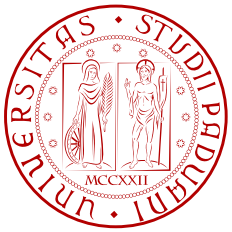



UNIVERSITÀ DI PADOVA FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

SCUOLA DI DOTTORATO IN INGEGNERIA DELL'INFORMAZIONE

INDIRIZZO IN SCIENZA E TECNOLOGIA DELL'INFORMAZIONE


XXIII Ciclo

# Distributed Optimization and Data Recovery for Wireless Networking


Dottorando

RICCARDO MASIERO

**Supervisore:**

Chiar.mo Dr. Michele Rossi

**Direttore della Scuola:**

Chiar.mo Prof. Matteo Bertocco



Anno Accademico 2010/2011

*I'm on fire to explain, and happiest when it's something reasonably intricate which I can make clear step by step. It's the easiest way I can clarify things in my own mind.*

Ardo dal desiderio di spiegare, e la mia massima soddisfazione è prendere qualcosa di ragionevolmente intricato e renderlo chiaro passo dopo passo. È il modo più facile per chiarire le cose a me stesso.

*(Isaac Asimov)*

*a Elena*

*a Mamma e Papà*

*a Valentina*

*Ai miei amici:*

*a chi c'è sempre stato*

*e a chi vorrei ci fosse ancora.*

# Ringraziamenti

A Mirna e Giuseppe, mamma e papà, che amo e rispetto, e che non riuscirò mai a ringraziare abbastanza.
A Valentina, che mi vuole bene davvero, e lo so.
A Mariuccia e Paolo, che mi hanno accolto con affetto nella loro famiglia.
Ad Elena, che mi sostiene, sopporta, coccola e ama.

Agli amici di sempre, Alice A., Alice C., Andy, Claudia, Giulio, Japo, Laura, Lele, Mary, Marta, Meba, Michael, Michele, Scussi, Simon, senza i quali non saprei stare.

Ai miei amici e compagni di avventura, Giorgio, Davide, Federico, Francesco, Marco, Riccardo, che hanno reso speciali i miei tre anni di dottorato.

A Michele R. e Michele Z., che hanno saputo insegnarmi e guidarmi nella mia attività di ricerca e crescita professionale, e che ancora continuano a farlo.
A Giovanni, che mi ha regalato nuova e sincera passione per la ricerca.

A tutto il gruppo SIGNET in particolare, e al gruppo DIGIT in generale, che ogni giorno mi fanno rendere conto di lavorare in un ambiente bellissimo.
À l'équipe MAESTRO, qui m'a fait sentir à la maison.

Grazie, Riccardo.

# Contents

# List of Figures

# List of Tables

# Abstract

My research activity focused on the field of heterogeneous wireless networks and has been particularly inspired by the problem of sensing a city-wide environment through a large scale, partially distributed, mobile and low cost network (possibly composed of mobile phones or similar user's equipment). In my PhD thesis I have been guided by the grand vision of a two tier architecture which integrates existing cellular systems with different types of distributed networks (these could be mixtures of ad hoc, sensor networks and so on). In fact, a fully distributed infrastructure alone would be inappropriate when the network is very large in size and highly populated (e.g., urban area networks). In such a case, the network organization itself would be energy draining and probably impractical. On the other hand, a cellular system alone does not have the flexibility and the instruments to get a fine grained view of all the data generated within such a network.

This envisioned scenario, besides featuring a number of mobile phones, also consists of a mixture of embedded devices, which are expected to have on-board radio and sensing capabilities. Nowadays technology makes us more and more able to control the environment we are in through motion sensors, GPS, health care devices, microphones and video-cameras. Wireless Sensor Networks (WSNs), for instance, are infrastructures made of small devices (nodes) equipped with "intelligent sensors" able to sense their surroundings for, e.g., light, temperature, humidity and/or pollution. Therefore, mobile phones as well as other network elements, including base stations, routers and access points hosting diverse wireless and wired technologies, can cooperate to accomplish a common task like the detection of a fire or the monitoring of a physical phenomenon.

