DISTRIBUTED FIELD ESTIMATION IN WIRELESS SENSOR NETWORKS

By

Timothy Battisti

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
"SAPIENZA UNIVERSITY"
ROME, ITALY
APRIL 2010

To my family.

Table of Contents

Table of Contents **iv**

Introduction **1**

1 Distributed projection algorithms **10**
1.1 Introduction 10
1.2 Decentralized projection algorithms 13
1.2.1 Distributed polynomial approximation 20
1.3 Maximum convergence rate under topology constraints 28
1.4 Distributed projection algorithm robust against coupling noise 37
1.4.1 Trade-off between convergence speed and accuracy 40
1.5 Signal subspace order selection 43
1.5.1 Theory of Slepian functions 45
1.5.2 Basis Expansion Square Bias 49
1.5.3 Final considerations for the signal subspace order selection . . 51
1.6 A practical application: cooperative spectrum sensing using decentralized projection algorithms 52
1.7 Conclusions 61
1.8 Appendix A 62
1.9 Appendix B 64
1.10 Appendix C 68

2 Distributed estimation via Belief Propagation **69**
2.1 Introduction 69
2.2 The setting 72
2.2.1 Field Estimation via Belief Propagation 83
2.3 Field Estimation via Belief Propagation and multiple observations . . 90

Bibliography **100**

Introduction

Researchers see WSNs as an "exciting emerging domain of deeply networked systems of low-power wireless motes with a tiny amount of CPU and memory, and large federated networks for high-resolution sensing of the environment". Sensors in a WSN have a variety of purposes, functions, and capabilities. The field is now advancing under the push of recent technological advances and the pull of a myriad of potential applications. The radar networks used in air traffic control, the national electrical power grid, and nationwide weather stations deployed over a regular topographic mesh are all examples of early-deployment sensor networks; all of these systems, however, use specialized computers and communication protocols and consequently, are very expensive. Much less expensive WSNs are now being planned for novel applications in physical security, health care, and commerce. Sensor networking is a multidisciplinary area that involves, among others, radio and networking, signal processing, artificial intelligence, database management, systems architectures for operator-friendly infrastructure administration, resource optimization, power management algorithms, and platform technology (hardware and software, such as operating systems). The applications, networking principles, and protocols for these systems are just beginning to be developed. The near-ubiquity of the Internet, the advancements in wireless and wireline communications technologies, the network build-out (particularly in the

wireless case), the developments in IT (such as high-power processors, large random-access memory chips, digital signal processing, and grid computing), coupled with recent engineering advances, are in the aggregate opening the door to a new generation of low-cost sensors and actuators that are capable of achieving high-grade spatial and temporal resolution. The technology for sensing and control includes electric and magnetic field sensors; radio-wave frequency sensors; optical-, electrooptic-, and infrared sensors; radars; lasers; location/navigation sensors; seismic and pressure-wave sensors; environmental parameter sensors (e.g., wind, humidity, heat); and biochemical national securityoriented sensors. Todays sensors can be described as "smart" inexpensive devices equipped with multiple onboard sensing elements; they are low-cost low-power untethered multi-functional nodes. Sensor devices, or wireless nodes (WNs), are also (sometimes) called motes. A stated commercial goal is to develop complete microelectromechanical systems (MEMSs)based sensor systems at a volume of 1 mm^3. Sensors are internetworked via a series of multihop short-distance low-power wireless links (particularly within a defined sensor field); they typically utilize the Internet or some other network for long-haul delivery of information to a point (or points) of final data aggregation and analysis. Sensors are typically deployed in a high-density manner and in large quantities: a WSN consists of densely distributed nodes that support sensing, signal processing, embedded computing, and connectivity; sensors are logically linked by self-organizing means (sensors that are deployed in short-hop point-to-point masterslave pair arrangements are also of interest). WSNs have unique characteristics, such as, but not limited to, power constraints and limited battery life for the WNs, redundant data acquisition, low duty cycle, and, many-to-one flows. Consequently, new design methodologies are needed across a set of disciplines

including, but not limited to, information transport, network and operational management, confidentiality, integrity, availability, and, in-network/local processing. In some cases it is challenging to collect (extract) data from WNs because connectivity to and from the WNs may be intermittent due to a low-battery status (e.g., if these are dependent on sunlight to recharge) or other WN malfunction. Sensors span several orders of magnitude in physical size; they (or, at least some of their components) range from nanoscopic-scale devices to mesoscopic-scale devices at one end, and from microscopic-scale devices to macroscopic-scale devices at the other end. Nanoscopic (also known as nanoscale) refers to objects or devices on the order of 1 to 100 nm in diameter; mesoscopic scale refers to objects between 100 and 10,000 nm in diameter; the microscopic scale ranges from 10 to 1000 mm, and the macroscopic scale is at the millimeter-to-meter range. At the low end of the scale, one finds, among others, biological sensors, small passive microsensors (such as "Smart Dust"), and "lab-on-a-chip" assemblies. At the other end of the scale one finds platforms such as, but not limited to, identity tags, toll collection devices, controllable weather data collection sensors, bioterrorism sensors, radars, and undersea submarine traffic sensors based on sonars. Some refer to the latest generation of sensors, especially the miniaturized sensors that are directly embedded in some physical infrastructure, as microsensors. A sensor network supports any type of generic sensor; more narrowly, networked microsensors are a subset of the general family of sensor networks. Microsensors with onboard processing and wireless interfaces can be utilized to study and monitor a variety of phenomena and environments at close proximity. Sensors facilitate the instrumenting and controlling of factories, offices, homes, vehicles, cities, and the ambiance,

especially as commercial off-the-shelf technology becomes available. With sensor network technology (specifically, with embedded networked sensing), ships, aircraft, and buildings can "self-detect" structural faults (e.g., fatigue-induced cracks). Places of public assembly can be instrumented to detect airborne agents such as toxins and to trace the source of the contamination should any be present (this can also be done for ground and underground situations). Earthquake-oriented sensors in buildings can locate potential survivors and can help assess structural damage; tsunami-alerting sensors are useful for nations with extensive coastlines. Sensors also find extensive applicability on the battlefield for reconnaissance and surveillance. Implementations of WSNs have to address a set of technical challenges; however, the move toward standardization will, in due course, minimize a number of these challenges by addressing the issues once and then result in off-the-shelf chipsets and components. A current research and development (RD) challenge is to develop low-power communication with low-cost on-node processing and selforganizing connectivity/protocols; another critical challenge is the need for extended temporal operation of the sensing node despite a (typically) limited power supply (and/or battery life). In particular, the architecture of the radio, including the use of low-power circuitry, must be properly selected. In practical terms this implies low power consumption for transmission over low-bandwidth channels and low-power-consumption logic to preprocess and/or compress data. Energy efficient wireless communications systems are being sought and are typical of WSNs. Low power consumption is a key factor in ensuring long operating horizons for non-power-fed systems (some systems can indeed be power-fed and/or rely on other power sources). In general we taxonomize (commercial) sensor networks and systems into two categories:

1. Category 1 WSNs (C1WSNs): almost invariably mesh-based systems with multihop radio connectivity among or between WNs, utilizing dynamic routing in both the wireless and wireline portions of the network. Military theater systems typically belong to this category.

2. Category 2 WSNs (C2WSNs): point-to-point or multipoint-to-point (starbased) systems generally with single-hop radio connectivity to WNs, utilizing static routing over the wireless network; typically, there will be only one route from the WNs to the companion terrestrial or wireline forwarding node (WNs are pendent nodes). Residential control systems typically belong to this category.

C1WSNs support highly distributed high-node-count applications (e.g., environmental monitoring, national security systems); C2WSNs typically support confined short-range spaces such as a home, a factory, a building, or the human body. C1WSNs are different in scope and/or reach from evolving wireless C2WSN technology for short-range low-data-rate wireless applications such as RFID (radio-frequency identification) systems, light switches, fire and smoke detectors, thermostats, and, home appliances. C1WSNs tend to deal with large-scale multipoint-to-point systems with massive data flows, whereas C2WSNs tend to focus on short-range point-to-point, source-to-sink applications with uniquely defined transaction-based data flows. Traditionally, sensor networks have been used in the context of high-end applications such as radiation and nuclear-threat detection systems, over-the-horizon weapon sensors for ships, biomedical applications, habitat sensing, and seismic monitoring. More recently, interest has focusing on networked biological and chemical sensors for national security applications; furthermore, evolving interest extends to direct consumer applications. Existing and potential applications of sensor networks include, among others,

military sensing, physical security, air traffic control, traffic surveillance, video surveillance, industrial and manufacturing automation, process control, inventory management, distributed robotics, weather sensing, environment monitoring, national border monitoring, and building and structures monitoring . A short list of applications follows. Traditionally, sensor networks have been used in the context of high-end applications such as radiation and nuclear-threat detection systems, "over-the-horizon" weapon sensors for ships, biomedical applications, habitat sensing, and seismic monitoring. More recently, interest has focusing on networked biological and chemical sensors for national security applications; furthermore, evolving interest extends to direct consumer applications. Existing and potential applications of sensor networks include, among others, military sensing, physical security, air traffic control, traffic surveillance, video surveillance, industrial and manufacturing automation, process control, inventory management, distributed robotics, weather sensing, environment monitoring, national border monitoring, and building and structures monitoring. A short list of applications follows.

- Military applications
  - Monitoring inimical forces
  - Monitoring friendly forces and equipment
  - Military-theater or battlefield surveillance
  - Battle damage assessment and more...
- Environmental applications
  - Microclimates

  - Forest fire detection
  - Flood detection
  - Precision agriculture

- Commercial applications
-
  - Environmental control in industrial and office buildings
  - Vehicle tracking and detection
  - Inventory control
  - Traffic flow surveillance and more . . .

- Home applications
-
  - Home automation
  - Instrumented environment
  - Automated meter reading and more . ..

Wireless sensor networks have attracted considerable attention in recent years. Research in this area has focused on two separate aspects of such networks: networking issues, such as capacity, delay, and routing strategies; and application issues. This work is concerned with the second of these aspects, and in particular with the problem of distributed estimation of a physical field of interest, for example temperature or gas concentration distribution. Because of limited battery level and sensor complexity, the measurements gathered by the single node of a sensor network may be highly unreliable. Improving the reliability of the individual node would require higher complexity and cost, but this would negatively affect the economy of scale, which is a

fundamental concern in large scale sensor networks. It is then particularly important to improve the accuracy of each sensor by exploiting the interaction with the other nodes. This is possible if the environment monitored by the network exhibits a spatial correlation, which is typically the case in many physical field of interest, like in the distribution of temperatures or the concentration of a given contaminant. Moreover, considering that wireless sensor networks are typically characterized by limited communication capabilities due to tight energy and bandwidth limitations, any distributed mechanism for noise reduction is clearly beneficial to avoid the transmission of redundant bits. Centralized networks are prone to several shortcomings, like congestion around the sink nodes and vulnerability to selected attacks or failures of hub nodes. To avoid these critical aspects, it is desirable to design networks having distributed processing and decision capabilities, so that the nodes are able to reach a globally optimal decision without the need to send all the data to a fusion center. Also from a fundamental information theoretic perspective, if the goal of the network is to compute a function of the data which has structural properties, e.g. it is a divisible function, for example, an efficient network design requires some sort of *in-network* or distributed processing. The problem of distributed field estimation has been often considered in the context of *stochastic* models where strong assumptions are made about the statistical description of the physical field to be estimated. In general the observations collected by a sensor network are modeled through Gaussian variables whose statistical dependency structure is described by a Markov random field that is a particular *graphical model*. The success of stochastic methods in the estimation of field values is limited by the appropriateness of the statistical assumptions made by the model; in certain applications such strong modeling assumptions are warranted

and systems designed from these models show promise. However, in other scenarios, prior knowledge is at best vague and translating such knowledge into a statistical model is undesirable. Applications such as these pave the way for a study of distributed estimation based on assuming a *deterministic* model for the underlying field; the model of the field is given by a weighted sum of basis functions and *distributed regression* is implemented.

In this work we provide two different solutions for the problem of the field estimation in a wireless sensor network using a completely distributed approach. In Chapter 1 we propose a distributed projection algorithm that is able to perform a distributed spatial smoothing of the measurements gathered by a sensor networks, characterized by fast convergence properties and resilience against inter-sensor communication noise. In Chapter 2 we faced the problem of the field estimation exploiting a stochastic approach; we generalize the algorithms based on Belief Propagation present in literature solving the particular case of a clustered network where nodes inside the same cluster observe the same field value.

Chapter 1

Distributed projection algorithms

1.1 Introduction

Motivated in part by the success of reproducing kernel methods in machine learning most of works propose a distributed implementation of kernel least-square regression, e.g. see [28], [31] and [30]. In [30] through a relaxation of the problem that is derived from the topology of the sensor network a local-message passing algorithm based on the SOP algorithm (successive orthogonal projection) is proposed. This solution require that each node knows the locations of its neighbors and inverts a matrix whose dimensions are given by the number of node's neighbors. Such technique is however unadapted in practice as sensors are densely deployed, with heavier computational burden and higher neighborhood concentration. This limit is related to the main drawback of the application of classical kernel machine for regression in sensor network, i.e. the order of the resulting model is equal to the number of sensors (observations). In order to overcome this drawback in [31] by exploiting the natural link between reducing the model order and the topology of the network a

reduced-order model approach is proposed. The solution proposed limits the computational burden of each node but requires the establishment of a walk through the network and the transmission of too data in the last steps of this walk. Various algorithms able to solve sparse linear system of equations, similar to those described in the stochastic field estimation methods, could be used for computing the coefficients of the kernel least-square regression. Since the kernel function expresses the similarity between two measurements this sparsity "corresponds" with the topology of the network. Along these lines, [28] developed a distributed algorithm based on Gaussian distributed elimination algorithm executed on a cleverly engineered junction tree.

In the context of deterministic model we propose a novel approach to estimated in a completely distributed way a physical field of values; our approach is based on the assumption that in most cases, the useful signal is a smoothed function, as a result of a diffusion process. However, typically the set of measurements is not all smoothed because of the observation noise. In mathematical terms, the vector of measurements collected by a network composed of N nodes belongs, in general, to a vector space of dimension N. However, the useful signal field typically belongs to a subspace of dimension much smaller than N. Thus, one of the primary goals of a sensor network is to perform a projection of the observed vector onto the useful signal subspace, to eliminate all the noise components lying out of the useful subspace. Projecting data onto a given subspace is a typical signal processing task whose straightforward implementation in a sensor network requires all the nodes to send their data to a sink node (fusion center), which carries out the projection operation. The problem addressed in this work is how to carry out the projection operation through a decentralized network, with no fusion center, using a network where each node exchanges information

only with its neighbors. This problem has been studied extensively in the case where the useful signal is homogeneous, that is spatially constant. In such a case, the so called consensus algorithms are able to provide the globally optimal estimate with a network of only locally interacting sensors, see e.g.[2, 5]. The consensus algorithms are completely decentralized, but they represent an extreme form of smoothing, because they destroy any potential spatial variation in the field of interest, which in most cases is spatially inhomogeneous. Distributed algorithms able to reach globally optimal processing tasks are available and they are typically iterative, see e.g. [1, 2, 3]. However, it is precisely the iterative nature of distributed algorithms that makes them prone to a series of shortcomings, namely *convergence time* and *complexity*, as detailed next: 1) the iterative mechanism needs time to converge and the longer is convergence time, the higher is the energy consumption necessary to reach the final decision with the desired accuracy or reliability; 2) insuring the appropriate exchange of data through a shared medium requires a proper medium access control protocol, that needs to take into account the iteration index; 3) since the interaction among the nodes occurs through realistic channels, the iterative exchange of data involves an iterated addition of channel noise. Since energy and complexity are some of the major concerns in sensor networks, it is clear that using distributed algorithms becomes really attractive only if we are able to limit the energy consumption and complexity of each node. Our goal is to propose a distributed implementation of the projection operation through local exchange of data, with minimum convergence time and robustness against inter-sensor communication noise.

1.2 Decentralized projection algorithms

Let us consider a network composed of N sensors with arbitrary topology that monitors a stationary, inhomogeneous physical field of interest. Denote the measurement collected by the i-th sensor, located at (x_i, y_i), by $g(x_i, y_i) = z(x_i, y_i) + \xi(x_i, y_i)$, where $z(x_i, y_i)$ is the useful field and $\xi(x_i, y_i)$ is the observation error, assumed to be a zero mean random variable with variance $\sigma^2_{\xi_i}$. In vector notation, a quite general observation model is

$$\boldsymbol{g} = \boldsymbol{z} + \boldsymbol{v} = \boldsymbol{U}\boldsymbol{s} + \boldsymbol{\xi}, \tag{1.2.1}$$

where $\boldsymbol{z} = \boldsymbol{U}\boldsymbol{s}$ is the useful signal, $\boldsymbol{U}$ is a $N \times r$ matrix, with $r \leq N$, and $\boldsymbol{s}$ is a $r \times 1$ column vector. The columns of $\boldsymbol{U}$ constitute a basis spanning the useful signal subspace. In many applications, the useful signal is a smooth function. This property can be modeled by choosing the columns of $\boldsymbol{U}$ as the low frequency components of the Fourier basis or low-order polynomials. In practice, the dimension r of the useful signal subspace is typically much smaller than the dimension N of the observation space. Hence, a strong noise reduction may be obtained by projecting the observation vector onto the signal subspace. More specifically, if the noise vector is Gaussian, with zero mean and covariance $\sigma^2_{\xi}\boldsymbol{I}$, the maximum likelihood estimator of $\boldsymbol{z}$ is [15]

$$\hat{\mathbf{z}} = \boldsymbol{U}(\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{U}^T\mathbf{g}. \tag{1.2.2}$$

If the noise pdf is unknown, the estimator (1.2.2) is still significant, as it is the so called Best Linear Unbiased Estimator (BLUE) [15].

The operation performed in (1.2.2) corresponds to the orthogonal projection of the observation vector onto the subspace spanned by the columns of $\boldsymbol{U}$. Assuming, without any loss of generality (w.l.o.g.), the columns of $\boldsymbol{U}$ to be orthonormal, the

projector simplifies into

$$\hat{\boldsymbol{z}} = \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{g}. \tag{1.2.3}$$

Two examples of the application of the previous method are reported in Figures 1.1 and 1.2. In Figure 1.1 we consider the estimation of a 2D field; the useful field is a bi-dimensional sinusoid and the measurement is corrupted by zero mean white Gaussian noise. The number of sensors is 256 and the SNR defined respect to the maximum value of the useful field is 5dB. The observations are represented by the circles and the final state vector is represented by the 2D sinusoid shown in Figure 1.1, which is almost perfectly superimposed on the useful field. In Figure 1.2 we consider the estimation of a one-dimensional field, where the observed field is a sinusoid corrupted by zero mean white Gaussian noise. The number of sensors is 64 and SNR = 0dB. Also in the one-dimensional case, the reconstruction (red line) achieved projecting the observed field onto the low frequency components of the Fourier basis is very close to the useful signal.

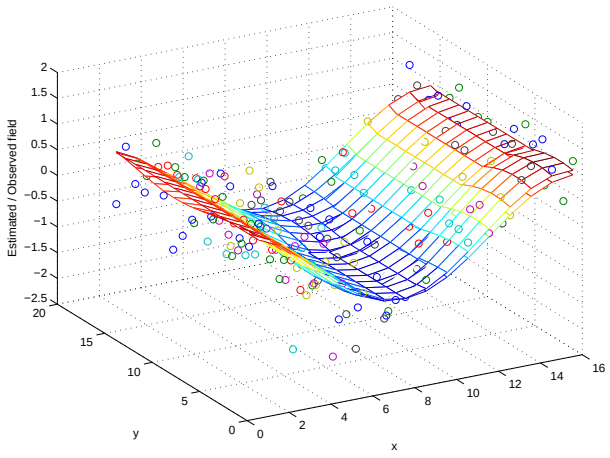

Figure 1.1: Reconstruction of a 2D field.

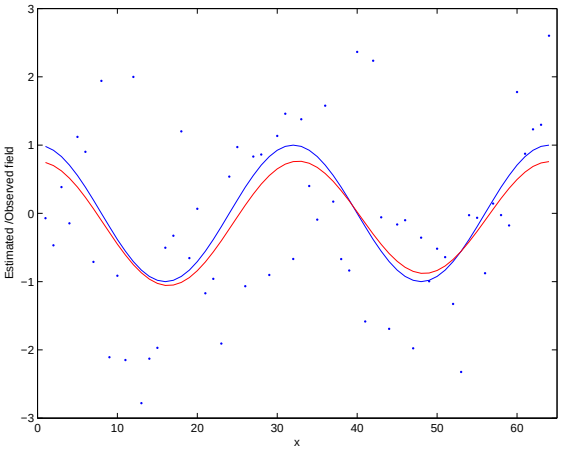

Figure 1.2: Reconstruction of a one-dimensional field.

With a centralized system, the computation of (1.2.3) requires that all nodes send their measurements (vector $\boldsymbol{g}$) to a fusion center that computes (1.2.3). Conversely, our problem is how to compute (1.2.3) with a decentralized network, where each node exchanges information with its neighbors only. We suppose that each sensor is equipped with three basic components:

1. a transducer that senses the physical parameter of interest;

2. a discrete dynamical system whose state is initialized with the local measurements;

3. a radio interface that transmits the state of the dynamical system and receives the state transmitted by the other nodes, thus ensuring the interaction among sensors.

The proposed approach is based on an iterative procedure, where each node initializes a state variable with the local measurement, let us say $z_i[0] = g(x_i, y_i)$, and then it

evolves by interacting with nearby nodes in the following way

$$z_i[k+1] = z_i[k+1] - \epsilon \sum_{j \in \mathcal{N}_i} L_{ij} z_j[k+1], \tag{1.2.4}$$

where $\mathcal{N}_i$ is the set of neighbors of node i. Denoting by $\boldsymbol{z}[k]$, the $N \times 1$ vector containing the states of all nodes at iteration k, the whole system evolves according to the following linear state equation:

$$\mathbf{z}[k+1] = \mathbf{W}\mathbf{z}[k], \qquad k = 0, 1, \ldots, \quad \mathbf{z}[0] = \boldsymbol{g} \in \mathbb{R}^N, \tag{1.2.5}$$

where $\mathbf{W} =$

-$\epsilon\mathbf{L} \in \mathbb{R}^{N \times N}$ is a *sparse* (not necessarily symmetric) matrix. We assume here that $\boldsymbol{W}$ does not vary with time. The sparsity of $\boldsymbol{W}$ is what characterizes the network topology and, in particular, it models the interaction of each node with its neighbors only. Given the interaction mechanism (1.2.5), our problem is twofold: 1) guarantee that system (1.2.5) converges to the desired vector (1.2.3), although using a sparse matrix $\boldsymbol{W}$; 2) find the sparse matrix $\boldsymbol{W}$, under a topological constraint, so that the convergence time is minimized. Let us denote by $\mathbf{P}_{\mathcal{R}(\mathbf{U})} \in \mathbb{R}^{N \times N}$ the orthogonal projector onto the r-dimensional subspace of $\mathbb{R}^N$ spanned by the columns of $\mathcal{R}(\mathbf{U})$, where $\mathcal{R}(\cdot)$ denotes the range space operator and $\mathbf{U} \in \mathbb{R}^{N \times r}$ is a full-column rank matrix, assumed, w.l.o.g., to be semi-unitary. System (1.2.5) converges to the desired orthogonal projection of the initial value vector $\mathbf{z}[0] = \boldsymbol{g}$ onto $\mathcal{R}(\mathbf{U})$, for *any* given $\mathbf{g} \in \mathbb{R}^N$, if and only if

$$\lim_{k \to +\infty} \boldsymbol{z}[k] = \lim_{k \to +\infty} \boldsymbol{W}^k \boldsymbol{g} = \mathbf{P}_{\mathcal{R}(\mathbf{U})} \boldsymbol{g}, \tag{1.2.6}$$

i.e.,

$$\lim_{k \to +\infty} \mathbf{W}^k = \mathbf{P}_{\mathcal{R}(\mathbf{U})}. \tag{1.2.7}$$

Necessary and sufficient conditions for (1.2.7) were proved in [11] and are given in the following. Given the dynamical system in (1.2.5) and the projection matrix $\mathbf{P}_{\mathcal{R}(\mathbf{U})}$, the vector $\mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[0]$ is globally asymptotically stable for *any fixed* $\mathbf{z}[0] \in \mathbb{R}^N$, if and only if the following conditions are satisfied:

$$\mathbf{W}\,\mathbf{P}_{\mathcal{R}(\mathbf{U})} = \mathbf{P}_{\mathcal{R}(\mathbf{U})} \tag{C.1}$$

$$\mathbf{P}_{\mathcal{R}(\mathbf{U})}\,\mathbf{W} = \mathbf{P}_{\mathcal{R}(\mathbf{U})} \tag{C.2}$$

$$\rho\left(\mathbf{W} - \mathbf{P}_{\mathcal{R}(\mathbf{U})}\right) < 1 \tag{C.3}$$

where $\rho(\cdot)$ denotes the spectral radius operator [12]. Under (C.1)-(C.3), the error vector $\mathbf{e}[k] \triangleq \mathbf{z}[k] - \mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[0]$ satisfies the following dynamics:

$$\mathbf{e}[k+1] = \left(\mathbf{W} - \mathbf{P}_{\mathcal{R}(\mathbf{U})}\right)\mathbf{e}[k], \qquad k = 0, 1, \ldots. \tag{1.2.8}$$

□

Remark 1−Interpretation of necessary and sufficient conditions: Interestingly, conditions C.1-C.3 have an intuitive interpretation, as described next. C.1 and C.2 state that, if system in (1.2.5) asymptotically converges, then it is guaranteed to converge to the desired value. In fact, C.1 guarantees that the projection of vector $\mathbf{z}[k]$ onto $\mathcal{R}(\mathbf{U})$ is an invariant quantity for the dynamical system, implying that the system in (1.2.5), during its evolution, keeps the component $\mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[0]$ of $\mathbf{z}[0]$ unaltered; whereas C.2 makes $\mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[0]$ a fixed point of matrix $\mathbf{W}$ and thus a potential accumulation point for the sequence $\{\mathbf{z}[k]\}_k$. However, both conditions C.1 and C.2 do not state anything about the convergence of the dynamical system; which is instead guaranteed by C.3, imposing that all the modes associated to the eigenvectors orthogonal to $\mathcal{R}(\mathbf{U})$ be asymptotically vanishing [cf. (1.2.8)].

Remark 2−Special cases: Observe that, as special case, our conditions C.1-C.3 contain the well-known convergence conditions of linear discrete-time dynamical systems toward the (weighted) average consensus (see, e.g., [1, 2]). It is sufficient to set in (1.2.6), $r = 1$ and $\mathbf{U} = \mathbf{u} = \frac{1}{\sqrt{N}}\mathbf{1}_N$, where $\mathbf{1}_N$ is the N-length vector of all ones. In such a case, C.1-C.3 can be restated as following: the digraph associated to the network described by $\mathbf{W}$ must be strongly connected and balanced.

Remark 3: Interestingly, conditions C.1-C.3 can be restated in terms of semistability properties of matrix $\mathbf{W}$, as detailed next. Denoting with OUD the Open Unit Disk, i.e. the set $\{x \in \mathcal{C} :\mid x \mid< 1\}$, a matrix $\mathbf{W}$ is *semistable* if its spectrum spec($\mathbf{W}$) satisfies spec($\mathbf{W}$) $\subset OUD \cup \{1\}$ and, if $1 \in$ spec($\mathbf{W}$), then 1 is semisimple, i.e. its algebraic and geometric multiplies coincide. If $\mathbf{W}$ is semistable, then

$$\lim_{k\to+\infty} \mathbf{W}^k = \mathbf{I} - (\mathbf{I} - \mathbf{W})^{\dagger}(\mathbf{I} - \mathbf{W}) = \mathbf{I} - \mathbf{L}^{\dagger}\mathbf{L} \tag{1.2.9}$$

where $\dagger$ denotes group generalized inverse (or Moore-Penrose inverse) and we have used the relation $\mathbf{W} = \mathbf{I} - \epsilon\mathbf{L}$. But $\mathbf{I} - \mathbf{L}^{\dagger}\mathbf{L}$ is the projector onto the null-space of $\mathbf{L}$. Hence, necessary and sufficient conditions ensuring convergence are that the columns of $\mathbf{U}$ span the null-space of $\mathbf{L}$ and that $\mathbf{W}$ is semistable; which corresponds to C.1-C.3.

