem.

Poem message:

“The recipe for a peaceful life: be kind to friends, show respect to enemies”.

Hāfez: A prominent Persian poet (1326-1390 CE) from Shirāz, Irān.