Exploiting the fact that cell phones are becoming a communication hub in our daily life, we can foresee the integration of standard cellular systems with overlayed distributed networks such as WSNs. The ultimate goal of this is to "connect" everything has some communication capability, possibly providing self-configurability and self-adaptability of the

network. We note that current cellular networks already implement some of these features: user positions, to a certain extent, can be tracked already and services can be provided based on contextual information. As a matter of fact, we are depicting a Delay Tolerant Network (DTN) scenario, where heterogeneous, sparse and/or mobile wireless networks communicate with each other, but where, due to the inherent nature of the infrastructure itself, no continuous connectivity can be assumed.

The above grand vision entails quite a few challenges, and during my research activity I have been focusing on the following ones: 1) the design of reconstruction algorithms that from a subset of the data (i.e., from the collection of the sensor readings from a small fraction of nodes) are able to reconstruct with high accuracy the data monitored over the entire sensor field (these algorithms allow for scalability of the system as they decrease the number of data packets to collect for a given accuracy goal); 2) the design of cooperative networking protocols, where cooperation is utilized to reach a common goal such as the detection of a fire or/and to increase the network performance in terms of optimization of given performance metrics, e.g., energy consumption, delivery time, delivery probability.

Concerning the first point, my study explores the capabilities of Compressive Sensing (CS), a technique that has been proved to be very effective for the compression and recovery of correlated signals, with the objective of designing and implementing a system for the efficient acquisition of large data sets in distributed (sensor) networks. The goal of this system is to reconstruct large signals through the collection of the smallest number of samples that will keep the reconstruction quality above a minimum target level. The steps of my research activity can be summarized as follows: 1.a) assess the applicability and potential benefits of CS in networking applications; 1.b) provide a sound theoretical justification of the effectiveness of CS recovery when coupled with Principal Component Analysis (PCA) along with a characterization of the optimality of the reconstruction process as a function of the statistics of the input signal; 1.c) design an algorithm for signal reconstruction based on CS and validating the proposed method through Matlab simulations as well as real signal traces.

For the second point, my work has been centered around distributed optimization methods whose objective is that of optimizing network wide (global) performance metrics. In detail, in the investigated scenario nodes collaborate to minimize the sum of local objective functions, which in general depend on global variables such as the network protocol parameters or actions taken by all the nodes in the network. In the case where the local

objective functions are convex, it is possible to adopt a framework that relies on local sub-gradient methods and consensus algorithms to average the information from each node, while granting convergence towards global optimal solutions. However, existing convergence results for this framework can only be applied in the case of synchronous operations of the nodes and mobility models without memory. My research addresses and solves these issues, and its fundamental steps were: 2.a) the extension of the convergence results to the optimal solution for a more general class of mobility models; 2.b) the application of distributed sub-gradient methods under asynchronous operations; 2.c) the presentation of a possible networking scenario to validate the analysis, showing the effectiveness of the considered distributed optimization technique.

The outcomes of my research are useful tools for the optimization of practical network protocols and provide recommendations for the design of the integrated communication and sensing system that we have envisioned above.

# Sommario

Durante la mia attività di ricerca mi sono concentrato sullo studio di problematiche relative alle reti wireless eterogenee, ispirandomi in particolare ad uno scenario di monitoraggio urbano realizzato per mezzo di una rete di comunicazione estesa su vasta scala, parzialmente distribuita, mobile e a basso costo (possibilmente composta da telefoni cellulari o simili). Nella mia tesi di dottorato, dunque, sono stato guidato dalla visione globale di una architettura a due livelli, che permettesse l'integrazione dei sistemi cellulari esistenti con varie tipologie di reti distribuite (quali, ad esempio, le reti ad hoc o di sensori). Infatti, una infrastruttura completamente distribuita sarebbe inappropriata nel caso di una rete di grandi dimensioni e composta da numerosi dispositivi (si pensi, ad esempio, a reti su scala urbana). In questo caso, l'organizzazione stessa dell'infrastruttura di comunicazione risulterebbe assai dispendiosa in termini energetici e probabilmente impraticabile. D'altro canto, il solo sistema cellulare non possiede la flessibilità e gli strumenti per sfruttare la granularità di informazione prodotta dai dati generati in una tale rete.