Our goal is to derive the optimal choice of matrix $\mathbf{W}$, consistent with the network topology constraints, that maximizes the convergence speed of dynamical system (1.2.5), while guaranteeing the convergence of (1.2.5) to the desired final vector $\mathbf{z}^{\star} = \mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{g}$. As we will show in the next section, this optimization leads to the minimization of the matrix spectral radius. Unfortunately, minimizing the spectral radius of a non-symmetric matrix is a notoriously difficult problem, intractable except for small-medium values of the dimensions [3]. Some optimization problems involving

the minimization of the spectral radius were indeed shown to be NP-hard [17, 18]. Since in typical sensor network problems, the dimension of $\boldsymbol{W}$ may be quite large and, furthermore, the advantage of using a non-symmetric matrix as opposed to a symmetric one is unclear, we will primarily restrict our search to the class of symmetric matrices $\mathbf{W}$, satisfying conditions C.1-C.3. Therefore, we consider matrices $\mathbf{W}$ having the following structure:

$$\mathbf{W} = \mathbf{I} - \epsilon\mathbf{L}, \qquad \epsilon \in \mathbb{R}, \qquad \mathbf{L} = \mathbf{L}^T, \qquad \mathbf{L} \succeq \mathbf{0}. \tag{1.2.10}$$

Under this position, conditions C.1-C.2 can be expressed in terms of $\boldsymbol{L}$ as

$$\mathbf{U}^T\mathbf{L} = \mathbf{0}. \tag{1.2.11}$$

Introducing the semi-unitary matrix $\mathbf{U}^\perp \in \mathbb{R}^{N-r\times N-r}$ such that $\mathbf{U}^T\mathbf{U}^\perp = \mathbf{0}$, in [11] the authors proved that condition (1.2.11), together with the symmetry of $\boldsymbol{L}$, leads to the following structure for $\mathbf{L}$:

$$\mathbf{L} = \mathbf{U}^\perp\overline{\mathbf{L}}\mathbf{U}^{\perp^T}, \tag{1.2.12}$$

with $\overline{\mathbf{L}} \in \mathbb{R}^{(N-r)\times(N-r)}$ satisfying $\overline{\mathbf{L}} = \overline{\mathbf{L}}^T$ and $\overline{\mathbf{L}} \succeq \mathbf{0}$. This condition states that every feasible $\mathbf{L}$ must belong to the range space of $\mathbf{U}^\perp \in \mathbb{R}^{N-r\times N-r}$. Similarly, condition C.3 becomes [11]

$$\rho\left(\mathbf{I} - \epsilon\mathbf{L} - \mathbf{P}_{\mathcal{R}(\mathbf{U})}\right) = \rho\left(\overline{\mathbf{I}} - \epsilon\overline{\mathbf{L}}\right) < 1. \tag{1.2.13}$$

where $\overline{\mathbf{I}}$ denotes the $N-r$ identity matrix. Using (1.2.12), condition (1.2.13) is equivalent to the following:

$$\epsilon\lambda_i(\overline{\mathbf{L}}) > 0 \qquad \text{and} \qquad \epsilon\lambda_i(\overline{\mathbf{L}}) < 2, \quad \forall i \in \{1, \ldots, N-r\}, \tag{1.2.14}$$

where $\{\lambda_i(\overline{\mathbf{L}})\}_{i=1}^{N-r}$ denotes the set of eigenvalues of $\overline{\mathbf{L}}$. Since $\overline{\mathbf{L}} \succeq \mathbf{0}$, (1.2.14) can be rewritten as:

$$\overline{\mathbf{L}} \succ \mathbf{0} \qquad \text{and} \qquad 0 < \epsilon < \frac{2}{\lambda_i(\overline{\mathbf{L}})}, \quad \forall i \in \{1, \ldots, N-r\}. \tag{1.2.15}$$

In words, the search for a matrix $\boldsymbol{W}$ satisfying (1.2.7) is equivalent to searching for a matrix $\boldsymbol{L}$ whose kernel space coincides with the useful signal subspace. The challenging question in our sensor network context is to find whether (1.2.7) can be satisfied using a *sparse* matrix. Before tackling the general case, in the following we consider a few special cases where there do exist sparse matrices satisfying (1.2.7).

1.2.1 Distributed polynomial approximation

In most applications, the useful field $z(x, y)$ is a continuous function of the spatial coordinates and then, according to the Weierstrass' theorem, it can be approximated by a two-dimensional polynomial of finite order in the variables x and y, with an arbitrarily small error. For simplicity, we assume that the nodes are uniformly spaced over a 2-dimensional grid and that the observed field does not vary with time. Let us denote with $K-1$ the order of the polynomial in both variables x and y. In such a case, given the observations (1.2.1), we may perform a spatial soothing of the observation by finding the vector $\hat{\boldsymbol{z}}$ that minimizes the following functional, as proposed in [8]:

$$V(\hat{\boldsymbol{z}}) = \frac{1}{2} \sum_{i \in \mathcal{N}} \sum_{m=0}^{K} [\nabla_x^{(K-m)} \nabla_y^{(m)} \hat{\boldsymbol{z}}(x_i, y_i)]^2, \tag{1.2.16}$$

where $\nabla_x^{(m)}$ and $\nabla_y^{(m)}$ denote the m-th order difference operator with respect to the variables x and y, respectively. More specifically, the operator is defined through the

following properties:

$$\begin{aligned}
&\nabla_x^{(0)}\hat{z}(x_i,y_i) = \hat{z}(x_i,y_i);\ \nabla_y^{(0)}\hat{z}(x_i,y_i) = \hat{z}(x_i,y_i);\\
&\nabla_x^{(1)}\hat{z}(x_i,y_i) = \hat{z}(x_i,y_i) - \hat{z}(x_{i-1},y_i);\\
&\nabla_y^{(1)}\hat{z}(x_i,y_i) = \hat{z}(x_i,y_i) - \hat{z}(x_i,y_{i-1});\\
&\nabla_x^{(n)}\hat{z}(x_i,y_i) = \nabla_x^{(1)}\left[\nabla_x^{(n-1)}\hat{z}(x_i,y_i)\right].
\end{aligned} \tag{1.2.17}$$

To take into account border effects, $\mathcal{N}$ is the set of indices for which the above differences can be properly computed; also recall that we have assumed a uniform 2D grid for simplicity of exposition. The cost function (1.2.16) is a quadratic form on $\hat{\boldsymbol{z}}$, which can be written as $J(\boldsymbol{z}) = \boldsymbol{z}^T\boldsymbol{L}\boldsymbol{z}$. The minimum of (1.2.16) can then be reached using the steepest descent method

$$\boldsymbol{z}[k+1] = \boldsymbol{z}[k] - \epsilon\boldsymbol{L}\boldsymbol{z}[k] \triangleq \boldsymbol{W}\boldsymbol{z}[k], \tag{1.2.18}$$

with initialization $\boldsymbol{z}[0] = \boldsymbol{g}$, having set $\boldsymbol{W} \triangleq \boldsymbol{I} - \epsilon\boldsymbol{L}$. It is useful to remark that $\boldsymbol{L}$ is, by construction, a positive semidefinite, symmetric, *sparse* matrix. More specifically, the sparsity of $\boldsymbol{L}$ depends on the maximum degree of the approximating polynomials[8]. Hence, there exist a unitary matrix $\boldsymbol{U}$ and a diagonal matrix Λ such that:

$$\mathbf{L} = \begin{pmatrix} \mathbf{U}^{\perp} & \mathbf{U} \end{pmatrix} \begin{pmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}^{\perp H} \\ \mathbf{U}^{H} \end{pmatrix} \tag{1.2.19}$$

where $\mathbf{U}$ has dimension $N \times L$, with L denoting the dimension of the kernel of $\boldsymbol{L}$. The columns of $\mathbf{U}$ are the vectors spanning the kernel of $\boldsymbol{L}$ that, because of the structure of (1.2.16), is spanned by the polynomials of orders up to $K-1$. Let $\lambda_i(\mathbf{L})$ and $\lambda_i(\mathbf{W})$ denote the eigenvalues of $\boldsymbol{L}$ and $\boldsymbol{W}$; we assume that these eigenvalues are ordered in non-decreasing order. We can always choose ε, so that the eigenvalues of $\mathbf{W}$ satisfy $0 < |\lambda_i(\mathbf{W})| < 1$, $\forall 1$. This property is clearly achieved by setting

$$0 < \varepsilon < \frac{2}{\lambda_N(\mathbf{L})}. \tag{1.2.20}$$

With this choice, it is straightforward to verify that

$$\lim_{k\to\infty} \mathbf{z}(k) = \lim_{k\to\infty} \sum_{i=1}^{N} \lambda_i^k(\mathbf{W})\mathbf{u}_i\mathbf{u}_i^*\mathbf{z}(0) = \mathbf{U}\mathbf{U}^H\mathbf{z}(0), \tag{1.2.21}$$

where the columns of $\boldsymbol{U}$ are exactly the vectors spanning the kernel of $\boldsymbol{L}$. Hence, expression (1.2.21) states that the *final value coincides with the projection of the observation vector onto the nullspace of* $\boldsymbol{L}$. Hence, (1.2.18) is an example of distributed orthogonal projector onto the signal subspace spanned by low order polynomials using only local interactions. If the sensors are uniformly spaced over a line, each node (except the border nodes) interacts with a number of neighbors which is equal to K, if $K-1$ is the maximum polynomial degree, for example:

a) $K = 1$,

$$\mathbf{L} = \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 1 \end{pmatrix}$$

In this case, we have $L_{ij} = 1$, if $|i-j| = 1$, and 0 otherwise. This happens when each node has only two neighbors (except the border nodes having only one neighbor). The matrix $\boldsymbol{L}$ has, in this case, a null eigenvalue of multiplicity one. Since each row of $\boldsymbol{L}$ has zero row sum, the eigenvector associated with the null eigenvalue of $\boldsymbol{L}$ is the vector $\mathbf{1}$ composed of all ones. Hence, the final result is the conventional average consensus algorithm.

b) $K = 2$,

$$\mathbf{L} = \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & 0 & \dots & 0 \\ -2 & 5 & -4 & \ddots & \ddots & \ddots & \ddots & 0 \\ 1 & -4 & 6 & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 6 & -4 & 1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & -4 & 5 & -2 \\ 0 & \dots & \dots & \dots & 0 & 1 & -2 & 1 \end{pmatrix}$$

In this case, the nullspace of $\boldsymbol{L}$ has dimensionality two and it is spanned by a linear combinations of polynomials of degree zero and one. An orthonormal set is given, in this case, by the Legendre polynomials of degree zero and one. Hence the final vector is a straight line. Since any continuous function can be approximated with an arbitrarily small error, by a polynomial, the above method provides then a distributed tool to approximate any continuous field of values. The well known average consensus algorithm [2] is strictly related to the above algorithm, as it is the result of the minimization of the disagreement function

$$J(\boldsymbol{z}) = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij}(z_i - z_j)^2, \tag{1.2.22}$$

with $a_{ij} = a_{ji}$ nonnegative real coefficients. In this case, we can still write (1.2.22) as a quadratic form $\boldsymbol{z}^T \boldsymbol{L} \boldsymbol{z}$, where the matrix $\boldsymbol{L}$ has coefficients $L_{ii} = \sum_{j=1}^{N} a_{ij}$ and $L_{ij} = -a_{ij}$, with $i \neq j$. The minimum of (1.2.22) can still be reached using a steepest descent algorithm, as in (1.2.18). The matrix $\boldsymbol{L}$ has, by construction, an eigenvector $\mathbf{1}_N$ composed by all ones, associated to a null eigenvalue. If the graph describing the interaction among the nodes is strongly connected and balanced, the multiplicity of the null eigenvalue is one and the asymptotic value of (1.2.18) is thus the orthogonal

projection onto the kernel of $\boldsymbol{L}$, i.e.,

$$\lim_{k\to\infty} \mathbf{z}[k] = \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\boldsymbol{g}. \tag{1.2.23}$$

This is clearly a particular case of (1.2.7), corresponding to a signal subspace spanned by the vector $\mathbf{1}_N$. What is important to remark here is that global consensus requires only the connectivity of the network. This can be achieved even if every node is connected with only one neighbor, i.e. with a very sparse matrix $\boldsymbol{L}$. An example of the application of the previous method to a 2D field is reported in Figure 1.3, where the useful field is a paraboloid and the measurement is corrupted by zero mean white Gaussian noise. The number of sensors is 144 and the SNR defined respect to the maximum value of the useful field is equal to 10dB. The observations are represented, in Figure 1.3, by the circles. The final state vector is represented by the paraboloid shown in Figure 1.3, which is almost perfectly superimposed on the useful field.

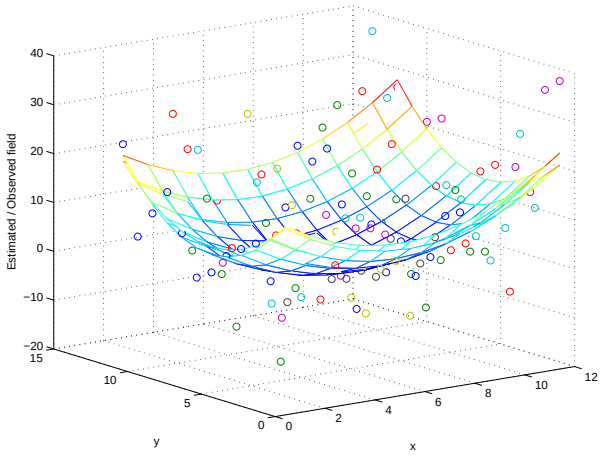

Figure 1.3: Reconstruction of a 2D field.

In principle, we can improve the approximation in the above method by increasing the value of K. However, the degree of the network, defined as the maximum number of neighbors of each node, increases with K. This induces a greater waste of energy. Hence, it is necessary to find the right trade-off between energy consumption and approximation error. Furthermore, if the useful field presents discontinuities, we could be forced to use very high values of K and still have a nonnegligible error. It is then useful to devise some variants of the previous method that still allow us to have a limited value of the network degree. To this end, we use the following cost function

$$J(\hat{\boldsymbol{z}}) = \mu\, V(\hat{\boldsymbol{z}}) + (1-\mu)\|\hat{\boldsymbol{z}} - \boldsymbol{g}\|^2, \tag{1.2.24}$$

with $V(\hat{\boldsymbol{z}})$ given in (1.2.16)and $\mu \in (0,1)$. In this case, the steepest descent method leads to

$$\mathbf{z}(k+1) = [(1-\varepsilon(1-\mu))\boldsymbol{I} - \varepsilon\mu\,\mathbf{L}\,]\boldsymbol{z}(k) + \varepsilon(1-\mu)\,\mathbf{z}(0), \tag{1.2.25}$$

with $\boldsymbol{z}(0) = \boldsymbol{g}$. Introducing the matrix $\boldsymbol{W} \triangleq (1-\varepsilon(1-\mu))\boldsymbol{I} - \varepsilon\mu\mathbf{L}$, we can guarantee that the eigenvalues $\lambda_i(\boldsymbol{W})$ of $\boldsymbol{W}$ are strictly between -1 and 1, by setting

$$0 < \varepsilon < \min_i \left\{ \frac{2}{\lambda_i(\boldsymbol{L}) + 1 - \mu} \right\} = \frac{2}{\lambda_N(\boldsymbol{L}) + 1 - \mu}. \tag{1.2.26}$$

With a few simple algebraic manipulations, we can rewrite $\boldsymbol{z}(k)$ as

$$\begin{array}{rcl} \boldsymbol{z}(k) & = & \boldsymbol{W}^k\mathbf{z}(0) + \varepsilon(1-\mu)\sum_{n=0}^{k-1}\mathbf{W}^k\boldsymbol{z}(0) \\ & = & \boldsymbol{W}^k\mathbf{z}(0) + \varepsilon(1-\mu)(\mathbf{I}-\mathbf{W})^{-1}(\mathbf{I}-\mathbf{W}^k)\boldsymbol{z}(0). \end{array} \tag{1.2.27}$$

Choosing ε according to (1.2.26), the state vector converges to

$$\lim_{k\to\infty} \mathbf{z}(k) = \left(\boldsymbol{I} + \frac{\mu}{1-\mu}\,\boldsymbol{L}\right)^{-1}\boldsymbol{g}. \tag{1.2.28}$$

$$= [\boldsymbol{U}\boldsymbol{U}^T + (1-\mu)\mathbf{U}^{\perp}[(1-\mu)\boldsymbol{I}_{N-k} - \mu\Lambda_{\boldsymbol{L}_1}]^{-1}\mathbf{U}^{\perp T}]\boldsymbol{g} \tag{1.2.29}$$

If $\mu = 1$ the final vector coincides with the projection of the observation onto the nullspace of the matrix $\boldsymbol{L}$; in general, for $0 < \mu < 1$, besides this projection, there is also a vector component that lies in the orthogonal subspace.Depending on the value of μ, we may give different relative importance to smoothing or fidelity to the original observation. In the extreme case of $\mu = 0$, the network does not apply any smoothing, i.e. $\lim_{k\to\infty} \mathbf{z}(k) = \boldsymbol{g}$, whereas, at the other extreme, when $\mu = 1$, the final value coincides with the projection of the observation onto the nullspace of the Laplacian matrix, as proved in the previous section. A numerical example is reported in Fig.1.4, relative to a one-dimensional network located over a straight line. The observed signal in this case is a sinusoid (dashed line) and the observation (dots) is corrupted by white Gaussian noise. The SNR is 5 dB. Smoothing has been performed using the simple algorithm (1.2.18), with $K = 3$. In this case, with $\mu = 1$, the method projects the observed vector onto the space spanned by second order polynomials. Since the observation is a noisy sinusoid, the final result (dash-dotted line) is not very good. However, as soon as μ is slightly less than one, the method is forced to take into account the fidelity to the observation, and the final result is much better than in the previous case. Using $\mu = 0.9999$, for example, the result of the smoothing operation is represented by the solid line and we can see that the approximation is now pretty good.

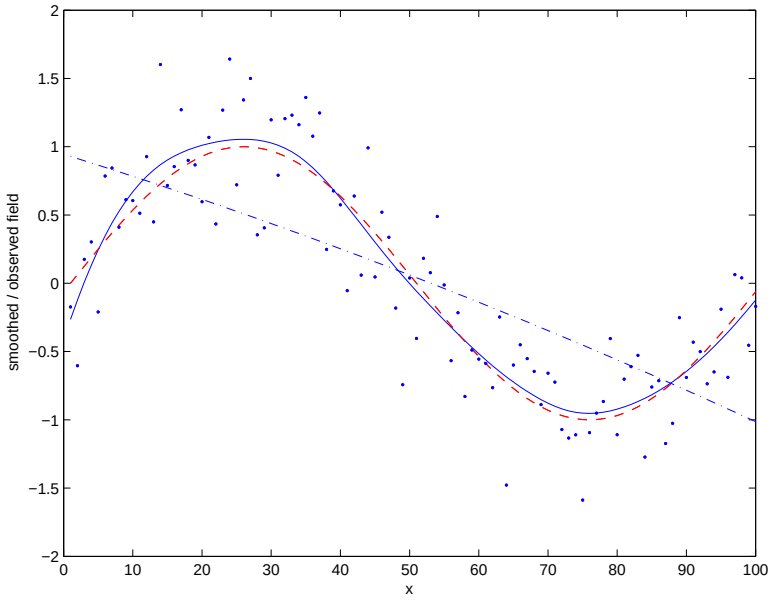

Figure 1.4: Reconstruction of a noisy sinusoid.

Even if this is only a simple example, Figure 1.4 suggests that the choice of μ can have a strong impact on the smoothing operation. To quantify the final distortion, we can compute the mean square error, averaged over the noise realizations. Introducing the matrix $\boldsymbol{P}(\mu) = \left(\boldsymbol{I} + \frac{\mu}{1-\mu}\boldsymbol{L}\right)^{-1}$, the final MSE is

$$MSE(\mu) = \|(\boldsymbol{P}(\mu) - \boldsymbol{I})\boldsymbol{f}\|^2 + \sigma_n^2 \mathrm{tr}(\boldsymbol{P}(\mu)\boldsymbol{P}(\mu)^T). \tag{1.2.30}$$

In the case of a sinusoidal function, this function, normalized to $\|\boldsymbol{f}\|^2$, is reported in Figure 1.5, for different values of μ and σ_n^2. As expected, there is an optimal value of μ that depends on the noise level: When there is no noise, it is better to apply no smoothing at all, and thus the best value of μ is zero; conversely, as the noise increases, it is better to use values of μ closer and closer to one.

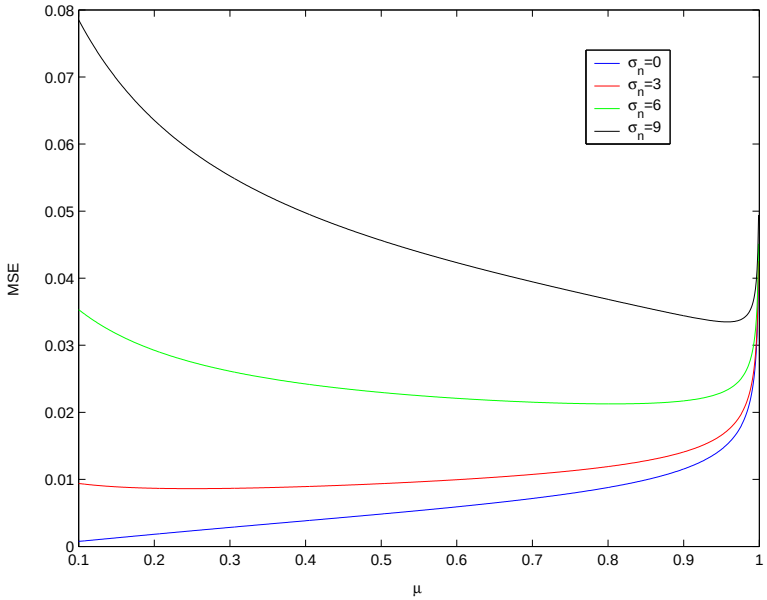

Figure 1.5: MSE as a function of μ and σ_n.

1.3 Maximum convergence rate under topology constraints

We can now focus on the optimal choice of matrix $\boldsymbol{W}$ that allows the asymptotic convergence onto the desired signal subspace [see (1.2.6)], still having a sparse structure, in the general case where the signal subspace is not necessarily spanned by low order polynomials and the sensors do not necessarily lie over a uniform grid. Energy consumption is one of the most critical aspects of wireless sensor networks. From this point of view, iterative algorithms are especially critical because of the iterative exchange of data. It is then fundamental to minimize the time necessary for the iterative algorithms to converge. The energy spent to reach the global projection with the desired accuracy or reliability is the product between the transmit power and the convergence time. For any given spatial distribution of the sensors, the transmit power

constraint reflects into a topology constraint, so that each node interacts only with the nodes lying within its coverage radius. According to these motivations, under conditions C.1-C.3, we focus on the design of the matrix $\mathbf{W}$, consistent with the network topology constraints, that minimizes the convergence rate of system in (1.2.5). We first introduce two different definitions of the convergence rate, then we prove that they give the same results in the case of symmetric matrix $\mathbf{W}$. The convergence rate can be either measured for the worst possible initial vector $\mathbf{z}[0]$ or on the average. In this work, we focus on the former approach. The convergence speed based on the worst possible initial vector $\mathbf{z}[0]$ is measured introducing the asymptotic convergence exponent, as detailed next. Denoting by $\mathbf{z}^\star = \mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[0]$ the final value toward which vector $\mathbf{z}[k]$ converges, under conditions C.1-C.3 the asymptotic convergence exponent for the worst-case convergence rate is given by [22, 23]

$$d(\mathbf{W}) = \sup_{z[0]\neq\mathbf{z}^\star} \lim_{k\to\infty} \frac{1}{k} \ln\left(\frac{\|\mathbf{z}[k]-\mathbf{z}^\star\|}{\|\mathbf{z}[0]-\mathbf{z}^\star\|}\right). \tag{1.3.1}$$

In definition (1.3.1) the distance at the k-th iteration between $\mathbf{z}[k]$ and $\mathbf{z}^\star$ measured by some vector norm $\|\mathbf{z}[k]-\mathbf{z}^\star\|$ is compared with the initial distance $\|\mathbf{z}[0]-\mathbf{z}^\star\|$. Since for large k

$$\|\mathbf{z}[k]-\mathbf{z}^\star\| \simeq Ce^{d(\mathbf{W})k}, \tag{1.3.2}$$

where C is a constant that depends on the initial conditions, $d(\mathbf{W})$ gives the *convergence time*

$$\tau(\mathbf{W}) = \frac{1}{\ln(1/d(\mathbf{W}))}. \tag{1.3.3}$$

The convergence time is the asymptotic number of iterations for the error to decrease by the factor $1/e$ for the worst possible initial vector. Another measure of the

convergence speed is the *per-step* convergence rate, defined as:

$$r_{\text{step}}(\mathbf{W}) = \sup_{z[k]\neq \mathbf{z}^\star} \frac{\|\mathbf{z}[k+1]-\mathbf{z}^\star\|}{\|\mathbf{z}[k]-\mathbf{z}^\star\|}, \tag{1.3.4}$$

which amounts to the worst-case one-step relative reduction. In the following we will consider both the above measurements, using in the definitions (1.3.1) and (1.3.4) the Euclidean norm.

Given the dynamical system (1.2.5), it follows from (1.2.8) and [22, Theorem 3.4] that

$$d(\mathbf{W}) = \rho(\mathbf{I}-\mathbf{W}) \qquad \text{and} \qquad r_{\text{step}}(\mathbf{W}) = \|\mathbf{I}-\mathbf{W}\|_2\,, \tag{1.3.5}$$

where $\rho(\mathbf{I}-\mathbf{W})$ denotes the spectral norm of matrix $\mathbf{I}-\mathbf{W}$ [12].From (1.3.5) we have that if the network topology leads to symmetric matrix $\mathbf{W}$, then $d(\mathbf{W})$ and $r_{\text{step}}(\mathbf{W})$ coincide. Since $\mathbf{W}$ in (1.2.10) is symmetric by construction, without loss of generality we can consider only the asymptotic convergence exponent in (1.3.5). Hence, the minimization of the convergence time in (1.3.3) while guaranteeing the convergence of the system to the desired final vector in (1.2.6), is equivalent to the minimization of $\rho(\mathbf{I}-\mathbf{W})$, under C.1-C.3 [or equivalently under (1.2.12)- (1.2.15)]. The existence of a solution requires that two conditions are satisfied:

1. the network must be connected, i.e. there exist a link between every pair of nodes, possibly composed of multiple hops;

2. the degree (number of neighbors) of each node is not smaller than the dimension of the signal subspace.

The second conditions means, in the context of this work, that the transmit power of each node must be sufficient to reach a number of nodes equal to the dimension of the

signal subspace. This means, for instance, that if we project onto a subspace spanned by a constant vector, as in average consensus algorithms, the signal subspace is one dimensional and then it is only necessary that each node has at least one neighbor. However, if we wish to project onto higher order subspace, the number of neighbors must increase consequently. From (1.2.12), the constraint imposing that each node interacts only with a set of neighbors can be formulated by setting the appropriate values of $\boldsymbol{W}$, and then $\boldsymbol{L}$, equal to 0, i.e., [see (1.2.12)]

$$[\mathbf{L}]_{ij} = \left[\mathbf{U}^{\perp}\overline{\mathbf{L}}\mathbf{U}^{\perp\mathbf{T}}\right]_{ij} = 0 \quad \forall i,j \in \mathcal{B}. \tag{1.3.6}$$

We are now ready to formulate our optimization problem, as given next

$$\begin{aligned} \text{minimize} \quad & \rho\left(\overline{\mathbf{I}} - \epsilon\overline{\mathbf{L}}\right) \\ & \overline{\mathbf{L}},\, \epsilon \\ \text{subject to} \quad & \overline{\mathbf{L}} \succ \mathbf{0}, \quad \overline{\mathbf{L}} = \overline{\mathbf{L}}^{T}, \\ & \left[\mathbf{U}^{\perp}\overline{\mathbf{L}}\mathbf{U}^{\perp\mathbf{T}}\right]_{ij} = 0 \quad \forall i,j \in \mathcal{B}, \\ & 0 < \epsilon < \tfrac{2}{\lambda_i(\overline{\mathbf{L}})}. \end{aligned} \tag{1.3.7}$$

The optimization problem (1.3.7) is not convex and might not be feasible. Assuming that the network topology constraints are such that the feasible set of (1.3.7) is nonempty we prove (see Appendix B) that an optimal solution to (1.3.7) can be

efficiently computed rewriting (1.3.7) as the following semi-definite programming

$$\begin{aligned} &\underset{\widetilde{\mathbf{L}},\,\gamma,\,\widetilde{\mu}}{\text{minimize}} \quad \gamma \\ &\text{subject to} \quad \begin{bmatrix} \widetilde{\mathbf{L}}-\bar{\mathbf{I}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \gamma\bar{\mathbf{I}}-\widetilde{\mathbf{L}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widetilde{\mu}\bar{\mathbf{I}} \end{bmatrix} \succeq \mathbf{0}, \\ &\qquad\qquad \widetilde{\mathbf{L}} = \widetilde{\mathbf{L}}^T, \\ &\qquad\qquad \left[\mathbf{U}^{\perp}\widetilde{\mathbf{L}}\mathbf{U}^{\perp\mathbf{T}}\right]_{ij} = 0 \quad \forall i,j \in \mathcal{B}. \end{aligned} \tag{1.3.8}$$

Once an optimal solution $\left(\widetilde{\mathbf{L}}^{\star},\,\gamma^{\star},\,\widetilde{\mu}^{\star}\right)$ to (1.9.11) is computed, the optimal original $\mathbf{L}^{\star}$ can be obtained through (1.9.9) and (1.2.12): $\mathbf{L}^{\star} = \widetilde{\mu}^{\star-1}\mathbf{U}^{\perp}\widetilde{\mathbf{L}}^{\star}\mathbf{U}^{\perp^T}$. It is important to remark that the maximization of the convergence rate as formulated in (1.3.7) differs from the approaches proposed in the literature to accelerate classical *consensus* algorithms (see, e.g., [4, 7]), since we solve a much more general problem than consensus and, in our formulation, we consider the *joint* optimization of the step size ϵ and the weight matrix $\boldsymbol{L}$, including also sparsity constraints on $\boldsymbol{W}$.

Figures 1.6, 1.7, 1.8 and 1.9 show the minimum convergence time obtained for a network of 25 and 16 sensors uniformly spaced over a line segment of length D, as a function of the number of neighbors. In Figures 1.6 and 1.8 the useful signal is modeled as the summation of the Fourier basis $\{1, \cos(2\pi mx/D), \sin(2\pi mx/D)\}$, with $m = 1, 2, \ldots$, while in Figures 1.7 and 1.9 as the summation of the low order polynomial basis. As expected, as the number of neighbors increases, the convergence time decreases. However, this entails a greater transmit power to cover a larger area. On the other hand, the convergence time increases if, for a given number of neighbors, the dimension of the kernel space (number of Fourier components or the polynomial order) increases. All these considerations hold also for the 2D case; in Figure 1.10

and 1.11 we report the minimum convergence time as a function of the number of neighbors for different dimensions of the polynomial and Fourier kernel in the case of 36 sensors uniformly spaced over a 2D grid. Figure 1.12 compares the minimum convergence time vs. number of neighbors for Fourier and polynomial basis and different kernel dimensions in the case of sensors uniformly spaced on a line.

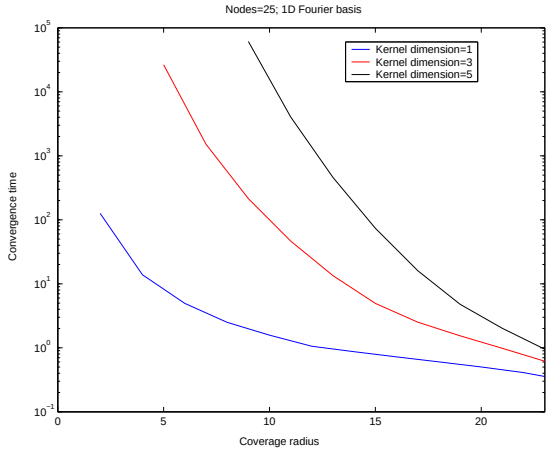

Figure 1.6: Minimum convergence time vs. number of neighbors for a Fourier basis and 25 sensors.

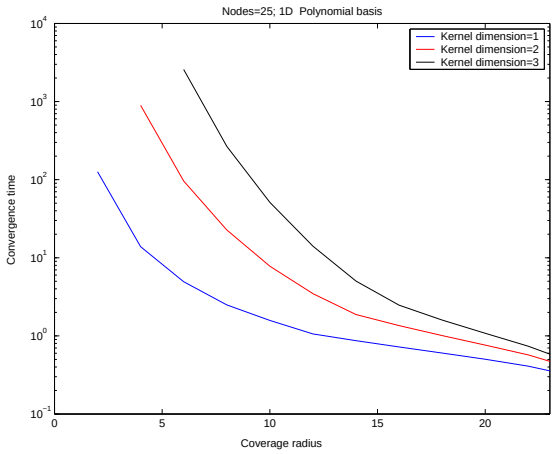

Figure 1.7: Minimum convergence time vs. number of neighbors for a polynomial basis and 25 sensors.

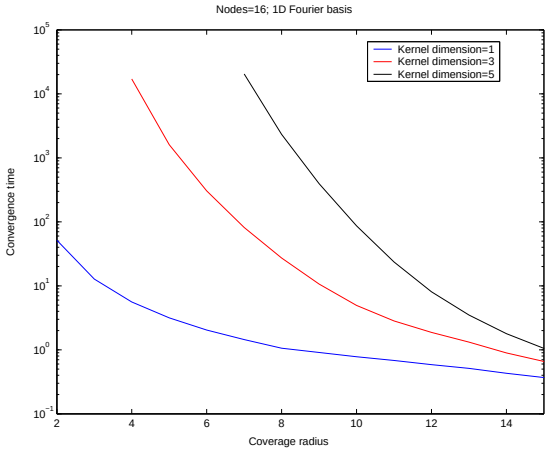

Figure 1.8: Minimum convergence time vs. number of neighbors for a Fourier basis and 16 sensors.

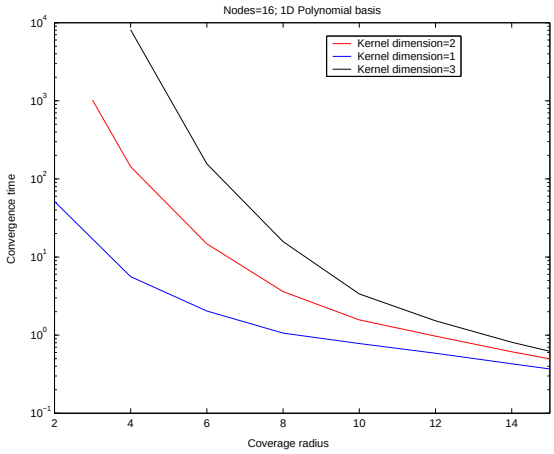

Figure 1.9: Minimum convergence time vs. number of neighbors for a polynomial basis and 16 sensors.

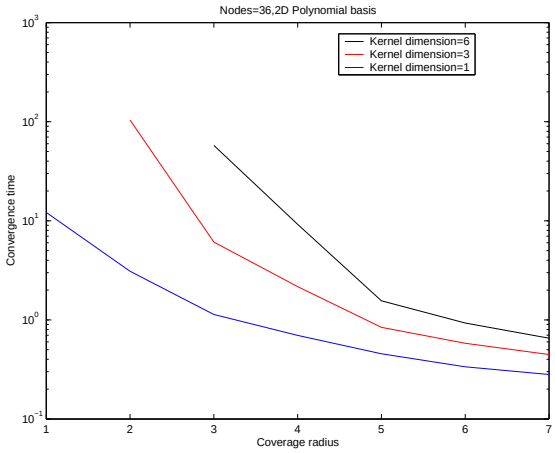

Figure 1.10: Minimum convergence time vs. number of neighbors for a 2D polynomial basis and 36 sensors.

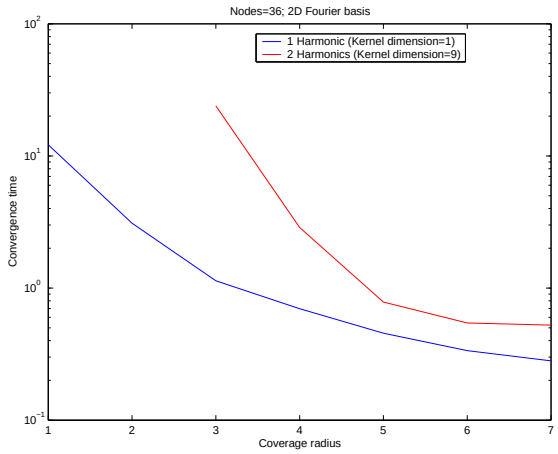

Figure 1.11: Minimum convergence time vs. number of neighbors for a 2D Fourier basis and 36 sensors.

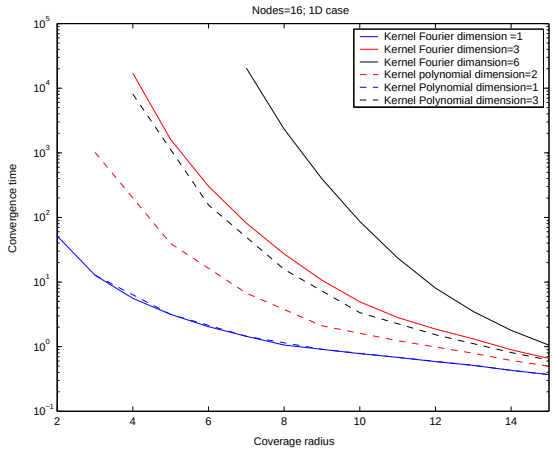

Figure 1.12: Minimum convergence time vs. number of neighbors for Fourier and polynomial basis.

1.4 Distributed projection algorithm robust against coupling noise

Finally, we consider the effect of the communication noise on the proposed distributed projection algorithm. The communication noise is modeled as an additive term in the update equation (1.2.5) that becomes

$$z_i[k+1] = z_i[k] - \epsilon \sum_{j \in \mathcal{N}_i} L_{ij} z_j[k] - \epsilon \sum_{j \in \mathcal{N}_i} L_{ij} n_{ij}[k], \tag{1.4.1}$$

so that in vector notation we have

$$\boldsymbol{z}[k+1] = \boldsymbol{W}\boldsymbol{z}[k] + \boldsymbol{v}[k] = \boldsymbol{W}^k \boldsymbol{z}[0] + \sum_{l=0}^{k} \boldsymbol{W}^{k-l} \boldsymbol{v}[l] \tag{1.4.2}$$

where $\boldsymbol{v}[k]$ is the vector noise, whose entries are assumed to be zero mean, uncorrelated random variables with variances

$$\sigma_i^2 = \epsilon^2 \sum_{j \in \mathcal{N}_i} |L_{ij}|^2 \sigma_n^2, \tag{1.4.3}$$

with σ_n^2 equal to the variance of any single noise contribute $n_{ij}[k]$. It is easy to show that the entries of the output additive noise term $\sum_{l=0}^{k} \boldsymbol{W}^{k-l} \boldsymbol{v}[l]$ tend to have variance diverging with time. This is indeed a generalization of what was observed in [7] for the simpler case of average consensus techniques. It can be showed with simple algebraic manipulations that the output additive noise vector $\mathbf{w}(k)$ has zero mean and covariance matrix

$$\mathbf{C}_{\mathbf{w}}(k) = \sum_{l=0}^{k-1} \mathbf{W}^l \boldsymbol{C}_v (\mathbf{W}^l)^H. \tag{1.4.4}$$

where $\boldsymbol{C}_v = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_N^2)$ is the covariance matrix of the noise vector $\boldsymbol{v}[k]$. Since $\mathbf{W}$ is symmetric by construction, it can be diagonalized as $\mathbf{W} = \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^H$, where the

diagonal matrix $\mathbf{\Lambda} = \text{diag}([\lambda_1(\mathbf{W}), \ldots, \lambda_N(\mathbf{W})])$ contains the eigenvalues of $\mathbf{W}$ and $\boldsymbol{R} = [\boldsymbol{U}|\boldsymbol{U}^{\perp}]$. The matrix $\boldsymbol{R}$ contains all the eigenvectors of the matrix $\boldsymbol{W}$, both the eigenvectors associated to its kernel and those relative to the kernel orthogonal subspace. The diagonalization of the l-th power of the matrix $\boldsymbol{W}$ is given by $\mathbf{W}^l = \mathbf{R}\mathbf{\Lambda}^l\mathbf{R}^H$, so that we have

$$\mathbf{C}_{\mathbf{w}}(k) = \sum_{l=0}^{k-1} \mathbf{R}\mathbf{\Lambda}^l\mathbf{R}^H \boldsymbol{C}_v (\mathbf{R}\mathbf{\Lambda}^l\mathbf{R}^H)^H$$

The eigenvalues of the covariance matrix $\mathbf{C}_{\mathbf{w}}(k)$ determine the average variance of the noise terms at the k-th iteration. The average noise variance resulting satisfies the following relation

$$\begin{aligned} \sigma^2_{\mathbf{w}}(k) &= \frac{1}{N}\text{Tr}(\mathbf{C}_{\mathbf{w}}(k)) \\ &\leq \frac{\sigma^2_{\min}}{N} \sum_{i=1}^{N} \sum_{l=0}^{k} \lambda_i(\boldsymbol{W})^{2l} \end{aligned} \tag{1.4.5}$$

with $\lambda_i(\mathbf{W}) = 1 - \epsilon\lambda_i(\mathbf{L}), \forall i \in \{1, \ldots, N\}$. To get some insight into the behavior of the noise variance as k increases, let us single out the effect of the largest eigenvalues of $\mathbf{W}$. Since these eigenvalues are equal to one (they correspond to the null eigenvalues of the matrix $\boldsymbol{L}$), the noise variance tend to diverge with time. A possible way to get rid of this annoying result is to use stochastic approximation theory to avoid the noise variance diverging problem and still converge, in mean square sense, to the desired projection vector. The proposed algorithm consists in using a decreasing step-size in the update equation, e.g. modifying it as follows

$$\begin{aligned} \mathbf{z}[k+1] &= (\mathbf{I} - \frac{\epsilon}{(k+1)^{\eta}}\mathbf{L})\mathbf{z}[k] + \mathbf{v}[k] \\ &= \prod_{m=0}^{k} \mathbf{W}_m \mathbf{z}[0] + \mathbf{w}[k+1] \end{aligned} \tag{1.4.6}$$

where $\mathbf{W}_m = (\mathbf{I} - \frac{\epsilon}{(m+1)^\eta}\mathbf{L})$ and $\mathbf{w}[k+1]$ represents the output additive noise term at the $(k+1)$-th iteration. In the following we consider the case of symmetric $\mathbf{L}$. We use the eigenvalue decomposition $\mathbf{L} = \mathbf{R}\mathbf{\Lambda}_L\mathbf{R}^\mathbf{T}$, where $\mathbf{\Lambda}_L = \text{diag}(\lambda_1(\mathbf{L}), \ldots, \lambda_N(\mathbf{L}))$ is the diagonal matrix whose diagonal entries are arranged in nondecreasing order, with $\lambda_1(\mathbf{L}) = \ldots \lambda_r(\mathbf{L}) = 0 < \lambda_{r+1}(\mathbf{L}) \leq \cdots \leq \lambda_N(\mathbf{L})$. It can be showed that the noiseless collective dynamic of the system asymptotically converges to [73]

$$\tilde{\mathbf{x}} = \mathbf{U}\mathbf{U}^T\mathbf{x}_0, \text{ if } \eta \in [0, 1]$$

or

$$\tilde{\mathbf{x}} = \mathbf{U}\mathbf{U}^T\mathbf{x}_0 + \sum_{i=r+1}^{N} \tilde{\lambda}_i \mathbf{r}_i {\mathbf{r}_i}^T \mathbf{x}_0, \text{ if } \eta > 1 \tag{1.4.8}$$

where $\mathbf{U}$ denotes the $N \times r$ matrix composed by the first r columns of $\mathbf{R}$; $\mathbf{r}_i, \forall i \in \{r+1, \ldots, N\}$, are the eigenvectors of $\mathbf{L}$ associated to its non-null eigenvalues (i.e. $\mathbf{U}^\perp$, the last $N - r$ columns of $\mathbf{R}$); the terms $\tilde{\lambda}_i, \forall i \in \{r+1, \ldots, N\}$, in the case $\eta > 1$, are upper and lower bounded by the following expressions

$$0 < \hat{\lambda}_i < \tilde{\lambda}_i < \bar{\lambda}_i < 1 \tag{1.4.9}$$

with $\hat{\lambda}_i = \exp(\ln(1 - \epsilon\lambda_{N-i+1}(\mathbf{L}))\zeta(\eta))$, and $\bar{\lambda}_i = \exp(-\epsilon\lambda_{N-i+1}(\mathbf{L})\zeta(\eta))$, where $\zeta(\eta)$ is the Riemann zeta function. Furthermore, the covariance matrix of the output noise vector $\mathbf{w}[k]$ in (1.4.6) can be written as

$$\mathbf{C}_w[k] = \frac{\sigma^2}{k^{2\eta}}\mathbf{I} + \sigma^2\tilde{\mathbf{U}}[\sum_{m=1}^{k-1} \frac{1}{m^{2\eta}} \prod_{l=m+1}^{k} (\mathbf{I} - \frac{1}{l^\eta}\mathbf{\Lambda}_L)^2]\tilde{\mathbf{U}}^T \tag{1.4.10}$$

and the average variance of its entries as

$$\sigma_w^2[k] = \frac{\sigma^2}{N}\sum_{i=1}^{N}(\frac{k}{2\eta} + \sum_{m=1}^{k-1} \frac{1}{m^{2\eta}} e^{2\sum_{l=m+1}^{k} \ln(1 - \frac{\epsilon}{l^\eta}\lambda_i(\mathbf{L}))}). \tag{1.4.11}$$

It can be showed that, as k goes to infinity, as long as η is chosen to be strictly greater than $1/2$, the output noise variance achieves asymptotically a value $\sigma_w^2[\infty]$ which is upper bounded as

$$\sigma_w^2[\infty] < \epsilon^2 \sigma^2 \zeta(2\eta), \tag{1.4.12}$$

whereas if $\eta \leq 1/2$ it diverges to infinity. Thus, if $\eta \in (1/2, 1]$, the algorithm projects the useful vector onto the kernel of $\mathbf{L}$, as desired, with an additive noise whose variance remains bounded.

1.4.1 Trade-off between convergence speed and accuracy

In this Section we prove that the optimal choice of the parameters η and ϵ to be used in the update equation (1.4.6) is given by a trade-off between convergence speed and accuracy of the final result in terms of noise variance. First of all, assuming that $\eta \in (1/2, 1]$, we achieve an expression for the error $\| z(k) - z^* \|$ that represents a measure of the distributed iterative algorithm convergence speed. This expression together with the results on the final noise variance given by (1.4.12) permits us to analyse the problem of the optimal choice of the parameters η and ϵ. We have

$$\begin{aligned} \|\boldsymbol{z}(k+1) - \boldsymbol{z}^*\| &= \|(\prod_{j=0}^{k} \boldsymbol{W}_j)\boldsymbol{z}(0) - \boldsymbol{U}\boldsymbol{U}^H \boldsymbol{z}(0)\| \\ &= \|\boldsymbol{W}_k(\prod_{j=0}^{k-1} \boldsymbol{W}_j \boldsymbol{z}(0)) - \boldsymbol{U}\boldsymbol{U}^H(\prod_{j=0}^{k-1} \boldsymbol{W}_j \boldsymbol{z}(0))\| \end{aligned} \tag{1.4.13}$$

where we have used the fact that

$$\boldsymbol{U}\boldsymbol{U}^H(\prod_{j=0}^{k-1} \boldsymbol{W}_j \boldsymbol{z}(0)) = \boldsymbol{U}\boldsymbol{U}^H \boldsymbol{z}(0). \tag{1.4.14}$$

From (1.4.13) and Appendix C, it follows that

$$\begin{aligned}\|\boldsymbol{z}(k+1)-\boldsymbol{z}^*\| &\leq \rho(\boldsymbol{W}_k-\boldsymbol{U}\boldsymbol{U}^H)\|(\prod_{j=0}^{k-1}\boldsymbol{W}_j\boldsymbol{z}(0))-\boldsymbol{U}\boldsymbol{U}^H(\prod_{j=0}^{k-1}\boldsymbol{W}_j\boldsymbol{z}(0))\| \\ &= \rho(\boldsymbol{W}_k-\boldsymbol{U}\boldsymbol{U}^H)\|\boldsymbol{z}(k)-\boldsymbol{z}^*\|. \qquad (1.4.15)\end{aligned}$$

Repeating the same argument for $j=0$ to k we finally get

$$\|\boldsymbol{z}(k+1)-\boldsymbol{z}^*\| \leq (\prod_{j=0}^{k}\rho(\boldsymbol{W}_j-\boldsymbol{U}\boldsymbol{U}^H))\|\boldsymbol{z}(0)-\boldsymbol{z}^*\|. \qquad (1.4.16)$$

Equation (1.4.1) establishes that the maximization of the convergence rate of the algorithm with decreasing step-size is equivalent to minimize the product of the spectral radius of the matrices sequence $\{\boldsymbol{W}_j\}$. Since the sequence of step-size $\frac{\epsilon}{(k+1)^\eta}$ is a decreasing function of the iteration index k, it is simple to prove the following result: minimizing the convergence rate of the algorithm given by (1.4.6) with respect to the matrix $\boldsymbol{L}$ and the initial step-size ϵ is equivalent to solve the optimization problem (1.3.7). In other words, the optimal choice of $\boldsymbol{L}$ and ϵ that maximizes the convergence rate of the algorithm with decreasing step-size is the same as in the case of the algorithm with constant step-size given by the update equation (1.2.5). Since the optimal initial step size is given by

$$\epsilon^\star = \frac{2}{\lambda_{(1)}\left(\overline{\mathbf{L}}\right)+\lambda_{(N-r)}\left(\overline{\mathbf{L}}\right)} \qquad (1.4.17)$$

it is simple to prove that

$$\|\boldsymbol{z}(k+1)-\boldsymbol{z}^*\| \leq (\prod_{j=0}^{k}(1-\lambda_2(\overline{\mathbf{L}})\epsilon^*/(k+1)^\eta)\|\boldsymbol{z}(0)-\boldsymbol{z}^*\|, \qquad (1.4.18)$$

where $\lambda_2(\overline{\mathbf{L}})$ is the smallest eigenvalues of the positive definite symmetric matrix $\overline{\mathbf{L}}$ introduced in the previous sections. Now, since $1-a \leq \exp^{-a}$, $0 \leq a \leq 1$, we have

$$\|\boldsymbol{z}(k+1)-\boldsymbol{z}^*\| \leq (\prod_{j=0}^{k}(\exp^{-\lambda_2(\overline{\mathbf{L}})\sum_{j=0}^{k}\epsilon^*/(k+1)^\eta})\|\boldsymbol{z}(0)-\boldsymbol{z}^*\| \qquad (1.4.19)$$

that shows the existence of a trade-off between the final noise variance and the convergence rate of the sequence $\boldsymbol{z}(k)$ to the desired vector $\boldsymbol{z}^*$. In fact, from (1.4.19) it follows that the convergence speed is closely related to the rate at which the step-size sequence, $\epsilon^*/(j+1)^\eta$, sums to infinity. For a faster rate, we want the step-size sequence to sum up fast to infinity, i.e., the step-size sequence elements to be large. As a consequence choosing a lower η yields a higher convergence rate; however, from equation (1.4.12), being $\zeta(2\eta)$ a decreasing function of η in the interval $(1/2, 1]$, the choice of the parameter η must necessarily consider both final noise variance and convergence rate. Observing equation (1.4.12), in order to limit the effect of the initial step-size on the final output noise variance, we could impose also a constraint on the initial step-size ϵ. In this case, the maximization of the convergence rate with respect to $\boldsymbol{L}$ and ϵ can be reformulated as the following optimization problem

$$\begin{aligned} &\underset{\overline{\mathbf{L}},\,\gamma}{\text{minimize}} && \gamma \\ &\text{subject to} && -\gamma\bar{\mathbf{I}} \leq \bar{\mathbf{I}} - \epsilon\overline{\mathbf{L}} \leq \gamma\bar{\mathbf{I}}, \\ & && \overline{\mathbf{L}} \succ \mathbf{0}, \quad \overline{\mathbf{L}} = \overline{\mathbf{L}}^T, \epsilon \leq \tilde{\epsilon} \\ & && \left[\mathbf{U}^{\perp}\overline{\mathbf{L}}\mathbf{U}^{\perp\mathbf{T}}\right]_{ij} = 0 \quad \forall i,j \in \mathcal{B}. \end{aligned} \tag{1.4.20}$$

where $\tilde{\epsilon}$ is the greatest possible value of the initial step-size according to the accuracy requirement. This optimization problem is not convex and in this case it is not possible to rewrite our problem in an alternative convex form. If we assume that, after suitable processing, the signal received by the i-th node on the channel from node j to node i has the following expression

$$r_{ij}[k] = \sqrt{P_r} z_j[k] + n_{ij}[k]$$

where P_r is the minimum required received power achieved implementing power control, the update equation (1.4.6) can be rewritten as

$$z_i[k+1] = z_i[k] - \beta\sqrt{P_r}\sum_{j\in\mathcal{N}_i} L_{ij} z_j[k] - \beta\sum_{j\in\mathcal{N}_i} L_{ij} n_{ij}[k], \tag{1.4.21}$$

with the product $\beta\sqrt{P_r}$ equivalent to the initial step-size ϵ. The final output noise variance assumes the following expression

$$\sigma_w^2[\infty] < \beta^2\sigma^2\zeta(2\eta); \tag{1.4.22}$$

from (1.4.22) it follows that the trade-off between accuracy and convergence rate with respect to ϵ can be faced exploiting another degree of freedom, i.e. the minimum required received power. The parameter β can be set to a certain value while the transmit power can be chosen in order to obtain a desired value of the initial step-size ϵ without increasing the final output noise variance.