Nello scenario considerato, oltre alla presenza di un certo numero di cellulari mobili, è implicita anche l'esistenza di dispositivi (*embedded devices*) di varia natura, capaci di comunicare tra loro e con gli altri elementi della rete, e che ci si aspetta possano anche "misurare" l'ambiente circostante. Al giorno d'oggi, infatti, la tecnologia ci rende sempre più capaci di controllare la realtà quotidiana attraverso sensori di movimento, GPS, strumentazioni per il monitoraggio medico, microfoni e video-camere. Le reti di sensori (*Wireless Sensor Networks*, WSN), ad esempio, sono infrastrutture costituite da piccoli dispositivi (nodi) dotati di "sensori intelligenti" capaci di misurare l'ambiente circostante in termini di luminosità, temperatura, umidità, inquinamento e/o altro. Perciò, è possibile pensare che i telefoni cellulari, così come altri elementi di rete (incluse le *base station*, i *router* e gli *access point* su cui si basano diverse tecnologie cablate e non), possano cooperare per realizzare un obiettivo comune come l'individuazione di un incendio o il monitoraggio di un fenomeno fisico.

Osservando il fatto che i telefoni cellulari sono sempre più al centro delle comunicazioni quotidiane, è possibile prevedere l'integrazione dei sistemi cellulari standard con reti distribuite aggiuntive come le WSN. Lo scopo ultimo sarebbe quello di "connettere" qualsiasi cosa in grado di comunicare, trasmettendo e ricevendo dell'informazione, e possibilmente dotare tale rete di meccanismi autonomi di configurazione e adattamento. Si noti che le reti cellulari odierne già implementano alcune di queste caratteristiche: sulla base di informazioni contestuali, infatti, è oggigiorno possibile determinare la posizione di un utente con una certa accuratezza e fornirgli determinati servizi. Di fatto, si sta considerando una infrastruttura di rete che può essere classificata come una *Delay Tolerant Network* (DTN). In questo tipo di infrastruttura, reti wireless eterogenee, sparse e/o mobili, comunicano tra loro, ma tale comunicazione non può essere assunta continua a causa della natura stessa della reti interagenti.

La visione generale di cui sopra porta con sé numerose problematiche, e durante la mia attività di ricerca mi sono concentrato in particolare sulle seguenti due: 1) la progettazione di algoritmi di ricostruzione che a partire da un sottoinsieme di dati (ossia, dalla raccolta parziale delle letture dei nodi che costituiscono l'intera rete) sono in grado di ricostruire con elevata accuratezza l'intero segnale misurato (tali algoritmi rendono il sistema scalabile, dal momento che permettono di ridurre il numero di pacchetti dati da raccogliere, fissato un certo livello di accuratezza che si vuole garantire sulla rappresentazione del segnale da monitorare); 2) la progettazione di protocolli di rete cooperativi, dove la cooperazione è utilizzata per raggiungere un obiettivo comune come l'individuazione di un incendio e/o l'aumento delle prestazioni della rete in termini di metriche quali il consumo energetico, la latenza, la probabilità di consegna.

Per quanto riguarda il primo punto, il mio studio indaga le potenzialità di *Compressive Sensing* (CS), una tecnica molto efficace per l'acquisizione e il recupero di segnali correlati, con l'obiettivo di progettare e implementare un sistema per la raccolta efficiente di elevate quantità di dati da reti (di sensori) distribuite. Tale sistema ha l'obiettivo di ricostruire segnali di grandi dimensioni, raccogliendo il minor numero di campioni necessario al recupero del segnale di interesse entro un fissato livello minimo di qualità. I passi della mia attività di ricerca possono essere riassunti come segue: 1.a) valutazione dell'applicabilità e dei benefici potenziali di CS in applicazioni di reti; 1.b) giustificazione dell'efficacia del recupero del segnale tramite CS, quando quest'ultimo è utilizzato in sinergia con la tecnica dell'analisi alle componenti principali (*Principal Component Analysis*, PCA) e caratteriz-

zazione dell'ottimalità del meccanismo di ricostruzione in funzione della statistica del segnale di ingresso; 1.c) progettazione di un algoritmo per la ricostruzione di segnali basato su CS, e successiva validazione del metodo proposto per mezzo di simulazioni (Matlab) e utilizzando tracce reali.