1.5 Signal subspace order selection

The projection operator used in this wok can be seen as the result of a minimum mean square error (MMSE) algorithm applied to the observed data vector, with the peculiarity of being implemented in a distributed way and able to converge with the minimum convergence time. As well known in the application of MMSE algorithms to real data [119], the selection of the signal subspace dimension k is a critical step. The MSE of the basis expansion can be described by the sum of a square bias and a variance term

$$MSE = \text{bias}^2 + \text{var}_n \tag{1.5.1}$$

where bias^2 depends on the actual set of basis functions and var_n, in the case of equal observation noise variance, depends linearly on σ_ξ^2 and the dimension of the basis

expansion

$$\mathrm{var}_n = \sigma_\xi^2 \mathrm{Tr}(\boldsymbol{U}\boldsymbol{U}^H\boldsymbol{U}^H\boldsymbol{U}) = \sigma_\xi^2 \frac{k}{N}. \tag{1.5.2}$$

Small values of k are useful to get a strong noise reduction, but at the expense of a large bias; conversely, higher values of k provide better estimates of the useful signal, but with a higher noise variance. Figure 1.13 shows an example of MSE obtained

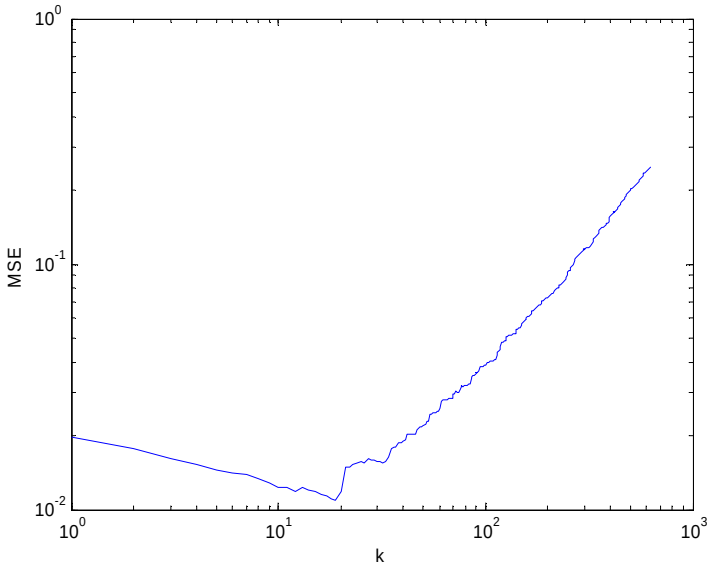

Figure 1.13: Mean Square Error vs. signal subspace dimension.

by projecting a power spatial density modeled as in , plus additive zero mean white noise with variance σ_ξ^2, onto a 2D Fourier basis, as a function of the model order. The MSE initially decreases because the approximation improves as the model order increases; however, the noise term increases as $k\sigma_\xi^2$. As a consequence, the overall MSE exhibits a minimum, as evident in Figure 1.13; therefore, it is important to choose the correct signal subspace order that guarantee the optimal trade-off between variance and bias of the estimation. We are interested in estimating a bandlimited spatial function defined only on a specific geographical area; it is clear that analyzing and

representing scientific data of this kind will be facilitated if a basis of functions can be found that are "spatio-spectrally" concentrated, i.e. "localized" in both domains at the same time. Here, we give a theoretical overview of one particular approach to this "concentration" problem, as originally proposed for time series by Slepian and coworkers, in the 1960s. Slepian functions represent an orthogonal family of functions that are all defined on a common, e.g. geographical, domain, where they are either optimally concentrated or within which they are exactly limited, and which at the same time are exactly confined within a certain bandwidth, or maximally concentrated therein. The measure of concentration is invariably a quadratic energy ratio, which, though only one choice out of many is perfectly suited to the nature of the problem considered. In the next section we briefly review the theory of the Slepian functions both in one dimension and in the Cartesian plane; these results provide us a value of the actual dimension of the subspace of the functions "essentially" space-band limited in a certain domain.

1.5.1 Theory of Slepian functions

We start with the one-dimensional case and use t to denote time or one-dimensional space and ω for angular frequency, and adopt a normalization convention in which a real-valued space(time)-domain signal $f(t)$ and its Fourier transform $F(\omega)$ are related by

$$f(t) = (2\pi)^{-1} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega \tag{1.5.3}$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} \mathrm{dt} \tag{1.5.4}$$

The problem of finding the strictly bandlimited signal

$$g(t) = (2\pi)^{-1} \int_{-W}^{W} G(\omega) e^{i\omega t} d\omega, \tag{1.5.5}$$

that is maximally (though by virtue of the Paley-Wiener theorem never completely) concentrated into a space(time) interval $|t| < T$ was first considered by Slepian, Landau and Pollak. The optimally concentrated signal is taken to be the one with the least energy outside of the interval,in other words such that

$$\lambda = \frac{\int_{-T}^{T} g^2(t)\mathrm{dt}}{\int_{-\infty}^{\infty} g^2(t)\mathrm{dt}} = \text{maximum} \tag{1.5.6}$$

Bandlimited functions $g(t)$ satisfying the previous variational problem have spectra $G(\omega)$ that satisfy the frequency domain convolutional integral eigenvalue equation

$$\int_{-W}^{W} D(\omega,\omega^{'})G(\omega^{'})d\omega^{'} = \lambda G(\omega) \;\; |\omega| \leq W \tag{1.5.7}$$

$$D(\omega,\omega^{'}) = \frac{\sin T(\omega-\omega^{'})}{\pi(\omega-\omega^{'})} \tag{1.5.8}$$

The corresponding time- or spatial-domain formulation is

$$\int_{-T}^{T} D(t,t^{'})g(t^{'})d\mathrm{t}^{'} = \lambda g(t) \;\; |t| \leq T \tag{1.5.9}$$

$$D(t,t^{'}) = \frac{\sin W(t-t^{'})}{\pi(t-t^{'})} \tag{1.5.10}$$

The "prolate spheroidal eigenfunctions" $g_1(t), g_2(t), \ldots$ that solves equation (1.5.9) form a doubly orthogonal set. When they are chosen to be orthonormal over infinite time they are also orthogonal over the finite interval $|t| \leq T$

$$\int_{-\infty}^{\infty} g_\alpha(t)g_\beta(t)dt = \delta_{\alpha\beta}, \tag{1.5.11}$$

$$\int_{-T}^{T} g_\alpha(t)g_\beta(t)dt = \lambda_\alpha\delta_{\alpha\beta}. \tag{1.5.12}$$

A change of variables and a scaling of the eigenfunctions transforms equation (1.5.7) into the dimensionless eigenproblem

$$\int_{-1}^{1} D(x,x^{'})\varphi(x^{'})d\mathrm{x}^{'} = \lambda\varphi(x) \tag{1.5.13}$$

$$D(x, x') = \frac{\sin TW(x - x')}{\pi(x - x')} \tag{1.5.14}$$

Equation (1.5.14) shows that the eigenvalues $\lambda_1 > \lambda_2 > \ldots$ and suitably scaled eigenfunctions $\varphi_1(x), \varphi_2(x), \ldots$ depend only upon the time-bandwidth product TW. The sum of the concentration eigenvalues λ relates to this product by

$$N^{1D} = \sum_{\alpha=1}^{\infty} \lambda_\alpha = \frac{2TW}{\pi}. \tag{1.5.15}$$

The shape of the eigenvalue spectrum has a characteristic step shape, showing significant $\lambda \approx 1$ and insignificant $\lambda \approx 0$ eigenvalues separated by a narrow transition band.Thus, this "Shannon number" is a good estimate of the number of significant eigenvalues, or, roughly speaking, N^{1D} is the number of signals $f(t)$ that can be simultaneously well concentrated into a finite time interval $|t| \leq T$ and a finite frequency interval $|\omega| \leq W$. In other words,N^{1D} is the approximate dimension of the space of signals that is "essentially" space(time)-limited to T and bandlimited to W, and using the orthogonal set $g_1, g_2, \ldots, g_{N^{1D}}$ as its basis is parsimonious. Now, we extend these results to the two-dimensional case. A square-integrable function $f(\mathbf{x})$ defined in the plane has the two-dimensional Fourier representation

$$f(\mathbf{x}) = (2\pi)^{-2} \int_{-\infty}^{\infty} F(k) e^{i\mathbf{k}\cdot\mathbf{x}} d\mathbf{k} \tag{1.5.16}$$

$$F(k) = \int_{-\infty}^{\infty} f(\mathbf{x}) e^{-i\mathbf{k}\cdot\mathbf{x}} \mathrm{d}\mathbf{x} \tag{1.5.17}$$

We use $g(\mathbf{x})$ to denote a function that is bandlimited to K, an arbitrary sub-region of spectral space,

$$g(\mathbf{x}) = (2\pi)^{-2} \int_K G(k) e^{i\mathbf{k}\cdot\mathbf{x}} d\mathbf{k}. \tag{1.5.18}$$

Following Slepian, we seek to concentrate the power of $g(\mathbf{x})$ into a finite spatial region $R \in \mathbb{R}^2$ of area A:

$$\lambda = \frac{\int_R g^2(\mathbf{x})\mathrm{d}\mathbf{x}}{\int_{-\infty}^{\infty} g^2(\mathbf{x})\mathrm{d}\mathbf{x}} = \text{maximum} \tag{1.5.19}$$

Bandlimited functions $g(\mathbf{x})$ that maximize the Rayleigh quotient (1.5.19) solve the Fredholm integral equation

$$\int_K D(\mathbf{k},\mathbf{k}^{'})G(\mathbf{k}^{'})d\mathbf{k}^{'} = \lambda G(\mathbf{k}) \;\; \mathbf{k} \in K \tag{1.5.20}$$

$$D(\mathbf{k},\mathbf{k}^{'}) = (2\pi)^{-2}\int_R \exp^{i(\mathbf{k}-\mathbf{k}^{'})\cdot\mathbf{x}}\,\mathrm{d}\mathbf{x}. \tag{1.5.21}$$

The corresponding problem in the spatial domain is

$$\int_R D(\mathbf{x},\mathbf{x}^{'})g(\mathbf{x}^{'})d\mathbf{x}^{'} = \lambda g(\mathbf{x}) \;\; \mathbf{x} \in R \tag{1.5.22}$$

$$D(\mathbf{x},\mathbf{x}^{'}) = (2\pi)^{-2}\int_K \exp^{i(\mathbf{x}-\mathbf{x}^{'})\cdot\mathbf{k}}\,\mathrm{d}k. \tag{1.5.23}$$

The bandlimited spatial-domain eigenfunctions $g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots$ and eigenvalues $\lambda_1, \lambda_2, \ldots$ that solve equation (1.5.22) may be chosen to be orthonormal over the whole plane $\| \mathbf{x} \| \leq \infty$ in which case they are also orthogonal over R:

$$\int_{-\infty}^{\infty} g_\alpha(t)g_\beta(t)dt = \delta_{\alpha\beta}, \tag{1.5.24}$$

$$\int_R g_\alpha(t)g_\beta(t)dt = \lambda_\alpha\delta_{\alpha\beta}. \tag{1.5.25}$$

Concentration to the disk-shaped spectral band $K = \{\mathbf{k} : \| k \| \leq K\}$ allows us to rewrite equation (1.5.22) after a change of variables and a scaling of the eigenfunctions as

$$\int_{R_*} D(\boldsymbol{\xi},\boldsymbol{\xi}^{'})\varphi(\boldsymbol{\xi}^{'})d\boldsymbol{\xi}^{'} = \lambda\varphi(\boldsymbol{\xi}) \tag{1.5.26}$$

$$D(\boldsymbol{\xi},\boldsymbol{\xi}^{'}) = \frac{K\sqrt{A/4\pi}}{2\pi}\frac{J_1(K\sqrt{A/4\pi}\,\| \boldsymbol{\xi}-\boldsymbol{\xi}^{'} \|)}{\| \boldsymbol{\xi}-\boldsymbol{\xi}^{'} \|} \tag{1.5.27}$$

where the region R_* is scaled to area 4π and J_1 is the is the first-order Bessel function of the first kind. Equation (1.5.26) shows that, also in the two-dimensional case, the eigenvalues $\lambda_1, \lambda_2, \ldots$ and the scaled eigenfunctions $\varphi_1(\boldsymbol{\xi}), \varphi_2(\boldsymbol{\xi}), \ldots$ depend only on the combination of the circular bandwidth K and the spatial concentration area A, where the quantity $K^2A/4\pi$ now plays the role of the time-bandwidth product TW in the one-dimensional case. The sum of the concentration eigenvalues λ defines the two-dimensional Shannon number N^{2D} as

$$N^{2D} = \sum_{\alpha=1}^{\infty} \lambda_\alpha = \frac{K^2 A}{4\pi}. \tag{1.5.28}$$

Just as N^{1D},N^{2D} is the product of the spectral and spatial areas of concentration multiplied by the Nyquist density.And, similarly, it is the effective dimension of the space of "essentially" space- and bandlimited functions in which the set of two-dimensional functions $g_1, g_2, \ldots, g_{N^{2D}}$ may act as a sparse orthogonal basis. The Slepian basis expansion by construction represents space(time)-band limited signals with a minimum number of basis functions; as a consequence, choosing this functions for the basis expansion we achieve the minimum bias in the useful signal reconstruction with the minimum possible signal subspace order.

1.5.2 Basis Expansion Square Bias

Since increasing the node density a random geometric graph tend to have the properties of a regular one it is useful to achieve an analytic expression for the square bias of the basis expansion assuming that sensors are on a $(M \times M)$ regular grid. We use Niedzwieckis results from [10] specializing them to our application. These results hold for any possible choice of the basis functions. Let us denote with $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k$ the columns of the matrix $\boldsymbol{U}$ that represent the basis spanning the useful signal subspace

onto which to project the observations. We introduce the row vector

$$\boldsymbol{u}(l(m,n)) = [\boldsymbol{u}_1(l(m,n)), \boldsymbol{u}_2(l(m,n)), \ldots, \boldsymbol{u}_k(l(m,n))], \tag{1.5.29}$$

that is the row of index $l = (m-1) * M + n$ of the matrix $\boldsymbol{U}$, i.e. the row containing the values assumed by all the basis functions in the point (m,n) where the l-th sensor of the network is located. We define the instantaneous frequency response of the basis expansion estimator according to

$$H(m,v,\nu_x,\nu_y) = \boldsymbol{u}(l(m,n))(\boldsymbol{U}^H\boldsymbol{U})^{-1}\sum_k\sum_s \boldsymbol{u}(l(m,n))^H \mathrm{e}^{-j2\pi(\nu_x(m-k)+\nu_y(n-s))} \tag{1.5.30}$$

where $k,s \in 1,\ldots,M$ and $\nu_x,\nu_y \in K$ with K the spectral region occupied by the physical process to be estimated. The sum in (1.5.30)

$$\sum_k\sum_s \boldsymbol{u}(l(m,n))^H \mathrm{e}^{-j2\pi(\nu_x(m-k)+\nu_y(n-s))} \tag{1.5.31}$$

projects the complex exponential onto the basis function at the sensors grid positions, i.e., we calculate the inner product with every basis function. Then, the realization at space position (m,n) is calculated by left multiplying with $\boldsymbol{u}(l(m,n))$. The complex exponential in (1.5.30) is shifted in the two spatial coordinates by m and n; thus,$|H(m,v,\nu_x,\nu_y)|$ is the instantaneous amplitude response of the basis expansion at space position (m,n). The phase of $H(m,v,\nu_x,\nu_y)$, which is expressed by $\arg(H(m,v,\nu_x,\nu_y))$, is the instantaneous phase shift of the basis expansion at space position (m,n). The design goal for a basis expansion is to have no amplitude error $|H(m,v,\nu_x,\nu_y)| = 1$ and no phase error $\arg(H(m,v,\nu_x,\nu_y)) = 0$. Therefore, the instantaneous error characteristic of the basis expansion is defined as

$$E(m,v,\nu_x,\nu_y) = |1 - H(m,v,\nu_x,\nu_y)|^2. \tag{1.5.32}$$

The square bias per symbol $\text{bias}^2(m,n)$ of the basis expansion estimator can be expressed as the integral over the instantaneous error characteristic $E(m,v,\nu_x,\nu_y)$ multiplied by the power spectral density of the physical process to be estimated $\boldsymbol{z}(m,n)$

$$\text{bias}^2(m,n) = \int\int_K E(m,v,\nu_x,\nu_y) S_{zz}(\nu_x,\nu_y) d\nu_x d_{\nu y} \tag{1.5.33}$$

where $S_{zz}(\nu_x,\nu_y)$ is given by

$$S_{zz}(\nu_x,\nu_y) = \sum_m \sum_n R_{zz}(m,n) \mathrm{e}^{-j2\pi(\nu_x m + \nu_y n)}, \tag{1.5.34}$$

and $R_{zz}(m,n)$ is the auto-correlation function of the physical process to be estimated. The square bias for the whole estimation is given by

$$\text{bias}^2(m,n) = \frac{1}{M^2} \sum_m \sum_n \text{bias}^2(m,n). \tag{1.5.35}$$

The result in (1.5.33) proves that the dimension of the useful signal subspace necessary to have a certain accuracy in the reconstruction is strictly related to the variability of the signal, i.e. to the properties of its spatial frequencies spectrum.

1.5.3 Final considerations for the signal subspace order selection

Using the results in Sections 1.5.2 and 1.5 we are able to find the optimal signal subspace order, i.e. the order that gives the minimum MSE in the useful signal reconstruction. Let us indicate by k_0 the order yielding the minimum MSE. The use of the distributed projection algorithm proposed in this work introduces a further element in the choice of the useful subspace dimension. The existence of a distributed algorithm converging to the projection onto a subspace of a given order k requires the transmit power used by each node be large enough to establish a direct link with

at least k neighbors, as indicated in [120]. The number of neighbors is clearly related to the transmit power of each node. If we assume a uniform spatial distribution of the nodes, with spatial density ρ, the coverage radius r_0 necessary to get an average number k of neighbors for each node is $r_0 = \sqrt{k/\Pi\rho}$. If we denote by P_R the minimum receive power necessary to have a link with a sufficient quality and we assume that the transmitted power P_T attenuates with the square of the distance, the average number of neighbors k_ρ reachable with a transmit power P_T is $k_\rho = \Pi\rho P_T/P_R$. Combining these arguments with the selection of the order providing the minimum MSE, it turns out that the optimal order, compatible with the transmit power constraint, is

$$k_{\text{opt}} = \min(k_0, k_p). \tag{1.5.36}$$

This means that if the transmit power is small, i.e. $k_\rho < k_0$, the method will not be able to minimize the MSE, because of a large bias, whereas if the transmit power is sufficiently large, it is not necessary to waste power, as it is sufficient to use the transmit power necessary to guarantee $k_\rho = k_0$.

1.6 A practical application: cooperative spectrum sensing using decentralized projection algorithms

The distinguishing characteristic of cognitive radio is the ability of its nodes to allocate power over temporally unoccupied portions of the spectrum. This adaptability to the electromagnetic environment is the basic feature enabling the potential spectral efficiency gain of cognitive radios. The basic information needed by a cognitive transmitter is the distribution of the power spectral density at the location of its intended receiver and of the primary users, i.e. the users who have the right of not being disturbed by opportunistic cognitive users. As a whole, the basic information

enabling a cognitive radio network to operate in this context is then the knowledge of the spatial distribution of the power spectral density, or spatial spectral density for short. This information could be acquired by a wireless sensor network whose nodes estimate the local power spectral density and send this information to a network control node that forms a spatial map of the spectral occupancy. Within this framework, once a pair of cognitive users wish to establish a link, they interrogate the control center and decide which channels are more appropriate for communication, without interfering with the primary users. The major criticality of this approach is that, as in any centralized system, there is a bottleneck represented by the control node that needs to periodically collect a lot of information from potentially many sensors. Furthermore, the spatial sampling of the spectral density operated by each node requires a high density of the sensing nodes. To make possible an economy of scale of such nodes, it is clear that they cannot be too sophisticated, they will have a limited power budget and maybe not all of them will be able to estimate the whole power spectral density of interest, but only a portion of it. Cooperation among sensing nodes can greatly improve the performance of the network, as already highlighted in [115],within a centralized framework. Conversely, we propose our distributed projection algorithm in order to allow the sensing nodes to cooperate with each other in order to achieve globally optimal goals, but without requiring the presence of a fusion center. The basic assumption underlying the proposed approach is that the spatial distribution of the estimated power, for each frequency channel, is a smooth function of space. More precisely, we assume that the power spatial density, for each subchannel, can be well approximated by a signal lying in a vector space of dimension (much) smaller than the number of sensing nodes. In this case, projecting the whole set of

estimated values onto the useful signal subspace, is a well known signal processing tool to reduce the effect of noise and contrast spatially uncorrelated shadowing phenomena. After projection, each sensing node will possess a more reliable estimate of the local spectral occupancy. These nodes could then deliver this information, upon request, to the interested cognitive users. We achieve a signal subspace projector without any node having full knowledge of the data gathered by all other nodes, with a very simple iterative algorithm where, at each iteration, each node simply takes a linear combination of the running estimates of its neighbors. Taking averages of the estimates present at each node neighbors is clearly a very simple way to reduce the effect of spatially uncorrelated noise. However, the overall effect of local averaging is to level out also the spatial distribution of the useful signal. This is in contrast with the application at hand where, conversely, it is precisely the power spatial variability that allows cognitive users to reuse locally unused spectrum holes. The well known average consensus algorithm [2], with all its variants like gossip algorithms for example, is a very particular case of our setup, as it corresponds to project the whole set of data onto a useful signal subspace spanned by a vector of all ones. But clearly this approach is too restrictive, as it leads to a spatially-invariant power distribution. Our solution to make local averages to reduce the noise, but without forcing the whole network to converge to a spatially-invariant distribution. The diffusion algorithms proposed in [116] are an alternative attractive strategy. However, our approach is more general, as it leads to totally general projection operators. An alternative approach, exploiting the sparsity of the power spectral and spatial density was recently proposed in [117], using the lasso operator to achieve a sparse estimation of the power spatial/spectral density. However, the approach of [117] requires the useful signal to

possess a sparse representation. Conversely, our approach does not really require a sparse representation and it is much simpler to implement than [117], as in our case every node simply takes a weighted linear combination of the estimates present at its neighbors. In this example, we make use of the theoretical results proved in the previous sections and apply them to the specific context of cooperative sensing. This allows us to make the optimal selection of the signal subspace order taking into account three major sources of error: additive noise or multiplicative fading; bias error due to mismatching between useful signal and finite order fitting; error due to using a finite number of iterations. We provide numerical examples supporting the validity of the proposed method for contrasting additive noise and fading.

Let us consider a network composed of N sensors, each measuring a wideband spectrum. We denote by

$$P(x_i, y_i) = A(x_i, y_i)S(x_i, y_i) + v(x_i, y_i) \tag{1.6.1}$$

the power spectral density measured by a node located at (x_i, y_i) at a given frequency, where $S(x_i, y_i)$ is the ideal power density, $A(x_i, y_i)$ models shadowing or fading effects, and $v(x_i, y_i)$ is observation noise. The spectrum to be monitored by the network is typically quite large (in the order of GHz). Since a proper spatial sampling requires the presence of many nodes, an economy of scale of the sensing nodes requires them to be quite simple and thus able to estimate only portions of the spectrum. The node measurement may be inaccurate just because of shadowing or noise effects. To improve the overall network accuracy, we propose a a cooperative strategy where nearby nodes interact with each other. The rational behind our approach is that the ideal spatial spectral density $S(x, y)$ is a smooth function of the spatial coordinates, whereas shadowing and noise are not. In mathematical terms, we assume that the

spatial distribution of the useful spectrum, for each frequency, can be well approximated by a two-dimensional (2D) signal lying in a vector space of dimension much smaller than the dimension of the observation space, typically equal to the number of nodes. Within the validity of this assumption, typically valid in practice, a strong noise reduction is achievable by projecting the observation onto the signal subspace. As an example, we consider a useful power spatial distribution, for each frequency, given by the superposition of the powers emitted by N_s primary sources, with each term modeled as a Cauchy bell:

$$S(x,y) = \sum_{i=1}^{N_s} \frac{P_i}{1 + ((x - x_i)^2 + (y - y_i)^2)/\sigma^2} \tag{1.6.2}$$

where P_i is the power emitted by node i and σ specifies the power spatial spread. This function can be expanded over a given basis, for example over the 2D Fourier or the wavelets basis, of infinite dimension. Our goal is to approximate the useful signal with a finite order expansion, possibly of low order, and to perform this operation in a distributed way, where each node interacts only with its immediate neighbors. In mathematical terms, let $\mathbf{z}_n$ denote the spectral power estimated by node n over a generic subcarrier. Collecting the measurements taken by every node over each subcarrier, we build the vector $\mathbf{z} := (\mathbf{z}_1, \ldots, \mathbf{z}_N)^T$ representing the spatial distribution of power across all the nodes. The smoothness assumption can be formulated by stating that $\mathbf{z}$ can be approximated as (1.2.1),where the columns of the matrix spanning the useful signal subspace can be chosen as the low frequency 2D Fourier or wavelet basis, for example. Hence, according to (1.2.2) a strong noise reduction may be obtained by projecting the observation vector onto the signal subspace;in order to achieve this goal we use our distributed algorithm. Each node initializes a state variable with its local measurement,i.e. the spectral power estimated by a node over

a generic subcarrier, and evolves, for each subcarrier, as the linear dynamic system in (1.2.5). Upon convergence, the proposed distributed projection algorithm allows every node of the sensing WSN to improve the reliability of its own spatial spectral density power measurement, through local interaction with its neighbors, reducing the effect of noise and fading phenomena. Local combination of observations from nearby nodes helps to reduce the effect of noise and fading without forcing the nodes to converge to a common (consensus) value allowing in this way cognitive users to find out locally unused spectrum holes.

We consider now the application of the proposed method to combat either additive or multiplicative noise. Given the signal model in (1.6.1) and (1.6.2), we consider for simplicity only the two extreme cases where either observation noise or shadowing are the dominant undesired effects. In the first case, i.e. setting $A(x_i, y_i) = 1$, we simply have $P(x_i, y_i) = S(x_i, y_i) + v(x_i, y_i)$. As an example, in Figure 1.14 we assume that the spatial distribution of the power received at a given frequency is given by the superposition of 4 Cauchy bells, as in (1.6.2), centered in the positions of the primary transmitters, plus additive spatially white noise. The wireless sensor network is composed of 2500 nodes uniformly distributed over a 2D grid. All the transmitters use the same power, i.e. $P_i = P$ in (1.6.2), and the noise has zero mean and variance $\sigma_n^2 = P$. Figure 1.14 shows the useful signal power (top left), the observation corrupted by noise (top right) and the reconstructions using two different orders, $k = 10$ and $k = 20$ (bottom). The signal subspace is composed by the 2D Fourier components up to order k. The projection is achieved using the simple dynamical system (1.2.5), with the matrix $\mathbf{W}$ computed according to the SDP reformulation of the minimization problem of the convergence time. It is evident the

strong noise reduction achievable with the proposed approach, at the expense of a small bias.

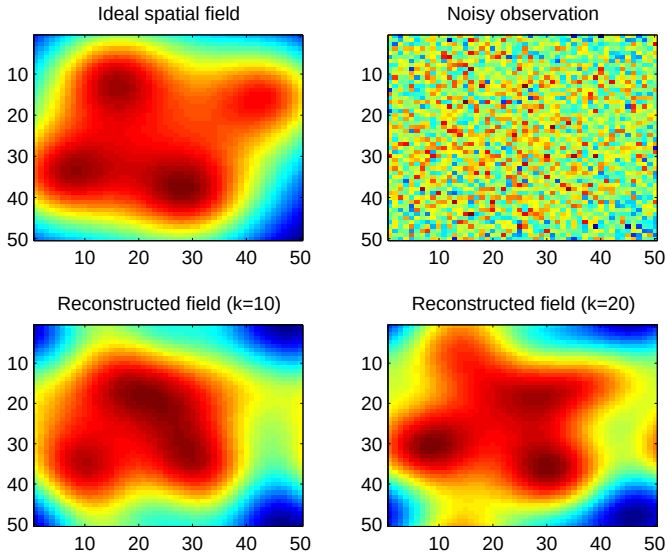

Figure 1.14: Example of field reconstruction in the presence of additive noise: ideal spatial field (top left); measured field (top right); field reconstructed with order $k = 10$ (bottom left) and $k = 20$ (bottom right).

A further application of the proposed method refers to an observation corrupted by a fading effect, modeled as a multiplicative, spatially uncorrelated, noise. In this case, it is useful to apply a homomorphic filtering to the measured field. In particular, we take the log of the measurement, thus getting

$$\log\left(P(x_i, y_i)\right) = log(S(x_i, y_i)) + log(A(x_i, y_i)). \tag{1.6.3}$$

Then, we apply the same algorithm as in the previous example and take the exp of the result. An example is shown in Figure 1.15, where it is evident the capability of the proposed distributed approach to provide a significant attenuation of the fading phenomenon as well.

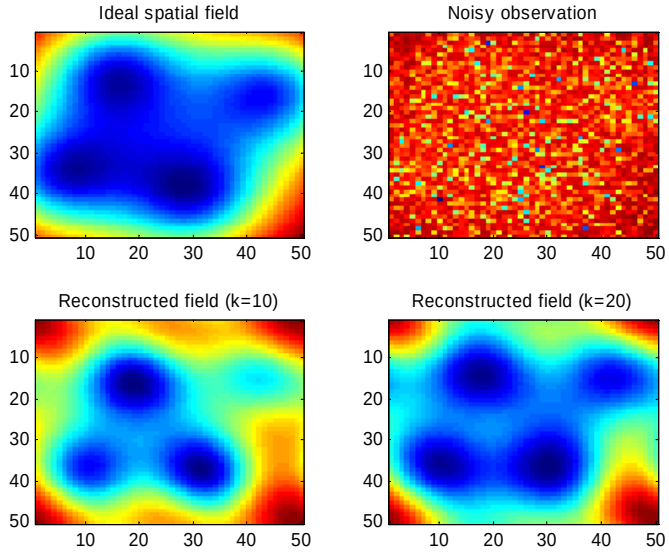

Figure 1.15: Example of field reconstruction in the presence of fading: ideal spatial field (top left); measured field (top right); field reconstructed with order $k = 10$ (bottom left) and $k = 20$ (bottom right).

If the additive noise variance is not too high we can apply a homomorphic filtering also in this case; for example, in Figure 1.16 we consider the case of useful signal

corrupted by fading effects and additive noise (SNRdB = 10).

Besides the bias error due to model mismatching or additive noise, a further error comes from the use of a finite number of iterations. Given a maximum delay necessary to reach the desired smoothing effect, the minimization of this error requires the maximization of the algorithm convergence rate. As an example, in Figure 1.17 we show the minimum convergence time obtained for a network of 25 sensors distributed over a unit square, as a function of the square coverage radius (assumed to be the same for every node). Assuming a power attenuation law $p_R = p_T/r^2$, the abscissa is proportional to the transmit power necessary to induce a unit receive power p_R. We considered both cases of sensors uniformly spaced (solid line) and randomly distributed (dashed line). Figure 1.17 refers to the projection onto a signal subspace spanned by two-dimensional Fourier bases including up to the harmonic of degree $d = 0$ and 1. The minimum number of neighbors is equal to the number of independent sinusoids of degree up to d. As expected, as the coverage area increases, the convergence time decreases. However, this entails a greater transmit power to cover a larger area. On the other hand, the convergence time increases if, for a given number of neighbors, the dimension of the kernel space increases.

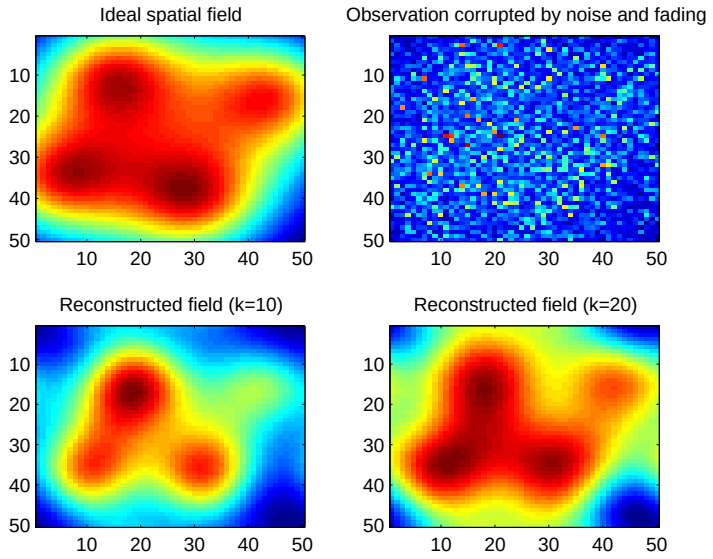

Figure 1.16: Example of field reconstruction in the presence of fading and additive noise: ideal spatial field (top left); measured field (top right); field reconstructed with order $k = 10$ (bottom left) and $k = 20$ (bottom right).

1.7 Conclusions

In summary, the distributed projection algorithm proposed in this Chapter allows every node of the sensing network to converge to the same value achievable by a centralized network with a node having full access to all the measurements. The price paid is the iterative nature of the proposed algorithm. To minimize the error implicit in the use of a finite number of iterations, we have chosen the mixing matrix $\mathbf{W}$ in order to maximize the convergence rate of the proposed algorithm, for any signal subspace and network topology, compatible with the existence of a solution. The numerical examples shown in this Chapter refer to a signal subspace spanned by the low frequency components of the 2D Fourier or Polynomial basis, but the performance can be improved by making other choices, like wavelets, for example.

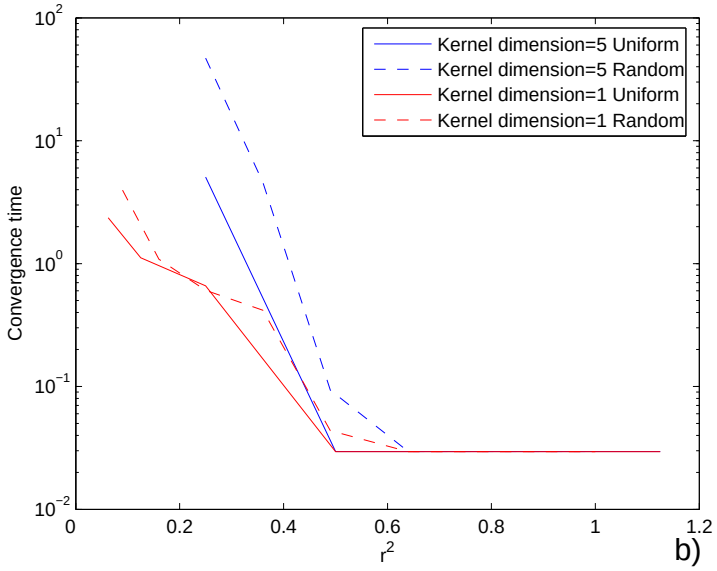

Figure 1.17: Minimum convergence time vs. number of neighbors, for uniform and random grids.

1.8 Appendix A

In this section we first prove the sufficiency of conditions C.1-C.3, then we focus on the necessity.

Sufficiency: Suppose that C.1-C.3 hold true. Conditions C.1 and C.2 imply, respectively:

$$\mathbf{W}^k \mathbf{P}_{\mathcal{R}(\mathbf{U})} = \mathbf{P}_{\mathcal{R}(\mathbf{U})} \tag{1.8.1}$$

and

$$\left(\mathbf{I} - \mathbf{P}_{\mathcal{R}(\mathbf{U})}\right) \mathbf{W} = \mathbf{W} \left(\mathbf{I} - \mathbf{P}_{\mathcal{R}(\mathbf{U})}\right). \tag{1.8.2}$$

Using (1.8.1) and (1.8.2), we can easily obtain the following chain of equalities:

$$\begin{aligned}
\mathbf{W}^k - \mathbf{P}_{\mathcal{R}(\mathbf{U})} &= \mathbf{W}^k \left(\mathbf{I} - \mathbf{P}_{\mathcal{R}(\mathbf{U})}\right) && (1.8.3)\\
&= \mathbf{W}^k \left(\mathbf{I} - \mathbf{P}_{\mathcal{R}(\mathbf{U})}\right)^k && (1.8.4)\\
&= \left[\mathbf{W} \left(\mathbf{I} - \mathbf{P}_{\mathcal{R}(\mathbf{U})}\right)\right]^k && (1.8.5)\\
&= \left(\mathbf{W} - \mathbf{P}_{\mathcal{R}(\mathbf{U})}\right)^k, && (1.8.6)
\end{aligned}$$

where: (1.8.3) follows from (1.8.1), (1.8.4) follows from the fact that $\mathbf{P}_{\mathcal{R}(\mathbf{U})}$ (and $\mathbf{I} - \mathbf{P}_{\mathcal{R}(\mathbf{U})}$) is a projection, and thus an idempotent matrix, i.e., $\mathbf{P}^k_{\mathcal{R}(\mathbf{U})} = \mathbf{P}_{\mathcal{R}(\mathbf{U})}$ for all $k \in \mathbb{N}_+$ [25], (1.8.5) follows from (1.8.2) and (1.8.6) follows from C.1.

According to (1.2.7), the asymptotic convergence of the dynamical system in (1.2.5) to the final vector $\mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[0]$ is guaranteed for *any* (fixed) initial condition $\mathbf{z}[0] \in \mathbb{R}^N$ if and only if

$$\lim_{k \to +\infty} \left(\mathbf{W}^k - \mathbf{P}_{\mathcal{R}(\mathbf{U})}\right) = \lim_{k \to +\infty} \left(\mathbf{W} - \mathbf{P}_{\mathcal{R}(\mathbf{U})}\right)^k = \mathbf{0}, \qquad (1.8.7)$$

where in the equality in (1.8.7) we used (1.8.6). Condition C.3 is necessary and sufficient for $\mathbf{W} - \mathbf{P}_{\mathcal{R}(\mathbf{U})}$ to be a convergent matrix [26], implying (1.8.7) to be hold. *Necessity:* Suppose that the limit in (1.2.7) exists. Invoking classical results on convergence matrices (see, e.g., [25, p. 630], [24]) we necessarily have:

$$\lim_{k \to +\infty} \mathbf{W}^k = \text{projection onto } \mathcal{N}(\mathbf{I} - \mathbf{W}) \text{ along } \mathcal{R}(\mathbf{I} - \mathbf{W}). \qquad (1.8.8)$$

Comparing (1.2.7) with (1.8.8) we infer that $\mathbf{P}_{\mathcal{R}(\mathbf{U})}$ in (1.2.7) must satisfy the following (necessary) conditions:

$$\mathcal{R}\left(\mathbf{P}_{\mathcal{R}(\mathbf{U})}\right) = \mathcal{N}\left(\mathbf{I} - \mathbf{W}\right) \quad \text{and} \quad \mathcal{N}\left(\mathbf{P}_{\mathcal{R}(\mathbf{U})}\right) = \mathcal{R}\left(\mathbf{I} - \mathbf{W}\right) \qquad (1.8.9)$$

or, equivalently

$$\begin{aligned}(\mathbf{I}-\mathbf{W})\,\mathbf{U}=\mathbf{0} \quad &\Leftrightarrow\ \mathbf{W}\mathbf{P}_{\mathcal{R}(\mathbf{U})}=\mathbf{P}_{\mathcal{R}(\mathbf{U})}\\ \mathbf{U}^T(\mathbf{I}-\mathbf{W})=\mathbf{0} \quad &\Leftrightarrow\ \mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{W}=\mathbf{P}_{\mathcal{R}(\mathbf{U})},\end{aligned} \tag{1.8.10}$$

which proves the necessity of conditions C.1 and C.2.

Using C.1, C.2 and (1.8.6), the limit in (1.2.7) can be written as in (1.8.7), implying the necessity of C.3 [26].

Given C.1-C.3, we prove now that the error vector $\mathbf{e}[k]=\mathbf{z}[k]-\mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[0]$ satisfies the dynamic equation in (1.2.8). First of all observe that the projection of vector $\mathbf{z}[k]$ onto $\mathcal{R}(\mathbf{U})$ is an invariant quantity for the dynamical system (1.2.5), i.e., for all k:

$$\mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[k]=\mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{W}\mathbf{z}[k-1]=\mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[k-1]=\ldots=\mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[0]. \tag{1.8.11}$$

Using (1.8.11) and (1.2.5), we obtain the following dynamics for the error vector:

$$\begin{aligned}\mathbf{e}[k+1] &= \mathbf{W}\mathbf{e}[k]+\mathbf{W}\mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[0]-\mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[0]\\ &= \mathbf{W}\mathbf{e}[k]-\mathbf{P}_{\mathcal{R}(\mathbf{U})}\left(\mathbf{z}[k]-\mathbf{P}_{\mathcal{R}(\mathbf{U})}\mathbf{z}[0]\right)=\left(\mathbf{W}-\mathbf{P}_{\mathcal{R}(\mathbf{U})}\right)\mathbf{e}[k],\end{aligned} \tag{1.8.12}$$

which completes the proof.

1.9 Appendix B

In this section we prove that the optimization problem (1.3.7) can be rewritten as a semi-definite programming.

Problem (1.3.7) is separable in the variables $\overline{\mathbf{L}}$ and ϵ, since the constraints in (1.3.7) depend only on $\overline{\mathbf{L}}$ or ϵ . Thus, in the following, we solve (1.3.7) by first minimizing the objective function over ϵ for a given feasible $\overline{\mathbf{L}}$, and then minimizing the resulting objective function over $\overline{\mathbf{L}}$.

Minimizing over ϵ: Denoting by $\left\{\lambda_{(i)}(\overline{\mathbf{I}}-\epsilon\overline{\mathbf{L}})\right\}$ and $\left\{\lambda_{(i)}(\overline{\mathbf{L}})\right\}$ the set of eigenvalues of $\overline{\mathbf{I}}-\epsilon\overline{\mathbf{L}}$ and $\overline{\mathbf{L}}$, respectively, arranged in increasing order [i.e., $\lambda_{(i)}(\cdot) \leq \lambda_{(i+1)}(\cdot)$, for all $i \in \{1,\ldots,N-r\}$], the objective function in (1.3.7) can be rewritten as:

$$\rho\left(\overline{\mathbf{I}}-\epsilon\overline{\mathbf{L}}\right) = \max_{k\in\{1,\ldots,N-r\}} \left\{\lambda_{(N-r)}(\overline{\mathbf{I}}-\epsilon\overline{\mathbf{L}}),\, -\lambda_{(1)}(\overline{\mathbf{I}}-\epsilon\overline{\mathbf{L}})\right\} \tag{1.9.1}$$

$$= \max_{k\in\{1,\ldots,N-r\}} \left\{1-\epsilon\lambda_{(1)}(\overline{\mathbf{L}}),\, \epsilon\lambda_{(N-r)}(\overline{\mathbf{L}})-1\right\}, \tag{1.9.2}$$

which is a piecewise-linear convex function in the variable $0 < \epsilon < \frac{2}{\lambda_i(\overline{\mathbf{L}})}$ [27]. It follows that, for any given $\overline{\mathbf{L}}$, the global minimum of (1.9.2) over $\epsilon \in (0,\, 2/\lambda_i(\overline{\mathbf{L}}))$ is achieved when $1-\epsilon\lambda_{(1)}(\overline{\mathbf{L}}) = \epsilon\lambda_{(N-r)}(\overline{\mathbf{L}})-1$ (recall that $0 < \lambda_{(1)}(\overline{\mathbf{L}}) \leq \lambda_{(N-r)}(\overline{\mathbf{L}})$, for any feasible $\overline{\mathbf{L}}$), implying the following optimal (feasible) value of ϵ:

$$\epsilon^\star = \frac{2}{\lambda_{(1)}\left(\overline{\mathbf{L}}\right)+\lambda_{(N-r)}\left(\overline{\mathbf{L}}\right)}. \tag{1.9.3}$$

Using (1.9.3), function in (1.9.2) can be rewritten as

$$\rho\left(\overline{\mathbf{I}}-\epsilon^\star\overline{\mathbf{L}}\right) = \frac{\lambda_{(N-r)}(\overline{\mathbf{L}})-\lambda_{(1)}(\overline{\mathbf{L}})}{\lambda_{(1)}(\overline{\mathbf{L}})+\lambda_{(1)}(\overline{\mathbf{L}})} \triangleq \frac{\kappa(\overline{\mathbf{L}})-1}{\kappa(\overline{\mathbf{L}})+1}, \tag{1.9.4}$$

where in the last equality we used $\overline{\mathbf{L}} \succ \mathbf{0}$ and introduced the condition number $\kappa(\overline{\mathbf{L}}) \triangleq \lambda_{(N-r)}(\overline{\mathbf{L}})/\lambda_{(1)}(\overline{\mathbf{L}})$ of matrix $\overline{\mathbf{L}}$. We can now find the optimal $\overline{\mathbf{L}}$ minimizing $\rho\left(\overline{\mathbf{I}}-\epsilon^\star\overline{\mathbf{L}}\right)$, under constraints (1.2.12)- (1.2.15).

Minimizing over $\overline{\mathbf{L}}$: Given the optimal $\epsilon^\star$ in (1.9.3) and the resulting objective function $\rho\left(\overline{\mathbf{I}}-\epsilon^\star\overline{\mathbf{L}}\right)$ in (1.9.4), the optimization problem (1.3.7) reduces to

$$\begin{aligned} &\underset{\overline{\mathbf{L}}}{\text{minimize}} \quad \kappa(\overline{\mathbf{L}}) \\ &\text{subject to} \quad \overline{\mathbf{L}} \succ \mathbf{0}, \quad \overline{\mathbf{L}} = \overline{\mathbf{L}}^T, \\ &\qquad\qquad \left[\mathbf{U}^\perp\overline{\mathbf{L}}\mathbf{U}^{\perp\mathbf{T}}\right]_{ij} = 0 \quad \forall i,j \in \mathcal{B}, \end{aligned} \tag{1.9.5}$$

where to write (1.9.5) from (1.3.7) and (1.9.4), we used the fact that $\rho\left(\overline{\mathbf{I}}-\epsilon^\star\overline{\mathbf{L}}\right)$ in (1.9.4) is an increasing function of $\kappa(\overline{\mathbf{L}})$. We convert now (1.9.5) into a convex SDP [27].

Writing the problem in epigraph form, we obtain:

$$\begin{array}{ll} \underset{\overline{\mathbf{L}},\,\gamma}{\text{minimize}} & \gamma \\ \text{subject to} & \dfrac{\lambda_{(N-r)}(\overline{\mathbf{L}})}{\lambda_{(1)}(\overline{\mathbf{L}})} \leq \gamma, \\ & \overline{\mathbf{L}} \succ \mathbf{0}, \quad \overline{\mathbf{L}} = \overline{\mathbf{L}}^T, \\ & \left[\mathbf{U}^{\perp}\overline{\mathbf{L}}\mathbf{U}^{\perp\mathbf{T}}\right]_{ij} = 0 \quad \forall i,j \in \mathcal{B}. \end{array} \tag{1.9.6}$$

For any given $\overline{\mathbf{L}} \succ \mathbf{0}$ and $\gamma > 0$, it is not difficult to prove the following equivalence:

$$\frac{\lambda_{(N-r)}(\overline{\mathbf{L}})}{\lambda_{(1)}(\overline{\mathbf{L}})} \leq \gamma \quad \Leftrightarrow \quad \exists \mu > 0 \,\text{such that}\, \mu\mathbf{I} \preceq \overline{\mathbf{L}} \preceq \gamma\mu\mathbf{I}. \tag{1.9.7}$$

Using (1.9.7), problem (1.9.6) can be rewritten as

$$\begin{array}{ll} \underset{\overline{\mathbf{L}},\,\gamma,\,\mu}{\text{minimize}} & \gamma \\ \text{subject to} & \mu\mathbf{I} \preceq \overline{\mathbf{L}} \preceq \gamma\mu\mathbf{I}, \quad \mu > 0, \\ & \overline{\mathbf{L}} = \overline{\mathbf{L}}^T, \\ & \left[\mathbf{U}^{\perp}\overline{\mathbf{L}}\mathbf{U}^{\perp\mathbf{T}}\right]_{ij} = 0 \quad \forall i,j \in \mathcal{B}. \end{array} \tag{1.9.8}$$

Introducing the following change of variables:

$$\widetilde{\mathbf{L}} = \overline{\mathbf{L}}\,\frac{1}{\mu} \qquad \widetilde{\mu} = \frac{1}{\mu}, \tag{1.9.9}$$

it is straightforward to see that (1.9.8) is equivalent to the following eigenvalue problem:

$$
\begin{array}{ll}
\text{minimize} & \gamma \\
\widetilde{\mathbf{L}},\, \gamma,\, \widetilde{\mu} & \\
\text{subject to} & \mathbf{I} \preceq \widetilde{\mathbf{L}} \preceq \gamma \mathbf{I}, \quad \widetilde{\mu} > 0, \\
& \widetilde{\mathbf{L}} = \widetilde{\mathbf{L}}^T, \\
& \left[\mathbf{U}^{\perp} \widetilde{\mathbf{L}} \mathbf{U}^{\perp \mathbf{T}} \right]_{ij} = 0 \quad \forall i, j \in \mathcal{B}.
\end{array}
\tag{1.9.10}
$$

In fact, if $\left(\overline{\mathbf{L}},\, \gamma,\, \mu\right)$ is a feasible point in (1.9.8), then $\left(\widetilde{\mathbf{L}},\, \gamma,\, \widetilde{\mu}\right)$, with $\widetilde{\mathbf{L}}$ and $\widetilde{\mu}$ defined in (1.9.9), is feasible in (1.9.10) with the *same* value of the objective function. The converse also holds true.

Using the fact that, at any optimum $\left(\widetilde{\mathbf{L}}^{\star},\, \gamma^{\star},\, \widetilde{\mu}^{\star}\right)$ in (1.9.10), $\widetilde{\mu}^{\star} > 0$, problem (1.9.10) can be rewritten as a SDP in standard form:

$$
\begin{array}{ll}
\text{minimize} & \gamma \\
\widetilde{\mathbf{L}},\, \gamma,\, \widetilde{\mu} & \\
\text{subject to} & \begin{bmatrix} \widetilde{\mathbf{L}} - \overline{\mathbf{I}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \gamma \overline{\mathbf{I}} - \widetilde{\mathbf{L}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widetilde{\mu} \overline{\mathbf{I}} \end{bmatrix} \succeq \mathbf{0}, \\
& \widetilde{\mathbf{L}} = \widetilde{\mathbf{L}}^T, \\
& \left[\mathbf{U}^{\perp} \widetilde{\mathbf{L}} \mathbf{U}^{\perp \mathbf{T}} \right]_{ij} = 0 \quad \forall i, j \in \mathcal{B}.
\end{array}
\tag{1.9.11}
$$

Once an optimal solution $\left(\widetilde{\mathbf{L}}^{\star},\, \gamma^{\star},\, \widetilde{\mu}^{\star}\right)$ to (1.9.11) is computed, the optimal original $\mathbf{L}^{\star}$ can be obtained through (1.9.9) and (1.2.12): $\mathbf{L}^{\star} = \widetilde{\mu}^{\star -1} \mathbf{U}^{\perp} \widetilde{\mathbf{L}}^{\star} \mathbf{U}^{\perp T}$.

1.10 Appendix C

We want to prove that

$$\|\boldsymbol{W}_j\boldsymbol{z} - \boldsymbol{U}\boldsymbol{U}^H\boldsymbol{z}\| \;\leq\; \rho(\boldsymbol{W}_j - \boldsymbol{U}\boldsymbol{U}^H)\|\boldsymbol{z} - \boldsymbol{U}\boldsymbol{U}^H\boldsymbol{z}\| \;\; \forall j \tag{1.10.1}$$

where $\mathbf{W}_j = (\mathbf{I} - \frac{\epsilon}{(j+1)^\eta}\mathbf{L})$ and $\rho(.)$ is the spectral radius. Decompose $\mathbf{W}_j$ through orthonormal eigenvectors as $\mathbf{W}_j = \boldsymbol{U}(j)\Lambda(j)\boldsymbol{U}(j)^H$. Hence,

$$\boldsymbol{z} = \boldsymbol{U}\boldsymbol{U}^H\boldsymbol{z} + \sum_{k=r}^{N} c_k(j)\boldsymbol{u}_k(j) \tag{1.10.2}$$

where $c_k(j) = \boldsymbol{u}_k(j)^H\boldsymbol{z}$,$k = r, \ldots, N$ and r is the dimension of the kernel of the matrix L. Then

$$\mathbf{W}_j\boldsymbol{z} = \boldsymbol{U}\boldsymbol{U}^H\boldsymbol{z} + \sum_{k=r}^{N} c_k(j)\lambda_k(\mathbf{W}_j)\boldsymbol{u}_k(j). \tag{1.10.3}$$

It follows that

$$\begin{aligned}
\|\boldsymbol{W}_j\boldsymbol{z} - \boldsymbol{U}\boldsymbol{U}^H\boldsymbol{z}\| &= \|\sum_{k=r}^{N} c_k(j)\lambda_k(\mathbf{W}_j)\boldsymbol{u}_k(j)\| \\
&\leq \rho(\boldsymbol{W}_j - \boldsymbol{U}\boldsymbol{U}^H)\|\sum_{k=r}^{N} c_k(j)\boldsymbol{u}_k(j)\| \\
&= \rho(\boldsymbol{W}_j - \boldsymbol{U}\boldsymbol{U}^H)\|\boldsymbol{z} - \boldsymbol{U}\boldsymbol{U}^H\boldsymbol{z}\|,
\end{aligned} \tag{1.10.4}$$

so that (1.10.1) is proved.

Chapter 2

Distributed estimation via Belief Propagation

2.1 Introduction

Considering the problem of distributed field estimation according to a stochastic approach the observations collected by a sensor network are modeled through Gaussian variables whose statistical dependency structure is captured by a Markov random field. In this setting the goal in designing the algorithms for wireless sensor networks is to guarantee that each node achieve in a completely distributed way the posterior probability of its measurement given the whole set of observations collected by the network so that it can compute for example the MAP or MMSE estimate of the monitored field value. There exist a lot of algorithms that were originally proposed to solve this kind of inference problems in other scientific contexts, where given a set of random variables,the problem of probabilistic inference can be cast as one of computing the posterior probability of a subset of variables, given the values of another subset (see for example [77]). In these problems we are given a joint distribution

to which a certain graphical model (such as Markov Random Fields, Bayesian Networks and Factor Graphs)is associated and these algorithms, based on the exchange of messages among "virtual" nodes that are neighbors in the statistical graph, are employed for efficient computation. A reverse thinking is required when it is applied to wireless sensor networks, intended to serve as a general framework for collaborative information processing and dissemination. A simple and effective approach for this novel application of great potential is proposed in different works where the real communication graph is treated as a Markov random field. In particular, each active node is taken as a vertex and there is an edge between two nodes when there is a feasible communication link between them. The key step lies in associating some "virtual" state variable(s) to each node, and building some statistical models indicating relationship among them, based on application characteristics and communication models. Assuming that in sensor network applications, the quantities of interest (for example, temperature or gas distribution) are often locally smooth, the true field's correlation could be modeled using a Gaussian Markov Random field where each node is connected only to the nodes that are spatially close (say, inside each coverage radius). In this way the message-passing algorithms proposed to solve the inference problem can be exploited to solve the field estimation in a completely distributed way,i.e. requiring that each node exchanges information only with its neighbors. The most popular algorithm to solve inference problems is the *Belief Propagation* (BP) algorithm [32], also known as the *Sum-product* algorithm [33]; the problem is that it is guaranteed to converge if and only if the graphical model considered contains no cycles(*tree-structured*). As an example, in [38] assuming a tree-structured graph a field of value is estimated by exploiting a Belief Propagation algorithm. Moreover, the

same algorithm is specialized to the particular case in which the wireless sensor network must reach consensus on the estimation of a single common observed variable. In theory, tree-based inference algorithms can be applied to any graph by clustering nodes so as to form a so-called *junction tree*, see e.g. [32] and [34]. However, in many cases of interest, the dimension of the clustered nodes is often quite large so that the computational cost of the algorithm is prohibitively. Moreover, in a sensor network the creation of clustered nodes implies the implementation of distributed algorithms based on a full knowledge of the network topology. *Loopy belief propagation*, i.e. the application of the Belief Propagation algorithm in a graph with arbitrary topology, has been studied for example in [35] and [37], where it is proved that in the case of Gaussian variables when the Belief Propagation algorithm converges it will gives the correct marginal posterior probabilities; on the other hand the conditional variances are in general incorrect. Plarre and Kumar [39] have proposed an extension of the loopy belief propagation algorithm based on a message passing algorithm that derives from the correspondence between recursive inference and Gaussian elimination. The implementation of the solution proposed in [39] implies the distributed extraction of a spanning tree from an arbitrary graph, or in other words the fact that each node must know the whole topology of the network. Since the conditional mean of a Gaussian inference problem can be interpreted as the solution of a linear system of equations, a wide range of iterative algorithms for solving inference problems derives from the solution of linear systems based on particular *matrix splittings*. Algorithms such as embedded polygons, embedded trees, embedded subgraphs [40], [41], [45] and [44] can be used to efficiently estimate the field distributively, but also in this case the main drawback is the fact that the nodes must be aware of the network topology in order

to organize themselves in trees or other topological structures.