A proposito del secondo punto, invece, il mio lavoro si è focalizzato sullo studio di metodi distribuiti il cui obiettivo è quello di ottimizzare una metrica globale, nel senso delle prestazioni dell'intera rete di interesse. Nel dettaglio, nello scenario considerato vi sono più nodi che collaborano per minimizzare la somma di funzioni obiettivo locali, che in generale dipendono da variabili globali quali parametri protocollari o decisioni prese dai nodi stessi. Nel caso in cui le funzioni obiettivo locali siano convesse, è possibile utilizzare una tecnica che si basa sul metodo del subgradiente e algoritmi di consenso per mediare l'informazione proveniente da ogni nodo, e che garantisce la convergenza verso una soluzione di ottimo globale. In letteratura si trovano risultati di convergenza per tale tecnica che considerano solo il caso di operazioni sincrone tra nodi e modelli di mobilità senza memoria. La mia ricerca si è occupata di estendere tali risultati ad un contesto più ampio. I passi fondamentali del mio lavoro sono stati: 2.a) estensione dei risultati di convergenza all'ottimo per una classe più generale di modelli di mobilità (con memoria); 2.b) applicazione del metodo del subgradiente nel caso di operazioni asincrone tra nodi; 2.c) presentazione di un possibile scenario di applicazione di rete per validare l'analisi svolta e mostrare l'efficacia della tecnica di ottimizzazione distribuita considerata.

I risultati della mia ricerca si sono rivelati strumenti utili per l'ottimizzazione pratica di protocolli di rete e permettono di formulare raccomandazioni per la progettazione del complesso sistema integrato discusso sopra.

# List of Acronyms

**6LoWPAN** IPv6 over LoW Power wireless Area Networks

**BIC** Bayesian Information Criterion

**BN** Bayesian Network

**CS** Compressive Sensing

**DA** Data Aggregation

**DAG** Directed Acyclic Graph

**DCP** Data Collection Point

**DCT** Discrete Cosine Transform

**DEI** Dipartimento di Ingegneria dell'Informazione (University of Padova)

**DTN** Delay Tolerant Network

**EPFL** École Polytechnique Fédérale de Lausanne

**GPS** Global Positioning System

**ID** identification (number)

**INRIA** Institut national de recherche en informatique et automatique

**JSP** Java Server Page

**LSE** Least Square Error

**MAP** maximum a posteriori

**MIT** Massachusetts Institute of Technology

**NC** Network Coding

**NP-hard** Non-deterministic Polynomial-time hard

**OLS** Ordinary Least Square

**PhD** Philosophiae Doctor

**PCA** Principal Component Analysis

**RS** Random Sampling

**rhs** right hand side

**SP** Service Provider

**WSN** Wireless Sensor Network

Chapter 1

# Introduction

**Contents**



My research activity focused on the field of heterogeneous wireless networks and has been particularly inspired by the problem of sensing a city-wide environment through a large scale, partially distributed, mobile and low cost network (possibly composed of mobile phones or similar user's equipment).

This scenario, which has guided me during three years of research in networking, besides featuring a number of mobile phones, also consists of a mixture of embedded devices, which are expected to have on-board radio and sensing capabilities. Moreover, considering the fact that cell phones are becoming a communication hub in our daily life, this scenario can be thought of as generated by the integration of standard cellular systems with overlayed distributed networks such as Wireless Sensor Networks (WSNs). The ultimate goal of this is to "connect" everything has some communication capability, possibly providing self-configurability and self-adaptability of the network.

Practically speaking, however, in the aforementioned scenario temporal unavailability of some devices (due e.g., to hardware brakes or software bugs), mobility and/or overloaded servers can result in the impossibility of guaranteeing full connectivity over a wide network: thus, the overall infrastructure can appear as made of different sub-networks that can communicate with each other, but where no continuous connections can be assumed. As a

matter of fact, we are depicting a Delay Tolerant Network (DTN) scenario, where different heterogeneous, sparse and/or mobile wireless networks interact.