In this Chapter, we provide a distributed algorithm for the estimation of a physical field in the particular case of a wireless sensor network organized in clusters, where nodes inside the same group observe the same field value. Our solution is based on a Belief Propagation algorithm and generalizes the solutions proposed in literature relative to the cases of a single variable estimation and field estimation with a single observation for each value using Belief propagation technique.

2.2 The setting

Given a set of random variables,the problem of probabilistic inference can be cast as one of computing the posterior probability of a subset of variables, given the values of another subset (see for example [77]). When the number of variables is large, inference requires integration over high dimensional spaces and can easily become intractable. In some cases, there are several conditional independence relationship between set of random variables. The collection of all such conditional independence relations gives rise to a factorization of the joint probability distribution into a product of functions, each of which depends on a subset of the variables. This factorization can significantly reduce the complexity of inference. In *graphical models*, all such conditional independence relations of a set of random variables are encoded in a graph. Each node in the graph represents a random variable and the independence relations are encoded in the edges. The graph can be directed(e.g, Bayesian networks) or undirected(e.g, Markov random fields). In this work we focus on undirected graphs. The Hammersley-Clifford theorem (e.g.,[61],[77],[32]) provides the connection between independence and factorization: a strictly positive probability distribution satisfies

all conditional independence relations implied by the graph, if and only if it factors according to the maximal cliques of the graph. When the underlying graph is singly connected (there is at most one path between any pair of nodes, i.e., it is a tree or a forest), efficient algorithms exist that solve the inference problem; see for example [62],[32].

Graphical models provide a framework for representing dependencies among the random variables of a statistical modelling problem and they constitute an elegant way to graphically represent the interaction among the random variables involved in a probabilistic system. A graphical model is a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes that that correspond to the random variables of a problem and $\mathcal{E}$ the set of edges that represent the dependencies among the variables. Graphical models can be either directed or undirected; directed graphical models are called *Bayesian networks* while in the other case they are known as *Markov random fields*. In Bayesian networks all the edges are considered to have a direction from parent to child denoting the conditional dependency among the corresponding variables; in the next, we consider only Markov Random fields without any loss of generality. Let $x_i, i \in 1, \ldots, N$ be random variables taking values in some discrete or continuous state space Λ, and form the random vector $\mathbf{x} = x_1, \ldots, x_N$ with configuration set $\Omega = \Lambda^N$. The joint probability distribution $p(\mathbf{x})$ exhibits a factorized form

$$p(\mathbf{x}) \propto \prod_{c \in C} f_c(\mathbf{x}_c), \tag{2.2.1}$$

where C consists of small index subsets c, the factor f_c depends only on the variable subset $\mathbf{x}_c = x_i, i \in c$ and $\prod_{c \in C} f_c(\mathbf{x}_c)$ is summable over Ω. If, in addition, the product

is positive ($\forall \mathbf{x} \in \Omega, p(\mathbf{x}) > 0$), then it can be written in exponential form

$$p(\mathbf{x}) = \frac{1}{Z} \exp\left(-\sum_{c} V_c(\mathbf{x}_c)\right), \tag{2.2.2}$$

this is the Gibbs distribution with *interaction potential* $V_c, c \in C$, *energy* $U = \sum_c V_c$ and *partition function* of parameters $Z = \sum_{\mathbf{x}\in\Omega} U(\mathbf{x})$. The interaction structure induced by the factorized form of the joint probability distribution is conveniently described by the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ that statisticians refer to as *independence graph*. It is important to specify that in the following we consider the terms node, vertex and sensor interchangeable. When i and j have an edge between them, i and j are neighbors denoted by $i \sim j$ (otherwise it is $i \vdash j$). The neighborhood function of a node i is the set of all other nodes having an edge with it, i.e.,

$$\mathcal{N}_{e(i)} = \{j \in V : j \neq i, (i, j) \in \mathcal{V}\} \tag{2.2.3}$$

The number of neighbors of a node i is called its degree, denoted by $\mathrm{Deg}(i)$. Let r_{ij} denotes the Euclidean edge length of (i, j). A node with a single edge i.e., its degree is one is known as a leaf and the corresponding edge as a leaf edge, otherwise it is known as an internal or interior edge. Given all these notations, the independence graph is defined as

$$i \sim j \Leftrightarrow \exists c \in C : i, j \subset c, \tag{2.2.4}$$

so that nodes i and j are neighbors if and only if the associated variables x_i and x_j appear simultaneously within the same factor f_c. From all these definitions, it follows that C is the set of *cliques* of the graph $\mathcal{G}$; a clique is a subset where for each pair of nodes there exist a link. A Markov Random field satisfies special conditional independence properties. A simple example is the first-order auto-regressive process, where the conditional independence of the observations is based on causality. However, a

spatial random field has a far richer set of conditional independencies, requiring a more general definition. The independence graph conveys the key probabilistic information by absent edges: if i and j are not neighbors, the joint probability distribution $p(\mathbf{x})$ can be split into two parts respectively independent from x_i and x_j so that the two random variables are independent given the others (pairwise-Markov property). We can express this property in this way

$$x_i \perp x_j | \mathbf{x}_{-ij} \iff (i,j) \text{ not in } \mathcal{E} \tag{2.2.5}$$

where $\perp$ denotes the conditional independence relation. Given a set $\mathbf{a} \subset \mathcal{V}$ of nodes, $p(\mathbf{x})$ splits into two parts

$$p(\mathbf{x}) \propto \prod_{c: c \bigcap \mathbf{a} \neq \mathbf{0}} f_c(\mathbf{x}_c) \prod_{c: c \bigcap \mathbf{a} = \mathbf{0}} f_c(\mathbf{x}_c) \tag{2.2.6}$$

where the second factor does not depend on $\mathbf{x}_a$. As a consequence $p_{\mathbf{x}_a|\mathbf{x}_{\mathcal{V}-a}}$ reduces to $p_{\mathbf{x}_a|\mathbf{x}_{\mathcal{N}_{e(a)}}}$; this is the local-Markov property, that we can express as it follows

$$x_i \perp \mathbf{x}_{-(i,\mathcal{N}_{e(i)})} | \mathbf{x}_{\mathcal{N}_{e(i)}} \tag{2.2.7}$$

If A,B and C are disjoint sets, with A and B non empty, and the set C separates A and B, i.e., on removing the nodes in C from the graph, nodes in A are no longer connected to the nodes in B, the global Markov property can be formulated as

$$\mathbf{x}_A \perp \mathbf{x}_B | \mathbf{x}_C, \tag{2.2.8}$$

Thus, in (2.2.7), the local Markov property states that the conditional distribution at a node in the graph given the observations at its neighbors is independent of the rest of the network. By the global Markov property in (2.2.13), all the connected components of a dependency graph are independent. It can be shown that the three

Markov properties are equivalent for strictly positive distributions. If $\mathcal{G}$ is the graph that represents a Markov random field the corresponding joint probability density $p(\mathbf{x})$ must satisfy the Markov properties imposed by the topology of $\mathcal{G}$. Conversely, if a strictly positive distribution $p(\mathbf{x})$ fulfills one of these Markov properties with respect to graph $\mathcal{G}$ then $p(\mathbf{x})$ is a MRF on G and $p(\mathbf{x})$ is a Gibbs distribution. This equivalence constitutes the *Hammersley-Clifford theorem.*

A special case of Markov Random field is the Gaussian Markov Random field; as previously mentioned, in Gaussian graphical models the problem of probabilistic inference is much less complicated because it reduces to find the correct posterior mean and covariance. A Gaussian Markov Random field is a stochastic process given by an unobserved $\mathbb{R}^N$ valued state vector $\mathbf{x} \sim \mathcal{N}(0, \mathbf{\Sigma}^{-1})$, with probability density function $p(\mathbf{x}) \propto \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{\Sigma}\mathbf{x}\right)$ where $\mathbf{\Sigma} = \mathbf{\Sigma}^t > \mathbf{0}$.A common approach to formulating a Gaussian Markov Random field is to specify the dependency graph through a neighborhood rule and then to specify the correlation function between these neighbors. Thus, in a Gaussian Markov Random field, local characteristics completely determine the joint distribution of the Gaussian field. The inverse of the covariance matrix of a Gaussian Markov Random field is known as the potential matrix or the precision matrix or the information matrix. The non zero elements of the precision matrix $\mathbf{A} = \mathbf{\Sigma}^{-1}$ are in one to one correspondence with the edges of its graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ in the sense that

$$i \vdash j \iff \mathbf{A}(i,j) = 0, \forall i, j \in \mathcal{V}, i \neq j. \tag{2.2.9}$$

This relationship between the precision matrix and the graph associated to a Gaussian Markov Random field is illustrated in Figure 2.1.

In practice a Gaussian Markov field is often defined simply by its quadratic energy

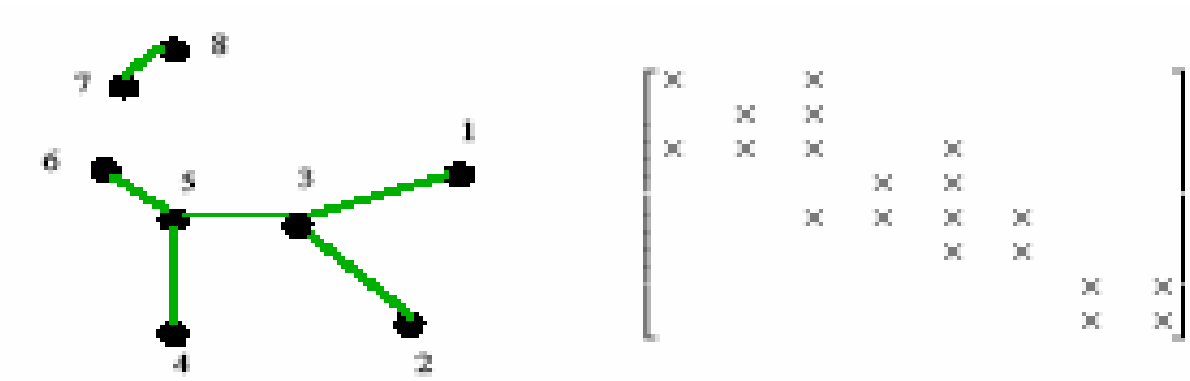

Figure 2.1: Dependency graph and potential matrix of a GMRF.

function

$$U(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b} = \sum_{ij} a_{ij}x_i x_j + \sum_i (\frac{a_{ii}}{2}x_i - b_i)x_i, \tag{2.2.10}$$

with $\mathbf{b} \in \mathbb{R}^N$. Any conditional distribution is Gaussian and can be explicitly written down using adequate block partitioning of $\mathbf{A}$ and $\mathbf{b}$, so that all Markovian properties can then be directly deduced from this. Site-wise conditional distributions in particular turn out to be

$$p(x_i|\mathbf{x}_{-i}) = \mathcal{N}(\frac{1}{a_{ii}}(b_i - \sum_{j\neq i} a_{ij}x_j), a_{ii}^{-1}), \tag{2.2.11}$$

where according to the pairwise-Markov property $(i,j) \in \mathcal{E} \Leftrightarrow a_{ij} = a_{ji} \neq 0$;from (2.2.5) and (2.2.11), it follows that

$$p(x_i|\mathbf{x}_{-i}) = \mathcal{N}(\frac{1}{a_{ii}}(b_i - \sum_{j\in\mathcal{N}_{e(i)}} a_{ij}x_j), a_{ii}^{-1}), \tag{2.2.12}$$

or in words, each node is independent of the others conditionally to its neighbors.

This simple correspondence between the conditional independence of the Gaussian Markov Random field and the zero structure of its precision matrix is not evident in the covariance matrix, which is generally a completely dense matrix. Therefore, it is

easier to evaluate the joint distribution of the Gaussian Markov Random field through the precision matrix. In practice, however, estimates of the covariance matrix are easier to obtain through the empirical observations. Therefore, it is desirable to have the joint distribution in terms of coefficients of the covariance matrix. An explicit expression between the coefficients of the covariance and the precision matrix and also an expression for the determinant of the precision matrix are achieved for a particular case of dependency graph in [121]. This special case of the dependency graph is the acyclic or a loop-free graph. Here, the neighbors of a node are not themselves neighbors. The joint distribution is somewhat easier to evaluate in this case. We note that an acyclic graph with at least one edge, always has a leaf ,i.e., it has a node with degree one and has at most $N-1$ edges in a N nodes graph. The covariance matrix $\mathbf{\Sigma}$ of a Gaussian Markov Random field satisfies some special properties. For instance, consider the cross covariance between the neighbors of a node, i.e., nodes that are two hops away in an acyclic undirected graph. By the global Markov property we have, for some $i \in \mathcal{V}$,$\text{Deg}(i) \geq 2$,$j, k \in \mathcal{N}_{e(i)}$,$j \neq k$,

$$\mathbf{\Sigma}_{(j,k)} = \frac{\mathbf{\Sigma}_{(i,j)}\mathbf{\Sigma}_{(i,k)}}{\mathbf{\Sigma}_{(i,i)}} \tag{2.2.13}$$

We can similarly find an expression for the covariance between any two nodes of the Gaussian Markov Random field. Thus, the covariance matrix of a Gaussian Markov Random field with acyclic dependency can be expressed solely in terms of the auto covariance of the nodes and the cross covariance between the neighbors of the dependency graph.

In general, we assume that sensors collect samples from a Gaussian Markov Random field, that is modelled as previously described through a graphical approach, in which a dependency graph specifies the stochastic dependence between different

sensor observations. This dependency graph can have different degrees and can even be fully connected. The algorithms used to solve inference problems, for example Belief Propagation or Embedded Trees algorithms, are based on exchanging messages among nodes according to the topology dictated by the statistical model graph. Because our goal is to adapt these algorithms to solve distributed estimation problems in wireless sensor networks, the statistical model graph that rules the exchange of the messages between sensors must be chosen not only according to the statistical characteristics of the applications but also and above all according to the real communication graph. In other words, the statistical model must be supported by the physical graph, i.e. the links of the statistical model must coincide with the links of the physical model or be a subset of them. In sensor networks estimation applications, graphical models should balance the trade-off between accurately capturing the correlation structure of the quantities being measured and supporting energy efficient distributed algorithms. In many sensor network estimation applications, the quantities of interest, such as temperature, wind speed, or concentration of some substance, are often locally smooth. Such quantities can be effectively modeled by loopy, locally connected graphical models, in which only spatially neighboring nodes are connected by edges (by analogy, locally smooth images have been successfully modeled using Gaussian Markov Random fields in which graphical models connect only adjacent pixels). Typically statistical spatial interactions are based on proximity, where the choice of edges to include being determined by the local point configuration according to some specified rule [122]. With a regular lattice structure(e.g., in image processing, Ising model), a fixed set of neighbors can be specified in a straight-forward manner [123]. However, the situation is more complicated for arbitrary placed nodes. In

[121], the nearest neighbor graph (NNG), which is the simplest proximity graph, is considered. The nearest neighbor relation has been used in several areas of applied science, including the social sciences, geography and ecology, where proximity data is often important. The nearest neighbor function of a node $i \in \mathcal{V}$ is defined as

$$\mathrm{nn}(i) \doteq \arg \min_{j \in \mathcal{V}, j \neq i} \mathrm{dist}(i, j), \tag{2.2.14}$$

where dist(.,.) is the Euclidean distance. The inter-point distances are unique with probability one, for uniform and Poisson point sets under consideration here. Therefore, nn(i) is well-defined function almost surely. The nearest-neighbor undirect graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is given by

$$(i, j) \in \mathcal{E} \Leftrightarrow i = \mathrm{nn}(j) \text{ or } j = \mathrm{nn}(i) \tag{2.2.15}$$

The nearest neighbor graph has a number of important properties; it is acyclic with a maximum node degree of six [124]. In [40], another solution in the construction of the graphical model based on proximity among nodes is introduced; a spatial *triangulation* of the sensor locations induces a graphical model that balance the trade-off between accuracy and efficiency in the distributed algorithms implementation. A triangulated graphical model assumes that a sensor's measurement is uncorrelated with the rest of the network given the close-by measurements. This is clearly reasonable for smoothly varying quantities. The *Delaunay triangulation* [125]induces a graphical model with some additional attractive properties. First, the Delaunay triangulation links together the closest neighbors in the graph, in the sense that the circumcircle of each triangle does not contain any points of the triangulation. Second, the Delaunay triangulation can be established in a distributed fashion [126], and for this reason is successfully used as an overlay topology in the networking field. In general, for the non-zero partial

correlation between two connected sensors, a decreasing function of the Euclidean distance among nodes is adopted.

For simplicity (only) we suppose that each node in graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ corresponds to a component of $\mathbf{x}$, and not to a sub-vector as in the more general case. To each x_i corresponds a noisy observation y_i collected by a sensor of the wireless network such that the observation vector $\mathbf{y}$, according to a linear observation model, satisfies

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{v} \tag{2.2.16}$$

with $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$. We append to the graph $\mathcal{G}$, N more nodes indexed by $y_1, \ldots, y_N$ and edges (x_i, y_i) for $i = 1, \ldots, N$. We call this new graph as $\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$. In $\bar{\mathcal{G}}$, since each $\mathbf{y}_i$ is connected only to x_i, it implies that the random variables $\{\mathbf{y}_i\}_{i=1}^N$ are conditionally independent given $\mathbf{x}$, which as a consequence means that $\mathbf{C}$ and $\mathbf{R}$ are diagonal matrices. We call this graph as an Hidden Gaussian Markov Random field (HGMRF). Figure 2.2 shows an example of such a graphical model, where the observed nodes, the nodes corresponding to $\mathbf{y}_i$ for $1 \leq i \leq N$, are colored black. The goal of inference problem is to determine the conditional marginals $p(x_i|\mathbf{y})$, i.e., the posterior probability of each x_i given the observations, when $\mathbf{\Sigma}$, $\mathbf{C}$ and $\mathbf{R}$ are given. If we suppose that, after information processing and dissemination, each node obtains the posterior distribution $p(x_i|\mathbf{y})$, then various estimates such as those corresponding to MAP or MMSE criteria can be easily computed at each node of the wireless sensor network. Let $\mathbf{P} = \mathbf{\Sigma}^{-1}$. As in standard, we consider $\mathbf{\Sigma} = \mathbf{P}^{-1}$ rather than $\mathbf{P}$ as given, since the joint probability distribution of $\mathbf{x}$ is given by the coefficients of $\mathbf{\Sigma}$. Since the joint distribution is gaussian, the posterior distribution is also Gaussian, and it suffices to determine the posterior mean $\hat{\mathbf{x}}$ and covariance $\hat{\mathbf{P}}$. It is known that $\hat{\mathbf{x}}$ and

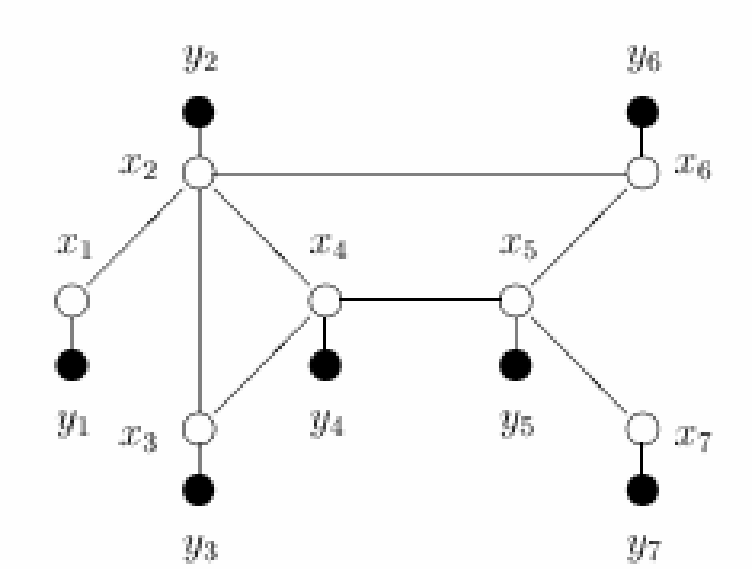

Figure 2.2: Example of a Gaussian Markov Random field defined on a loopy graph.

$\hat{\mathbf{P}}$ satisfies

$$\hat{\mathbf{x}} = \hat{\mathbf{P}}\mathbf{C}^T\mathbf{R}^{-1}\mathbf{y} \tag{2.2.17}$$

$$\hat{\mathbf{P}} = [\mathbf{P}^{-1} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}]^{-1} \tag{2.2.18}$$

Note that, since the conditional error variances are the diagonal elements of $\hat{\mathbf{P}}$, solving (2.2.17) and (2.2.18) is, in fact, more general than solving the inference problem as stated above, because the complete posterior covariance is computed, and not just its diagonal elements. Here we are interested only in computing the diagonal elements of $\hat{\mathbf{P}}$.

Solving this inference problem in a wireless sensor network is equivalent to estimate a spatial field associated to some quantities of interest, such as temperature or wind speed, under the hypothesis that the correlation among the measurements can be modeled as a Gaussian Markov Random field. As described in the introduction at the beginning of this chapter, several algorithms can be adopted to solve this estimation

problem in a completely distributed way;however, in this work, we focus on the Belief propagation algorithm.

In the first part of the next section, we consider a wireless sensor network where all nodes estimate the same variable and must reach the consensus on a common estimate of it; in [38], a distributed solution based on Belief Propagation algorithm is proposed. After, as described in [38] and [39], we illustrate the generalization of this solution to the case in which each sensor must estimate a different variable of the common Gaussian Markov random field. Finally, we consider a wireless sensor network where nodes are organized in clusters and each of them collect measurements relative to different variables; in this scenario, we propose a distributed solution based on Belief Propagation algorithm thank to which sensors in the same cluster reach consensus on the common measured variable but each cluster converges to a own value.

2.2.1 Field Estimation via Belief Propagation

For ease of exposition, we will focus on Markov random fields with only pairwise interactions, since MRF with higher order cliques(i.e., fully connected subgraphs) can always be converted to an equivalent pairwise MRF. A pairwise Markov Random field is an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with maximum clique of size two, where each node $i \in \mathcal{V}$ is associated with a random variable (or a more general random vector). The Hammersley-Clifford theorem dictates that, if a joint distribution can be represented by a pairwise MRF, it should admit the following form (and viceversa)

$$p(\mathbf{x}/\mathbf{y}) = \prod_{(i,j)\in\mathcal{E}} \Psi_{ij}(x_i, x_j) \prod_{i\in\mathcal{V}} \phi(x_i, \mathbf{y}_i) \tag{2.2.19}$$

for a set of single node functions $\{\Phi(x_i, \mathbf{y}_i)\}$ (called local functions, defined for each $i \in \mathcal{V}$, and a set of pairwise functions $\{\Psi_{ij}(x_i, x_j)\}$ (called compatibility functions,

defined for each $(i,j) \in \mathcal{E}$, and a normalization factor Z (called partition function in physics). The essence of Belief Propagation algorithm is the message-passing rule and belief-updating rule. The message from node i to j at the n-th iteration is a function of x_j, defined as

$$m_{ij}^n(x_j) = \sum_{x_i} \Psi_{ij}(x_i, x_j)\phi_i(x_i, \mathbf{y}_i) \prod_{k \in N_{e(i)}/j} m_{ki}^{n-1}(x_i) \tag{2.2.20}$$

where $N_{e(i)}$ is the set of neighbors of node i. The sum in (2.2.20) is replaced with the integral when continuous random variables are considered. This message is often normalized for numerical stabilization though not necessarily. Roughly speaking, it represents the current belief (approximated posterior probability distribution) that node i has about x_j, given its own observations and received messages from other parts of the graph in the last round. The belief node i has about its own variable is updated as (with normalization factor α)

$$b_i^n(x_i) = \alpha\phi_i(x_i, \mathbf{y}_i) \prod_{k \in N_{e(i)}} m_{ki}^{n-1}(x_i). \tag{2.2.21}$$

Usually the messages are initialized with unbiased (constant) ones to trigger the iteration. If the computing graph is a tree, it is known that the Belief Propagation algorithm is guaranteed to converge to the true marginals, i.e., $b_i^n(x_i) \to p(x_i) = \sum_{\mathbf{x}/x_i} p(\mathbf{x})$. Belief Propagation can be naturally applied on graphs with cycles as well. In this scenario, the iteration is typically stopped when improvement on beliefs is marginal, or sufficiently many numbers of iteration have passed. However, little is known about the convergence and correctness of Belief Propagation on loopy graphs, though its effectiveness has been verified through experiments in various areas. The extended message passing algorithm proposed in [39] is guaranteed to converge in finite time for any parametrization and for any graph. A drawback of this new

algorithm is that each node needs information about the structure of the graph, while to apply traditional Belief Propagation, each sensor need to know only its neighbors.

Gaussian distribution is a widely adopted assumption in theoretical studies. It is a good approximation of practical situations in many scenarios of interest, while amenable to analysis and often can provide useful insights. The following result is useful for message passing with Gaussian distribution. Let $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ be a Gaussian random vector with mean $\boldsymbol{\mu}$ and positive definite covariance $\mathbf{\Sigma}$. One can define a new set of parameters $(\mathbf{\Theta}, \mathbf{\Lambda})$ by $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}\boldsymbol{\mu}$, $\mathbf{\Lambda} = \mathbf{\Sigma}^{-1}$, and alternatively denote $\mathbf{X} \sim \mathcal{N}^{-1}(\mathbf{\Theta}, \mathbf{\Lambda})$. Let $p_1(\mathbf{X}) = \mathcal{N}^{-1}(\mathbf{\Theta}_1, \mathbf{\Lambda}_1)$ and $p_2(\mathbf{X}) = \mathcal{N}^{-1}(\mathbf{\Theta}_2, \mathbf{\Lambda}_2)$ be two different distributions on the same Gaussian random vector $\mathbf{X}$, and consider the product density $p_{12}(\mathbf{X}) = \alpha p_1(\mathbf{X}) p_2(\mathbf{X})$. Then $p_{12}(\mathbf{X}) = \mathcal{N}^{-1}(\mathbf{\Theta}_{12}, \mathbf{\Lambda}_{12})$ with $\mathbf{\Theta}_{12} = \mathbf{\Theta}_1 + \mathbf{\Theta}_2$ and $\mathbf{\Lambda}_{12} = \mathbf{\Lambda}_1 + \mathbf{\Lambda}_2$. Similarly, the quotient $p_1(\mathbf{X})/p_2(\mathbf{X})$ produces an exponential quadratic form with parameters $\mathbf{\Theta}_1 - \mathbf{\Theta}_2, \mathbf{\Lambda}_1 - \mathbf{\Lambda}_2$. However, this quotient will define a valid probability density only if $\mathbf{\Lambda}_1 - \mathbf{\Lambda}_2$ is positive definite.

Assume the variable to be estimated is $\mathbf{x} \sim \mathcal{N}^{-1}(\boldsymbol{\mu}_s/\sigma_s^2, 1/\sigma_s^2)$. Each sensor makes a noisy linear observation

$$\mathbf{y}_i = \mathbf{H}_i \mathbf{x} + \mathbf{n}_i \quad i = 1, \ldots, N \tag{2.2.22}$$

where for generality we consider a vector observation of $\mathbf{y}_i \in \mathbb{R}^{d_i}$ for each sensor, channel gain matrix $\mathbf{H}_i$ is assumed known, and noise $\mathbf{n}_i$ is Gaussian with zero mean and covariance matrix $\mathbf{R}_i$. It is easy to derive that, the conditional probability $f_i(\mathbf{y}_i|\mathbf{x})$, viewed as a function of $\mathbf{x}$, assumes the form of

$$\mathcal{N}^{-1}(\mathbf{H}_i^T \mathbf{R}^{-1} \mathbf{y}_i, \mathbf{H}_i^T \mathbf{R}^{-1} \mathbf{H}_i) \tag{2.2.23}$$

up to some scaling constant. In the consensus estimation problem, the state variable

associated with each node is the common source $\mathbf{x}$. Since this variable is the same for all nodes, considering (2.2.19) we have the following instantiation of the belief Propagation algorithm

$$\phi_i(\mathbf{x}) = f_i(\mathbf{y}_i|\mathbf{x})(p(\mathbf{x}))^{1/N} \tag{2.2.24}$$

and

$$\Psi_{ij}(x_i, x_j) = 1(x_i = x_j). \tag{2.2.25}$$

In other words, we impose joint distribution of the form $p(\mathbf{X}_v = \mathbf{x}_v, v \in \mathcal{V}) = 1\{x_1 = x_2 = \cdots = x_n = X\}p(\mathbf{x})\prod_{i=1}^{N} f_i(\mathbf{y}_i|\mathbf{x})$ with $1(.)$ denoting the indicator function. The message passing rule is thus concretized as

$$\log\left(\mathbf{m}_{ij}^n(x_i)\right) = \log\left(\phi_i(x_i)\right) + \sum_{k \in N_{e(i)}/j} \log\left(\mathbf{m}_{ki}^{n-1}(x_i)\right) \tag{2.2.26}$$

which reveals a simple linear relationship (without convolution) for messages between successive rounds due to the special form of compatibility functions. Clearly the messages and node beliefs in Belief Propagation algorithms are all Gaussian distributed. Assuming that

$$m_{ij}^n(\mathbf{x}) \sim \mathcal{N}^{-1}(\boldsymbol{\mu}_{ij}^n, \mathbf{V}_{ij}^n), \tag{2.2.27}$$

and

$$b_i^n(\mathbf{x}) \sim \mathcal{N}^{-1}(\mathbf{q}_i^n, \mathbf{W}_i^n), \tag{2.2.28}$$

we have the following message updating and belief updating rules:

$$\boldsymbol{\mu}_{ij}^n = \boldsymbol{\mu}_s/(N\sigma_s^2) + \mathbf{H}_i^T\mathbf{R}^{-1}\mathbf{y}_i + \sum_{k \in N_{e(i)}/j} \boldsymbol{\mu}_{ki}^{n-1}, \tag{2.2.29}$$

$$\mathbf{V}_{ij}^n = 1/(N\sigma_s^2) + \mathbf{H}_i^T\mathbf{R}^{-1}\mathbf{H}_i + \sum_{k \in N_{e(i)}/j} \mathbf{V}_{ki}^{n-1} \tag{2.2.30}$$

and

$$\mathbf{q}_i^n = \boldsymbol{\mu}_s/(N\sigma_s^2) + \mathbf{H}_i^T\mathbf{R}^{-1}\mathbf{y}_i + \sum_{k\in N_{e(i)}/j} \boldsymbol{\mu}_{ki}^n, \tag{2.2.31}$$

$$\mathbf{W}_i^n = 1/(N\sigma_s^2) + \mathbf{H}_i^T\mathbf{R}^{-1}\mathbf{H}_i + \sum_{k\in N_{e(i)}/j} \mathbf{V}_{ki}^n \tag{2.2.32}$$

with $\boldsymbol{\mu}_{ij}^0$ and $\mathbf{V}_{ij}^0$ initialized with zero for all i,j. Noting the similarity of the previous expressions, the implementation of the Belief propagation algorithm in a wireless setting can exploit the broadcast nature of the medium. Instead of sending messages of this form from each node i to its neighbors, we let node i broadcasts its belief to them with the following modified form:

$$\mathbf{q}_i^n = \boldsymbol{\mu}_s/(N\sigma_s^2) + \mathbf{H}_i^T\mathbf{R}^{-1}\mathbf{y}_i + \sum_{k\in N_{e(i)}/j} \boldsymbol{\mu}_{ki}^{n-1}, \tag{2.2.33}$$

$$\mathbf{W}_i^n = 1/(N\sigma_s^2) + \mathbf{H}_i^T\mathbf{R}^{-1}\mathbf{H}_i + \sum_{k\in N_{e(i)}/j} \mathbf{V}_{ki}^{n-1}. \tag{2.2.34}$$

Meanwhile, it calculates and stores its intended messages for all $j \in N_{e(i)}$ to facilitate processing in the next round:

$$\boldsymbol{\mu}_{ij}^n = \mathbf{q}_i^n - \boldsymbol{\mu}_{ji}^{n-1}, \tag{2.2.35}$$

$$\mathbf{V}_{ij}^n = \mathbf{W}_i^n - \mathbf{V}_{ji}^{n-1}. \tag{2.2.36}$$

On the other hand, upon receiving $\mathbf{q}_j^n$ and $\mathbf{W}_j^n$ from some $j \in N_{e(i)}$, node i figures out the true messages from j as

$$\boldsymbol{\mu}_{ji}^n = \mathbf{q}_j^n - \boldsymbol{\mu}_{ij}^{n-1}, \tag{2.2.37}$$

$$\mathbf{V}_{ji}^n = \mathbf{W}_j^n - \mathbf{V}_{ij}^{n-1}, \tag{2.2.38}$$

and also store them for processing in the next round. When a node i collects all broadcast from its neighbors and figures out their intended messages, it can form its

own broadcast messages for next iteration. Again $\boldsymbol{\mu}_{ij}^0$ and $\mathbf{V}_{ij}^0$ are initialized with zero for all i,j. In practice node broadcasting needs to be coordinated with some MAC schemes. At convergence, the MAP estimate of the common variable $\mathbf{x}$ is easily computable at each sensor and is given by

$$\hat{\mathbf{x}} = \mathbf{W}_i^{-1}\mathbf{q}_i, \tag{2.2.39}$$

where $\mathbf{W}_i$ and $\mathbf{q}_i$ are the values available to each node at last iteration of the algorithm.

We have discussed the Belief propagation algorithm for consensus estimation of a single Gaussian source; this can be readily extended to multiple independent sources (variables) by treating $\mathbf{x}$ as a Gaussian vector. In the following, we consider the application of field gathering where $\mathbf{x}$ is a Gaussian Markov Random field and each node only observes a spatial component x_i of it. In this scenario, x_i associated with each node are not identical but nonetheless correlated through a joint distribution. Instead of achieving a common estimate at each node as previously discussed, here we intend to apply the Belief Propagation algorithm to improve the estimate at each node through collecting useful information from other parts of the network. Here we consider a good approximation for the underlying random field. Assuming that a spanning tree is formed among the distributed nodes, we only consider the pairwise interaction among x_i associated with each node. In other words, we ignore the correlation among nodes that are not direct neighbors on the spanning tree. In this setting, we have [43] [36]

$$p(\mathbf{x},\mathbf{y}) = \frac{\prod_{(i,j)\in\mathcal{E}} p_{ij}(x_i,x_j)}{\prod_{i\in\mathcal{V}} p_i(x_i)^{N(i)-1}} \prod_{i\in\mathcal{V}} f_i(\mathbf{y}_i/x_i). \tag{2.2.40}$$

where

$$p_i(x_i) = \mathcal{N}^{-1}(\boldsymbol{\mu}_s/\sigma_s^2, 1/\sigma_s^2), \tag{2.2.41}$$

and

$$p_{ij}(x_i, x_j) = \mathcal{N}^{-1}(\mathbf{C}_{ij}\boldsymbol{\mu}_s[1,1]^T, \mathbf{C}_{ij}) \tag{2.2.42}$$

with $\mathbf{C}_{ij}$ equal to the covariance matrix; $f_i(\mathbf{y}_i, x_i)$, viewed as a function of x_i, assumes the form in (2.2.23). Comparing (2.2.40) with (2.2.19) reveals

$$\phi_i(x_i) = \mathcal{N}^{-1}(\boldsymbol{\mu}_i, \mathbf{V}_i) \tag{2.2.43}$$

with

$$\boldsymbol{\mu}_i = \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{y}_i + (1 - N_{e(i)})\boldsymbol{\mu}_s/\sigma_s^2 \tag{2.2.44}$$

$$\mathbf{V}_i = \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i + (1 - N_{e(i)})/\sigma_s^2 \tag{2.2.45}$$

and

$$\Psi_{ij}(x_i, x_j) = \mathcal{N}^{-1}(\mathbf{C}_{ij}\boldsymbol{\mu}_s[1,1]^T, \mathbf{C}_{ij}). \tag{2.2.46}$$

After some manipulation, we have the following message updating and belief updating rules

$$\boldsymbol{\mu}_{ij}^n = \frac{\rho_{ij}(\boldsymbol{\mu}_i + \sum_{k\in N_{e(i)/j}} \boldsymbol{\mu}_{ki}^{n-1} - \rho_{ij}\boldsymbol{\mu}_s/\sigma_s^2(1-\rho_{ij}^2))}{(1+\sigma_s^2(1-\rho_{ij}^2)(V_i + \sum_{k\in N_{e(i)/j}} \mathbf{V}_{ki}^{n-1}))} + \rho_{ij}\boldsymbol{\mu}_s/\sigma_s^2(1-\rho_{ij}^2) \tag{2.2.47}$$

$$\mathbf{V}_{ij}^n = \frac{(\mathbf{V}_i + \sum_{k\in N_{e(i)/j}} \mathbf{V}_{ki}^{n-1} + 1/\sigma_s^2)}{(1+\sigma_s^2(1-\rho_{ij}^2)(V_i + \sum_{k\in N_{e(i)/j}} \mathbf{V}_{ki}^{n-1}))} + \rho_{ij}\boldsymbol{\mu}_s/\sigma_s^2(1-\rho_{ij}^2) \tag{2.2.48}$$

and

$$\mathbf{q}_i^n = \boldsymbol{\mu}_i + \sum_{k\in N_{e(i)}} \boldsymbol{\mu}_{ki}^n, \tag{2.2.49}$$

$$\mathbf{W}_i^n = \mathbf{V}_i + \sum_{k\in N_{e(i)}} \mathbf{V}_{ki}^n \tag{2.2.50}$$

with $\boldsymbol{\mu}_{ij}^0$ and $\mathbf{V}_{ij}^0$ initialized with zero for all i,j. At convergence each sensor computes the MAP estimate of its observed variable according to

$$x_i = \mathbf{W}_i^{-1}\mathbf{q}_i \quad \forall i \tag{2.2.51}$$

where $\mathbf{W}_i$ and $\mathbf{q}_i$ are the values available to each node at last iteration of the algorithm.

2.3 Field Estimation via Belief Propagation and multiple observations

The field of values to be monitored is characterized by a spatial structure such that the monitoring area can be divided in different subregions each one of them corresponds to a given field value. According to this we assume that some distributed clustering algorithm has organized the network in clusters, where each cluster is composed of sensors that measure the same value of field. Also in this case we assume a linear observation model such that the measurement collected by the j-th sensor in the i-th cluster is given by

$$\mathbf{y}_{ij} = \mathbf{H}_{ij}x_i + \mathbf{n}_{ij} \quad i = 1,\ldots,N \text{ and } j = 1,\ldots,M_i \tag{2.3.1}$$

where N is the number of clusters or equivalently the number of variables to be estimated, M_i the number of sensors in the i-th cluster and $\mathbf{n}_{ij}$ is Gaussian with zero mean and covariance matrix $\mathbf{R}_{ij}$.In each cluster, sensors must reach consensus on the MAP estimation of the common field value; let $\hat{x}_{ij}$ denote the value achieved at convergence by the j-th sensor in the i-th cluster, our goal is to obtain in a totally

decentralized way

$$\hat{x}_{ij} = \underset{x_i}{\operatorname{argmax}}\, p(x_i/\mathbf{y}) \quad \forall i \in \{1,\ldots,N\} \quad \text{and} \quad j \in \{1,\ldots,M_i\} \tag{2.3.2}$$

where $\mathbf{y} = \{\mathbf{y}_{ij}\}_{i=1,\ldots,N}^{j=1,\ldots,M_i}$ is the whole set of observations. The result in (2.3.2) represents a generalization of the MAP estimation in (2.2.39) and (2.2.51);in this setting, the MAP estimation exploits both the correlation structure encoded in the Gaussian Markov Random field and the multiple independent observations collected by the sensors that belong to the same cluster. Intuitively speaking, the result in (2.3.2) could be achieved considering a single node for each cluster that has available all the observations of the same variable; therefore, we want an algorithm that simultaneously guarantees in each cluster the consensus on the common MAP estimation in (2.3.2) and at the same time the exchange of information among different clusters. We propose a solution based on Belief Propagation technique in which the structure of the messages exchanged among sensors changes dependently on the fact wether the nodes belong or not to the same cluster. In order to implement all this we introduce a particular set of nodes $\mathcal{B}$, called *bridge nodes* such that

1)for each cluster i there exists a single bridge node ij with $j \in \{1,\ldots,M_i\}$

2)the sub-network composed of only bridge nodes must be connected

Figure 2.3 reports an example of wireless sensor network organized in clusters, where for each cluster the red sensor represents the bridge node; the arrows among nodes have different colors dependently on the type of nodes that are linked. In the following, under the hypothesis that some distributed mechanism has organized the wireless sensor network in clusters and elected for each of them a bridge node, we

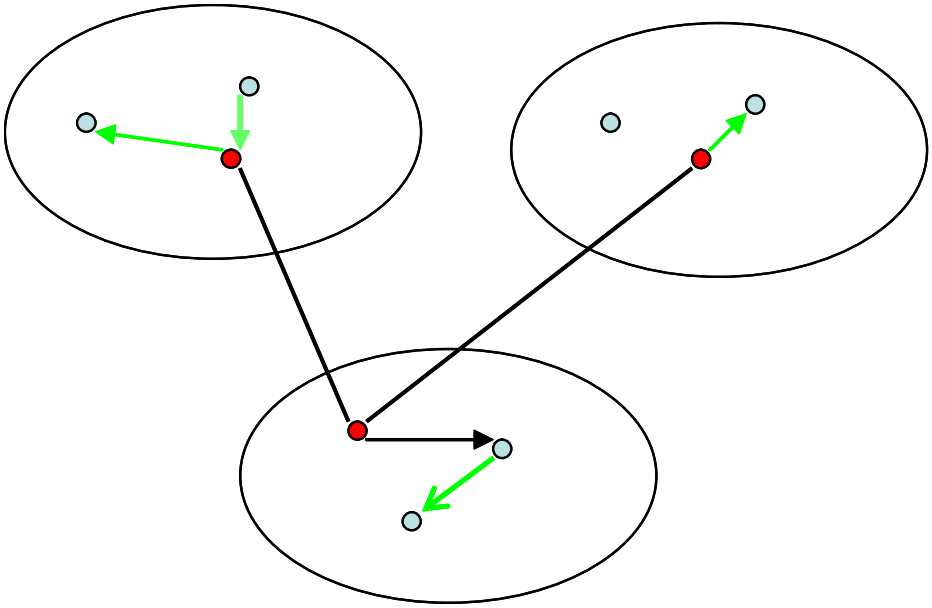

Figure 2.3: An example of wireless sensor network organized in clusters.

formulate our algorithm. Assuming that the samples of noise $\mathbf{n}_{ij}$ collected by sensors are statistically independent and a spanning tree is formed among the distributed bridge nodes, the posterior distribution of $\mathbf{x}$ given the observations is given, up to a scaling factor, by

$$p(\mathbf{x}/\mathbf{y}) \propto \frac{\prod_{(i,j)\in\mathcal{E}} p_{ij}(x_i, x_j)}{\prod_{i\in\mathcal{V}} p_i(x_i)^{N^*_{e(ki)}-1}} \prod_{i\in\mathcal{V}} \prod_{k=1,\ldots,M_i} p(x_i/\mathbf{y}_{ik}). \tag{2.3.3}$$

where $p(x_i)$,$p_{ij}(x_i, x_j)$ and $p(x_i/\mathbf{y}_{ik})$ are given respectively in (2.2.41),(2.2.42),(2.2.23) and $N^*_{e(ki)}$ the set of neighbors of the bridge node i that do not belong to the same its cluster. Remembering the factorization (2.2.19) we have as local functions

$$\phi_i(x_i) = \prod_{k=1,\ldots,M_i} p(x_i/\mathbf{y}_{ik})/p_i(x_i)^{N^*_{e(ki)}-1} \tag{2.3.4}$$

and the following compatibility functions

$$\Psi_{ij}(x_i, x_j) = p_{ij}(x_i, x_j), \tag{2.3.5}$$

where both are Gaussian distributed with $\phi_i(x_i) \sim \mathcal{N}^{-1}(\boldsymbol{\mu}_i^*, \mathbf{V}_i^*)$. The mean and covariance of each local function $\phi_i(x_i)$ are given by the following expressions

$$\boldsymbol{\mu}_i^* = (1 - N_{e(ki)}^*)\mu_s/\sigma_s^2 + \sum_{j=1}^{M_i} \mathbf{H}_{ij}^T \mathbf{R}_{ij}^{-1} \mathbf{y}_{ij} + \boldsymbol{\mu}_s/\sigma_s^2 \tag{2.3.6}$$

$$\mathbf{V}_i^* = (1 - N_{e(ki)}^*)/\sigma_s^2 + \sum_{j=1}^{M_i} \mathbf{H}_{ij}^T \mathbf{R}_{ij}^{-1} \mathbf{H}_{ij}^T + 1/\sigma_s^2, \tag{2.3.7}$$

that represent a generalization of (2.2.44) and (2.2.45) to the case of multiple independent observations of the same variable (these observations are collected by all the sensors belonging to the cluster of the i-th bridge node). The summations $\sum_{j=1}^{M_i} \mathbf{H}_{ij}^T \mathbf{R}_{ij}^{-1} \mathbf{y}_{ij}$ and $\sum_{j=1}^{M_i} \mathbf{H}_{ij}^T \mathbf{R}_{ij}^{-1} \mathbf{H}_{ij}^T$ in the expressions (2.3.6) and (2.3.7) will be available to the bridge node i and to all sensors in the i-th cluster thank to the exchange of messages that is implemented inside the cluster. In order to achieve this goal, the type of messages to be exchanged inside each cluster must be the same as the one that lead to the belief updating rule in (2.2.31) and (2.2.32), i.e. the message rule that guarantees the convergence to a common estimate in the case of a single observed variable.In fact these two terms which we desire to make available to all sensors in each cluster represent the estimate of a single variable of the Gaussian Markov Random field which we would have if the correlation structure of the field was not considered. Therefore, considering (2.3.3) inside the i-th cluster messages sent form node k to node j can have one of the two possible forms depending on the fact that node k is a bridge node or not

$$\mathbf{m}_{kj}^n(x_i) = \xi_k(x_i) \prod_{s \in N_{e(k)}/j} \mathbf{m}_{sk}^{n-1}(x_i) \quad k \text{ is not in } \mathcal{B} \tag{2.3.8}$$

$$\mathbf{m}_{kj}^n(x_i) = \xi_k^*(x_i) \prod_{s \in N_{e(k)}^*/j} \bar{\mathbf{m}}_{sk}^{n-1}(x_i) \quad k \in \mathcal{B} \tag{2.3.9}$$

achieved imposing in the message structure $\xi_k(x_i) = p(x_i/\mathbf{y}_{ik})$,$\xi_k^*(x_i) = p(x_i/\mathbf{y}_{ik})/p_i(x_i)^{N^*_{e(ki)}-1}$,$\zeta_{kj} = 1\{x_k = x_j\}$, with $N_{e(k)}$ given by the set of neighbors of each node inside the same cluster, $N^*_{e(k)}$ the set of neighbors of a bridge node that belong both its same cluster and the bridge set. The messages sent from a bridge node inside each cluster given in (2.3.9) provides the mechanism through which the statistical information relative to the correlation structure given by the Gaussian Markov Random field flows in each cluster; the messages $\bar{\mathbf{m}}_{sk}^{n-1}$ in (2.3.9) can have two different forms dependently on the fact that they are sent from other bridge nodes or not. The messages in (2.3.8) and (2.3.9) are clearly Gaussian distributed so that they can be parameterized by their mean and covariance; the message updating rule inside the i-th cluster is given by the following expressions

$$\boldsymbol{\mu}_{kj}^n = \mathbf{H}_{ik}^T \mathbf{R}_{ik}^{-1} \mathbf{y}_{ik} + \sum_{s \in N_{e(k)}/j} \boldsymbol{\mu}_{sk}^{n-1}, \quad k \text{ is not in } \mathcal{B} \tag{2.3.10}$$

$$\mathbf{V}_{kj}^n = \mathbf{H}_{ik}^T \mathbf{R}_{ik}^{-1} \mathbf{H}_{ik} + \sum_{s \in N_{e(k)}/j} \mathbf{V}_{sk}^{n-1} \quad k \text{ is not in } \mathcal{B} \tag{2.3.11}$$

and

$$\boldsymbol{\mu}_{kj}^n = (1 - N^*_{e(ki)})\mu_s/\sigma_s^2 + \mathbf{H}_{ik}^T \mathbf{R}_{ik}^{-1} \mathbf{y}_{ik} + \sum_{s \in N^*_{e(k)}/j} \bar{\boldsymbol{\mu}}_{sk}^{n-1}, \quad k \in \mathcal{B} \tag{2.3.12}$$

$$\mathbf{V}_{kj}^n = (1 - N^*_{e(ki)})/\sigma_s^2 + \mathbf{H}_{ik}^T \mathbf{R}_{ik}^{-1} \mathbf{H}_{ik} + \sum_{s \in N^*_{e(k)}/j} \bar{\mathbf{V}}_{sk}^{n-1} \quad k \in \mathcal{B} \tag{2.3.13}$$

where $\bar{\boldsymbol{\mu}}_{sk}^{n-1}$ and $\bar{\mathbf{V}}_{sk}^{n-1}$ are given by (2.3.10) and (2.3.11) if node s belongs to the same cluster of the bridge node k, otherwise they assume the forms in (2.3.14) and (2.3.15)

$$\boldsymbol{\mu}_{sk}^n = \frac{\rho_{sk}(\boldsymbol{\mu}_k^{n-1} + \sum_{y \in N^*_{e(s)/k}} \bar{\boldsymbol{\mu}}_{ys}^{n-1} - \rho_{sk}\boldsymbol{\mu}_s/\sigma_s^2(1-\rho_{sk}^2))}{(1 + \sigma_s^2(1-\rho_{sk}^2)(\boldsymbol{\mu}_k^{n-1} + \sum_{y \in N^*_{e(s)/k}} \bar{\mathbf{V}}_{ys}^{n-1}))} + \rho_{sk}\boldsymbol{\mu}_s/\sigma_s^2(1-\rho_{sk}^2) \tag{2.3.14}$$

$$\mathbf{V}_{sk}^n = \frac{(\mathbf{V}_k^{n-1} + \sum_{y \in N^*_{e(s)/k}} \bar{\mathbf{V}}_{ys}^{n-1} + 1/\sigma_s^2)}{(1 + \sigma_s^2(1-\rho_{sk}^2)(\mathbf{V}_k^{n-1} + \sum_{y \in N^*_{e(s)/k}} \bar{\mathbf{V}}_{ys}^{n-1}))} + \rho_{sk}\boldsymbol{\mu}_s/\sigma_s^2(1-\rho_{sk}^2) \tag{2.3.15}$$

similar to (2.2.47) and (2.2.48). From expressions (2.3.14) and (2.3.15) it follows that each bridge node updates the variables $\mathbf{V}_i^{n-1}$ and $\boldsymbol{\mu}_i^{n-1}$ during the execution of the algorithm.The updating rule uses the messages exchanged with the nodes inside its same cluster and is given by

$$\boldsymbol{\mu}_k^n = (1 - N^*_{e(ki)})\mu_s/\sigma_s^2 + \mathbf{H}_{ik}^T\mathbf{R}_{ik}^{-1}\mathbf{y}_{ik} + \sum_{s\in N_{e(k)}} \bar{\boldsymbol{\mu}}_{sk}^{n-1}, \quad k \in \mathcal{B} \tag{2.3.16}$$

$$\mathbf{V}_k^n = (1 - N^*_{e(ki)})/\sigma_s^2 + \mathbf{H}_{ik}^T\mathbf{R}_{ik}^{-1}\mathbf{H}_{ik} + \sum_{s\in N_{e(k)}} \bar{\mathbf{V}}_{sk}^{n-1} \quad k \in \mathcal{B}. \tag{2.3.17}$$

Considering that the beliefs for any node in the wireless sensor network are Gaussian distributed, the belief updating rules at a generic sensor that belongs to the bridge set $\mathcal{B}$ are given by the next expressions

$$\mathbf{q}_{ik}^n = (1 - N^*_{e(ki)})\mu_s/\sigma_s^2 + \mathbf{H}_{ik}^T\mathbf{R}_{ik}^{-1}\mathbf{y}_{ik} + \sum_{s\in N^*_{e(k)}} \boldsymbol{\mu}_{sk}^{n-1}, \quad k \in \mathcal{B} \tag{2.3.18}$$

$$\mathbf{W}_{ik}^n = (1 - N^*_{e(ki)})/\sigma_s^2 + \mathbf{H}_{ik}^T\mathbf{R}_{ik}^{-1}\mathbf{H}_{ik} + \sum_{s\in N^*_{e(k)}} \mathbf{V}_{sk}^{n-1} \quad k \in \mathcal{B}, \tag{2.3.19}$$

while for a node that is not a bridge we have

$$\mathbf{q}_{ik}^n = \mathbf{H}_{ik}^T\mathbf{R}_{ik}^{-1}\mathbf{y}_{ik} + \sum_{s\in N_{e(k)}} \boldsymbol{\mu}_{sk}^{n-1}, \quad k \text{ is not in } \mathcal{B}, \tag{2.3.20}$$

$$\mathbf{W}_{ik}^n = \mathbf{H}_{ik}^T\mathbf{R}_{ik}^{-1}\mathbf{H}_{ik} + \sum_{s\in N_{e(k)}} \mathbf{V}_{sk}^{n-1} \quad k \text{ is not in } \mathcal{B}, \tag{2.3.21}$$

As a numerical example, we consider the application of the proposed algorithm to a particular Gaussian Markov Random field composed of four variable to be estimated;the graphical model and the communication graphs inside each cluster are loops free, so that the convergence is guaranteed. In Figure 2.4 we report the MSE in the estimation of the Gaussian Markov Random field variables as a function of the SNR (defined considering the observation noise) for different values of the number of

sensors inside each cluster (nodes that observes the same variable). Clearly, the MSE decreases as the dimension of each cluster increases, at the expense in general of a greater number of iterations necessary to converge. Figure 2.5 reports for each node in the wireless sensor network the MAP estimation as a function of the algorithm iterations;for this particular network and graphical model the convergence is reached in a few iterations.

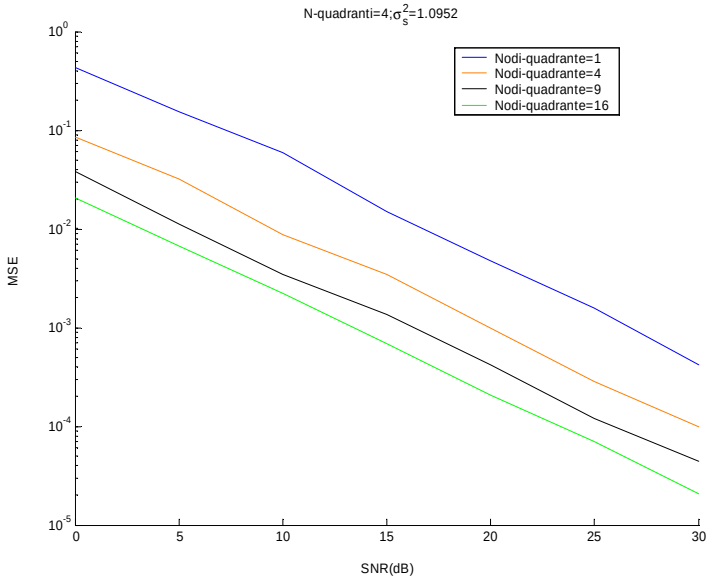

Figure 2.4: MSE as a function of SNR for different dimesions of the clusters.

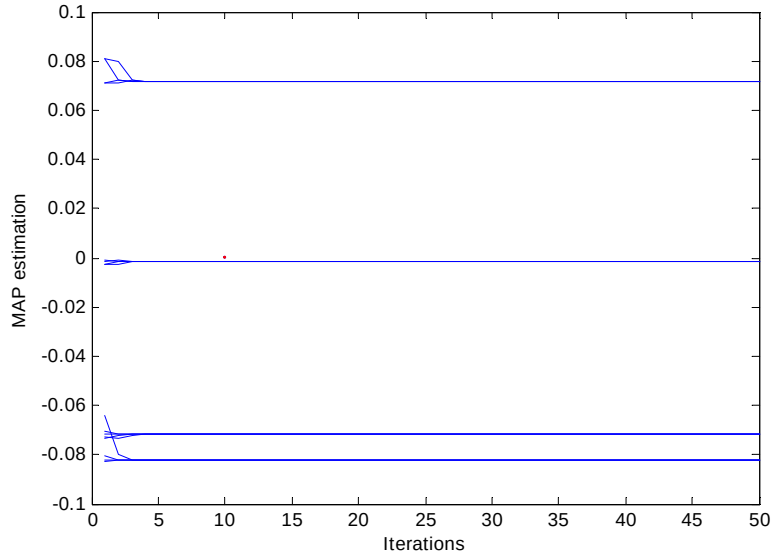

Figure 2.5: MAP estimation of all nodes as a function of the algorithm iterations.

The proposed approach is based on Belief propagation technique and it is known that this kind of message passing algorithm is guaranteed to converge if and only if the underlying graphical model is a graph without cycles, or in other words a tree. In our setting, we must distinguish two different topological levels in the structure of the wireless sensor network. The graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ that describes the statistical structure associated to the underlying Gaussian Markov Random field defines the statistical links that must exist among bridge nodes on which the exchange of message among clusters is implemented if a corresponding physical link can be established. Inside each cluster, considering the messages necessary to reach consensus, the exchange of messages must happen according to a physical structure that guarantees connectivity without presence of cycles. If this condition is satisfied and graph $\mathcal{G}$ is without loops, our algorithm is guaranteed to converge; in particular, it converges in a finite number of steps that depends on the topology of the network and graphical model but that

in the worst case is given by the following number of iterations:

$$N_{max} = N + 2 \max_{i \in \{1,\ldots,N\}} M_i \tag{2.3.22}$$

The previous considerations suggest that the bridge nodes must be chosen in order to guarantee that the exchange of messages relative to the graphical model $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is totally implementable. The messages exchanged inside each cluster (2.3.8) and (2.3.9) can be rewritten in the form (2.2.26); upon collecting messages (in log domain) corresponding to each source value from all edges into a column vector $\mathbf{z}_x^n$ of size $2\|E\| \times 1$, and similarly defining a vector $\mathbf{u}_x$ for the first term in the right hand side (RHS) of (2.2.26), we achieve

$$\mathbf{z}_x^n = \mathbf{u}_x + \mathbf{A}\mathbf{z}_x^{n-1} \tag{2.3.23}$$

where the square matrix $\mathbf{A}$ captures the characteristics of the graph relative to nodes inside the cluster considered as represented by the second term in the right hand side of (2.2.26). Viewing this equation as a mapping $f(\mathbf{x}) = \mathbf{u}_x + \mathbf{A}\mathbf{x}$, with $f^{'}(\mathbf{x}) = \mathbf{A}$ and the contraction mapping principle, it can be shown that if the spectral radius of $\mathbf{A}$, $\rho(\mathbf{A}) < 1$,$\mathbf{z}_x^n \rightarrow \mathbf{z}_x^\infty = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{u}_x$ for any initial messages. Also, upon convergence the final belief at node v is given by (2.2.21)

$$\mathbf{b}_v(\mathbf{x}) = \alpha\phi_v(\mathbf{x}) \exp\left(\mathbf{1}_v^T \mathbf{z}_x^\infty\right) = \alpha\phi_v(\mathbf{x}) \exp\left(\mathbf{1}_v^T \sum_{k=0}^{\infty} \mathbf{A}^k \mathbf{u}_x\right) \tag{2.3.24}$$

where $\mathbf{1}_v$ denotes a vector of the same dimension as $\mathbf{z}_x^\infty$, with ones at positions corresponding to the incoming edges of node v and zeroes otherwise. Note that $\mathbf{1}_v^T \mathbf{A}^k \mathbf{u}_x = \sum_{v' \in N_{e(v)}^{k+1}} w_{v'} \log \phi_{v'}(\mathbf{x})$ admits a simple interpretation: it effectively collects local information from nodes $v^{'} \in N_{e(v)}^{k+1}$ that are distance $k+1$ away inside the same cluster, weighted by the number of paths $w_{v'}$ between them. If $w_{v'} = 1$ and

the communication graph relative to a cluster is connected, our goal is achieved, i.e., $\mathbf{b}_v(\mathbf{x}) \propto \prod_{v' \in \mathcal{V}} \phi_{v'}(\mathbf{x}) \propto p(\mathbf{x}|\mathbf{y})$. This is obviously true when the graph is a tree; in this scenario it is easy to verify that the corresponding matrix $\mathbf{A}$ is nilpotent so $\rho(\mathbf{A}) = 0$. For general graphs, some local information may be over counted (i.e., $w_{v'} > 1$) so the final beliefs may not be correct. Nonetheless, we do not need correct beliefs to make correct MAP estimates. Intuitively, we can still be on the correct side (though may be over confident) as long as all evidence are equally over counted, which dictates a certain symmetry on the communication graph relative to a cluster. Though message synchronization is assumed for simplicity, the algorithm is guaranteed to converge when $\rho(\mathbf{A}) < 1$ even with total asynchronism (i.e., arbitrary delays in message arrivals).

Bibliography

[1] J. N. Tsitsiklis, D. P. Bertsekas, M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. on Automatic Control*, pp. 803–812, Sep.1986.

[2] R. Olfati-Saber, J. A. Fax, R. M. Murray, "Consensus and Cooperation in Networked Multi-agent Systems," in *Proc. of the IEEE*, vol. 95, no. 1, pp. 215-233, Jan. 2007.

[3] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, A. Ribeiro, "Consensus in Ad Hoc WSNs With Noisy LinksPart II: Distributed Estimation and Smoothing of Random Signals", *IEEE Trans. on Signal Proc.*, April 2008, pp. 1650–1666.

[4] L. Xiao and S. Boyd, "Fast Linear Iterations for Distributed Averaging," in *Systems and Control Letters*, 53:65-78, 2004.

[5] S. Barbarossa and G. Scutari, "Bio-inspired Sensor Network Design: Distributed Decision Through Self-synchronization," *IEEE Signal Processing Magazine*, vol. 24, no. 3, May 2007, pp. 26–35.

[6] R. Olfati-Saber, "Distributed Kalman Filtering with Embedded Consensus Filters," in *Proc. of CDC 2005*, Seville, Spain, 2005, pp. 8179–8184.

[7] L. Xiao, S. Boyd, S.J. Kim, "Distributed average consensus with Least Mean Square Deviations," *Journal of Parallel and Distributed Computing*, vol. 67, Jan. 2007, pp. 33–46.

[8] S. Barbarossa, T. Battisti, A. Swami, "Globally optimal decentralized spatial smoothing for wireless sensor networks with local interactions", *Proc. of ICASSP 2008*, Las Vegas, April 2008.

[9] S. Sardellitti, M. Giona, S. Barbarossa, "Fast Distributed Consensus Algorithms Based on Advection-Diffusion Processes ", *Proc. of IEEE SAM 2008*, Darmstadt, July 2008.

[10] M. Niedzwiecki, "Identification of Time-Varying Processes", *New York: Wiley, 2000.*

[11] S. Barbarossa, G. Scutari, T. Battisti, "Distributed projection of observed data onto signal subspace in wireless sensor networks using local exchange algorithms", submitted to *IEEE Trans. on Signal Processing*, (May 2008).

[12] R. A. Horn and C. R. Johnson,*Matrix Analysis*Cambridge Univ. Press, 1985.

[13] C. Mayer, *Matrix Analysis and Applied Linear Algebra.* SIAM 2001.

[14] C. D. Meyer, R. J. Plemmons, "Convergence Powers of Matrix with Applications to Iterative Methods for Singular Linear Systems," *SIAM Jour. on Numerical Analysis* , Vol. 14, No. 4, pp. 699-705, 1977.

[15] S. M. Kay, "Fundamentals of Statistical Signal Processing - Estimation Theory", Prentice Hall (Int. Ed.), London, 1993.

[16] C, Godsil, G.Royle, "Algebraic Graph Theory", Springer-Verlag, New York, 2001.

[17] V. Blondel, J.N. Tsitsiklis, "NP-hardness of some linear control design problems," *SIAM Jour. on Control and Optimization*Vol. 35, pp. 2118–2127, 1997.

[18] A. Nemirovskii, "Several NP-hard problems arising in robust stability analysis," *Mathematics of Control, Signal and Systems* , vol. 6, pp. 99–105, 1993.

[19] R. Varga, *Matrix Iterative Analysis*, Springer-Verlag, 2th Ed., 2000.

[20] L. Pesocosolido, S. Barbarossa, and G. Scutari, "Average Consensus Algorithms Robust Against Channel Noise," in *Proc. 9th IEEE Int. Workshop on Signal Proc. Adv. in Wireless Comm. (SPAWC '08),* July 6-9, Recife, Brasil.

[21] C. Mayer, *Matrix Analysis and Applied Linear Algebra.* SIAM 2001.

[22] R. Varga, *Matrix Iterative Analysis* . Springer-Verlag, 2th Ed., 2000.

[23] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables* . SIAM Ed., 2000.

[24] C. D. Meyer, R. J. Plemmons, "Convergence Powers of Matrix with Applications to Iterative Methods for Singular Linear Systems," *SIAM Jour. on Numerical Analysis* , Vol. 14, No. 4, pp. 699-705, 1977.

[25] C. Mayer, *Matrix Analysis and Applied Linear Algebra.* SIAM 2001.

[26] D. P Bertsekas and J.N. Tsitsiklis,*Parallel and Distributed Computation: Numerical Methods*Athena Scientific, 2nd Ed., 1989.

[27] S. Boyd and L. Vandenberghe, *Convex optimization* . Cambridge University Press, 2004.

[28] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, S. Madden, "Distributed Regression: an efficient framework for modeling sensor network data", in *IPSN 2004: Proceedings of the Third International Symposium on Information Processing in Sensor networks*, pp. 1-10, ACM Press, New York, 2004.

[29] P.R. Kumar and P. Varaiya,"Stochastic systems: Estimation, identification and adaptive control", Prentice Hall, NJ, 1986.

[30] J. B. Predd, S. R. Kulkarni, H. V. Poor, "Regression in sensor networks: training distributively with alternating projections",2005.

[31] P. Honenine, M. Hessoloh, C. Richard and H. Snoussi"Distributed regression in sensor networks with a reduced-order kernel model",2008.

[32] J. Pearl, *Probabilistic Reasoning in Intelligent Systems.* San Mateo, CA:Morgan Kaufman 1988.

[33] F. R. Kschischang, B. J. Frey, H. A. Loeliger, "Factor graphs and the sum-product algorithm", in *IEEE Trans. on Information Theory 2004*, vol. 47, pp. 498-519, February 2001.

[34] S. L. Lauritzen, D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems (with discussion) ", in *J. Roy. Statistic. Soc. B*, vol. 50, pp. 155-224, January 1988.

[35] Y. Weiss, W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology ", in *Neural computation*, vol. 13, pp. 2173-2200, 2001.

[36] J.S. Yedidia, W. T. Freeman and Y. Weiss "Generalized belief propagation",*Advances in Neural Information Processing Systems*, vol. 13, pp. 689-695, 2000.

[37] P. Rusmevichientog, B. Van Roy, "An analysis of belief propagation on the turbo decoding graph with Gaussian densities ", in *IEEE Trans. Information Theory*, vol. 47, pp. 745-765, 2001.

[38] H. Dai, Y. Zhang, "Consensus estimation via belief propagation", *Conference on Information Sciences and Systems*, 2007.

[39] K.H. Plarre, P. R. Kumar, "Extended message passing algorithm for inference in loopy Gaussian graphical models", *Ad hoc networks*, vol. 2, pp. 153-169, 2004.

[40] V. Delouille, R. Neelamani, R. Baraniuk, "Robust distributed estimation in sensor networks using the embedded polygons algorithm", *IPSN' 04* , 26-27 April 2004, Berkeley, USA.

[41] E. B. Sudderth, M. J. Wainwright, A. S. Willsky, "Embedded trees: estimation of Gaussian processes on graphs with cycles", *IEEE Transaction on Signal Processing* , vol. 52, no. 11, November 2004.

[42] M. J. Wainwright, E. Sudderth and A. S. Willsky, "Tree based modeling and estimation of Gaussian processes on graphs with cycles", in *Advances in Neural Information Processing* , vol. 13, MIT press,2001.

[43] M. J. Wainwright, T.S. Jaakkola and A. S. Willsky, "Tree based reparameterization framework for analysis of sum-product and related algorithms",*IEEE Transaction on Information Theory*, vol. 49,no. 5,pp. 1120-1146, 2003.

[44] V. Delouille, R. M. Neelamani, R. G. Baraniuk, "Robust distributed estimation using the embedded subgraphs algorithm", *IEEE Transaction on Signal Processing* , vol. 54, no. 8, August 2006.

[45] V. Delouille, R. M. Neelamani, R. G. Baraniuk, V. Chandrasekaran, R.G. Baraniuk, "The embedded triangles algorithm for distributed estimation in sensor networks", 2003.

[46] A. Dogandzic, B. Zhang, "Distributed estimation and detection for sensor networks using hidden markov random field models", *IEEE Transaction on Signal Processing* , vol. 54, no. 8, August 2006.

[47] I. D. Schizas, A. Ribeiro, G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links-Part I: Distributed estimation of deterministic signals", *IEEE Transaction on Signal Processing* , vol. 56, no. 1, January 2008.

[48] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, A. Ribeiro, "Consensus in ad hoc WSNs with noisy links-Part II: Distributed estimation and Smoothing of random signals", *IEEE Transaction on Signal Processing* , vol. 56, no. 4, April 2008.

[49] A. Nedic, A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization", October 2007.

[50] T. S. Han, and K. Kobayashi, "A new achievable rate region for the interference channel", *IEEE Trans. on Inform. Theory*, vol. 27, pp. 49-60, Jan. 1981.

[51] E. C. van der Meulen, "Some Reflections on the Interference Channel", *Boston, MA, Kluwer*, pp. 409-421, 1994.

[52] R. Cendrillon, W. Yu, M. Moonen, J. Verlinder, and T. Bostoen,"Optimal Multi-User Spectrum Managment for Digital Subscriber Lines`, in *IEEE Trans. on Commun.*, Vol. 54, No. 5, pp. 922 - 933, May 2006.

[53] J. Huang, R. Berry and M. Honig,"Distributed Interference Compensation for Wireless Networks", to appear in *IEEE on Journal on Selected Areas in Commun., special issue on "Price-based Access Control and Economics for Communication Networks".*

[54] W. Yu, and Raymond Lui, "Dual Methods for Nonconvex Spectrum Optimization of Multicarrier Systems", accepted in *IEEE Trans. on Commun.*, 2006. Available at `http://www.comm.toronto.edu/ weiyu/publications.html`.

[55] G. Scutari, D. P. Palomar, and S. Barbarossa,"Simultaneous Iterative Water-Filling for Gaussian Frequency-Selective Interference Channels", in *Proc. 2006 IEEE Int. Symp. on Inform. Theory*, July 2006

[56] D. P Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, 2nd Ed., 1989.

[57] G. Scutari, *Competition and Cooperation in Wireless Communication Networks*, PhD. Dissertation, University of Rome, "La Sapienza", November 2004.

[58] N. Yamashitay and Z. Q. Luo, "A Nonlinear Complementarity Approach to Multiuser Power Control for Digital Subscriber Lines", accepted for publication on *Optimization Methods and Software.*

[59] Z.-Q. Luo and J.-S. Pang, "Analysis of Iterative Waterfilling Algorithm for Multiuser Power Control in Digital Subscriber Lines," in the special issue of *EURASIP Journal on Applied Signal Processing on Advanced Signal Processing Techniques for Digital Subscriber Lines,* Article ID 24012, pp. 1–10, April 2006.

[60] R. Etkin, A. Parekh, D. Tse, "Spectrum sharing for unlicensed bands", in *Proc. of the Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sept. 28-30, 2005.

[61] J. Besag, "Spatial interaction and the statistical analysis of lattice systems", in *Journal of Royal Statistics Society*, Series B 36 (1974), pp. 192-223.

[62] K. Chou, A.S. Willsky, R. Nikoukhah, "Multiscale systems, Kalman filters, and Riccati equations",*IEEE Transactions on Automatic Control*, 39 (1994), pp. 479-492.

[63] B. Frey, D.J.C. MacKay,"A revolution:Belief Propagation in graphs with cycles ",in *Advances in neural information processing systems*, vol. 10, MIT press, 1998.

[64] S. T. Chung, S. J. Kim, J. Lee, and J. M. Cioffi, “A game-theoretic approach to power allocation in frequency-selective Gaussian interference channels", in *Proc. 2003 IEEE Int. Symp. on Inform. Theory (ISIT 2003)*, p. 316, June 2003.

[65] G. Scutari, S.Barbarossa, and D.Ludovici, "Cooperation Diversity in Multihop Wireless Networks Using Opportunistic Driven Multiple Access", in *Proc. of the 2003 IEEE Workshop on Sig. Proc. Advances in Wireless Comm., (SPAWC-2003)*, pp. 170-174, June 2003.

[66] G. Scutari, D. P. Palomar, and S. Barbarossa, "A Game Theoretic Approach for the MIMO Interference Channel",submitted to *IEEE JSAC*, August 2007.

[67] D. P. Bertsekas and J.N. Tsitsiklis, "Convergence Rate and Termination of Asynchronous Iterative Algorithms", in *Proc. of the 1989 International Conference on Supercomputing*, Irakleion, Greece, June 1989, pp. 461-470.

[68] J.N. Tsitsiklis, "A Comparison of Jacobi and Gauss-Seidel Parallel Iterations", *Applied Mathematics Letters*, Vol. 2, No. 2, 1989, pp. 167-170.

[69] G. Scutari, D. P. Palomar, and S. Barbarossa, "Asynchronous Iterative Water-filling for Gaussian Frequency-Selective Interference Channels: A Unified Framework",submitted on *IEEE Trans. on Information Theory*,August 2006.

[70] R. Gallager, "A perspective on multiaccess channels",*IEEE Trans. on Information Theory*, vol. 31, no. 2, pp. 124-142, 1985.

[71] A. Ephremides and B. Hajek, "Information theory and communication networks: unconsummated union",*IEEE Trans. on Information Theory*, vol. 44, no. 6, pp. 2416-2434, 1998.

[72] G. Scutari, D. P. Palomar, and S. Barbarossa, "Optimal Linear Precoding Strategies for Wideband Non-Cooperative Systems based on Game Theory-Part I: Nash Equilibria",to appear on *IEEE Trans. on Signal Processing* ,2007.

[73] L. Pesocosolido, S. Barbarossa, and G. Scutari, "Average Consensus Algorithms Robust Against Channel Noise", in *Proc. 9th IEEE Int. Workshop on Signal Proc. Adv. in Wireless Comm. (SPAWC 08)*, July 6-9, Recife, Brasil.

[74] G. Scutari, D. P. Palomar, and S. Barbarossa, "Optimal Linear Precoding Strategies for Wideband Non-Cooperative Systems based on Game Theory-Part II: Algorithms",to appear on *IEEE Trans. on Signal Processing*, 2007.

[75] J. G. Proakis *Digital Communications*,Third ed. NewYork: McGraw- Hill, 1995.

[76] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity",*IEEE Trans. on Information Theory*, vol. 39, no. 2, pp. 466-478, 1993.

[77] M. Jordan, C. Bishop, "Introduction to Graphical models ",preprint 2001.

[78] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar and P. Whiting, "Providing quality of service over a shared wireless link",*IEEE Communications Magazine*, vol. 39, no. 2, pp. 150-154, 2001.

[79] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2003.

[80] G. Wunder and C. Zhou, "Queuing Analysis for the OFDMA Downlink: Throughput regions, delay and exponential backlog bounds".

[81] W. Yu, G. Ginis, and J. M. Cioffi, "Distributed multiuser power control for digital subscriber lines ", *IEEE J. Select. Areas Commun.*, vol. 20, pp. 1105-1115, June 2002.

[82] S. T. Chung, S. J. Kim, J. Lee, and J. M. Cioffi, "A game-theoretic approach to power allocation in frequency-selective Gaussian interference channels ", in *Proc. 2003 IEEE Int. Symp. on Inform. Theory (ISIT 2003)*, p. 316, June 2003.

[83] T. Michel and G. Wunder, "Minimum rates scheduling for OFDM broadcast channels",*Proc. ICASSP 2006*, 2006.

[84] Guocong Song and Y. Li, "Cross-layer optimization for OFDM wireless networks:Part I and II",*IEEE Transactions on Wireless Communications* , vol. 4, no. 2,pp. 614-634, March 2005.

[85] N. Yamashitay and Z. Q. Luo, "A Nonlinear Complementarity Approach to Multiuser Power Control for Digital Subscriber Lines"", accepted for publication on *Optimization Methods and Software.*

[86] Z.-Q. Luo and J.-S. Pang, "Analysis of Iterative Waterfilling Algorithm for Multiuser Power Control in Digital Subscriber Lines"," in the special issue of *EURASIP Journal on Applied Signal Processing on Advanced Signal Processing Techniques for Digital Subscriber Lines,* Article ID 24012, pp. 1–10, April 2006.

[87] G. Wunder, C. Zhou and A. Feistel, "Optimal OFDM downlink scheduling for UMTS HSDPA evolution",*Asilomar Conference on Signals, Systems and Computers 2004* , November 2004.

[88] G. Song, Y. Li, L. Cimini and H. Zheng, "Joint channel-aware and queue aware data schedulingin multiple shared wireless channels",*IEEE Wireless Communications and Networking Conference 2004* , WCNC 2004.

[89] R. Leelahakriengkrai and R. Agrawal, "Scheduling in multimedia wireless networks",*IEEE Transactions on Vehicular Technology* , vol. 52, pp. 226-239, 2003.

[90] S. Shakkottai and A. L. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule",*Translations of the AMS, A volume in memory of F. Karpelevich*,2002.

[91] A. Erylmaz, R. Srikant and J. Perkins, "Stable scheduling policies for broadcast channels",in *Proc. ISIT*, p. 382, 2002.

[92] K. Kumaran and L. Qian, "Uplink scheduling in cdma packet-data systems",in *Proceedings of Infocom*, 2003.

[93] Neely, M.J.,Modiano,E., and Rohrs, C.E. (2002). "Power and server allocation in a multi beam satellite with time varying channels", *In Proc. Infocom 2002*,New York City, pp. 138-152.

[94] Yeah, E., and Cohen, A. (2004). Information theory, queuing, and resource allocation in multi-user fading communications.*In Proc. Conf. Information Sciences and Systems*,Princeton,NJ, pp. 1396-1401.

[95] Berry, R.A., and Yeah,E.M. (2004). Cross-layer wireless resource allocation *IEEE Signal Processing Magazine*,pp. 59-68.

[96] Lau, V.K.N., and Kwok, Y.K.R.(2006). Channel adaptive technologies and cross-layer designs for wireless systems with multiple antennas: Theory and Applications, John Wiley and Sons Inc.

[97] J. Nash, “Equilibrium Points in n person Game”, in *Proc. National Academy of Science*, vol. 36, pp. 48-49, 1950.

[98] B. Prabhakar, E. Uysal-Biyikoglu and A. E. Gamal,"Energy efficient transmission over a wireless link via lazy packet scheduling",*In Proc. Infocom 2001*,Alaska,2001.

[99] I. E. Telatar and R. Gallager," Combining queuuing theory with information theory for multiaccess",*IEEE Journal on Selected Areas in Communications*,vol. 13, no. 6, pp. 963-969, 1995.

[100] M. Medard, J. Huang, A. Goldsmith and S. Meyn ," Capacity of time-slotted ALOHA packetized multiple-access system ",*in Proc. ISIT*,Sorrento,2000.

[101] E. Yeh ,"Multiaccess and Fading in Communication networks ", Phd Thesis,Massachussets Institute of Technology, 2001.

[102] E. Yeh ,"An inter-layer view of multiaccess communications ", *In Proceeding of ISIT 2002*,p. 112, 2002.

[103] E. Yeh and A. Cohen ,"Throughput and delay optimal resource allocation in multiaccess fading channels ", *In Proceeding of ISIT 2002*,Japan, p. 245, 2003.

[104] M. Goyal, V. Sharma and A. Kumar ,"optimal resource allocation policies for multiaccess fading channel with a quality of service constraint", *In Proceeding of ISIT 2002*, p. 81, 2002.

[105] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.

[106] J. P. Aubin, *Mathematical Method for Game and Economic Theory*, Elsevier, Amsterdam, 1980.

[107] J. G. David Forney and M. V. Eyuboglu, "Combined equalization and coding using precoding" ", *IEEE Commun. Mag.*, vol. 29, no. 12, pp. 25–34, Dec. 1991.

[108] A. J. Goldsmith and S.-G. Chua, "Variable-rate variable-power MQAM for fading channels "", *IEEE Trans. on Commun.*, vol. 45, no. 10, pp. 1218–1230, Oct. 1997.

[109] D. P. Palomar, and S. Barbarossa, "Designing MIMO Communication Systems: Constellation Choice and Linear Transceiver Design" ", *IEEE Trans. on Signal Processing*, Vol. 53, No. 10, pp. 3804-3818, Oct. 2005.

[110] J. Rosen, "Existence and Uniqueness of Equilibrium Points for Concave n-Person Games", *Econometrica*, vol. 33, no. 3, pp. 520–534, July 1965.

[111] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley and Sons, 1991.

[112] D. Tse and P. Viswanath *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.

[113] G. Scutari, *Competition and Cooperation in Wireless Communication Networks*, PhD. Dissertation, University of Rome, "La Sapienza", November 2004.

[114] D. P Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, 2nd Ed., 1989.

[115] Z. Quan, S. Cui, V. H. Poor, A. H. Sayed "Collaborative wideband sensing for cognitive radios",*IEEE Signal Processing Magazine*, pp. 6073, Nov. 2008.

[116] C. G. Lopes, A. H. Sayed, "Diffusion least mean squares over adaptive networks: Formulation and performance analysis",in *IEEE Trans. on Signal Processing*, vol. 56, May 2007, pp. 31223136.

[117] J. A. Bazerque, G.B. Giannakis, "Distributed Spectrum Sensing for Cognitive Radios by Exploring Sparsity", in *Proc. of 42nd Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 629, 2008

[118] D. S. Bernstein, "Matrix Mathematics", Princeton Univ. Press, 2005.

[119] C.M. Bishop, "Pattern Recognition and Machine Learning", Springer, Oct. 2007.

[120] S. Barbarossa, G. Scutari, T. Battisti, "Distributed projection algorithms in wireless sensor networks through local interaction mechanisms",in *Proc. of ICASSP 2009*, April 2009, Taipei.

[121] A. Anandkumar, L. Tong and A. Swami,"Detection of Gauss-Markov fields with nearest neighbor dependency",*IEEE Transaction on Information Theory*, January 2007.

[122] M.D. Penrose and J.E. Yukich,"Central limit theorems for some graphs in computational geometry",*Annals of applied probability*,vol. 11, no. 4,pp. 1005-1041, 2001.

[123] J.M.F. Moura and N. Balram,"Recursive structure of noncausal Gauss Markov Random fields", *IEEE Transaction IT.*,vol. IT-38, no. 2,pp. 334-354, 1992.

[124] D. Eppstein, M.S. Paterson and F.F. Yao,"On nearest-neighbor graphs ", *Discrete and computational geometry*,vol. 17,pp. 263-282, 1997.

[125] C.B. Barber, D.P. Dobkin and H.T. Hundhanpaa,"The Quickhull algorithm for convex hulls ", *ACM Transactions Math. Software*,vol. 22,pp. 469-483, 1996.

[126] J. Liebeherr, M. Nahas and W. Si,"Application layer multicast with Delaunay triangulations ",tech. report,CS Dept.,University of Virginia, 2001.