The above grand vision entails quite a few challenges, and during my research activity I have been focusing particularly on the following two: 1) the design of gathering and reconstruction algorithms for the efficient monitoring of signals over a wide WSN; 2) the design of cooperative networking protocols to increase the network performance in terms of optimization of given performance metrics, e.g., energy consumption, delivery time, delivery probability.

Concerning the first point, which is better introduced in Section 1.1, my study explores the capabilities of Compressive Sensing (CS), a technique that has been proved to be very effective for the compression and recovery of correlated signals, with the objective of designing and implementing a system for the efficient acquisition of large data sets in distributed (sensor) networks.

For the second point, which is introduced in Section 1.2, my work has been centered around distributed optimization methods whose objective is that of optimizing network wide (global) performance metrics.

A detailed organization of the thesis is presented and discussed in Section 1.3, whilst the list of publications generated from the research carried out during my PhD can be found in the Appendix 1.A.

## 1.1 Data Gathering and Recovery in Distributed Networks

The area of communication and protocol design for Wireless Sensor Networks (WSN) has been widely researched in the past few years. An important research topic which needs further investigation is in-network aggregation and data management to increase the efficiency of data gathering solutions (in terms of energy cost) while being able to measure large amounts of data with high accuracy. We note that often the proposed solutions for data aggregation are rather ad hoc [1], i.e., they are often specific to certain networks or signals and lack a solid theoretical foundation.

Gathering data while jointly performing compression, therefore, has been one of the first problems that spurred my research activity. One of the first studies addressing this issue is [2], which highlights the interdependence among bandwidth, decoding delay and the routing strategy employed. Under certain hypotheses of regularity of the observed process, justifiable from a physical point of view, the authors claim the feasibility of large-scale multi-hop networks from a transport capacity perspective. Classical source coding, suitable routing algorithms and re-encoding of data at relay nodes have been proposed as key ingredients for joint data gathering and compression. In fact, sensor network applications often involve multiple sources which are correlated both temporally and spatially. Subsequent work such as [3–7] proposed algorithms that involve collaboration among sensors to implement classical source coding (e.g., see [8–10]) in a distributed fashion. Along the same line, [11] shows the relation between routing and location of the aggregation/compression points according to the joint correlation of data among sources. In this way, it is possible to enforce the collaboration among nodes that are well suited to the statistical description of the signal measurements.

In [12], the authors consider a scenario where a number of different compression schemes are available at each node in the network. The selection of which compression scheme to use is based on the expected tradeoff between computation and communication costs; each node contributes to this goal through its local data processing. Following the same objective of minimizing the total energy for compressing and transporting information, [13] investigates a tunable data compression technique to deal with the tradeoff between computation and communication costs. In general, for a given connectivity structure, this technique needs to compute the optimal data gathering tree, which is topology dependent. Moreover, the authors show that when node entropies and the cost for compression are not known, a simple

greedy approximation of the Minimal Steiner Tree provides acceptable performance.

New methods for distributed sensing and compression have been developed based on the recent theory of Compressive Sensing [14–16]. CS is a novel data compression technique that exploits the inherent structure of some input data set to compress it by means of quasi-random matrices; recovery of the original data is achieved solving a convex optimization problem, i.e., $\ell_1$-norm minimization. In detail, if the compression matrix and the original data $\mathbf{x}$ have certain properties, $\mathbf{x}$ can be reconstructed from its compressed version $\mathbf{y}$, with high probability, by minimizing a distance metric over a solution space (whose dimension is equal to the difference between the size of the original data vector $\mathbf{x}$ and that of its compressed version $\mathbf{y}$).

CS was originally developed for the efficient storage and compression of digital images, which show high spatial correlation. Successively, there has been a growing interest in this technique also by the telecommunication community, as testified by [17]. In this context, during the first and the second year of PhD, my research activity has been mainly focused on the study of CS, following the believe that this technique could help in designing efficient solutions for WSN monitoring. The overall objective of my work has been the design of a very general framework exploiting CS, i.e., a solution suitable to be implemented as protocol for a monitoring application independently of the observed signal. This requirement is very appealing when we think about a network of nodes equipped with different sensors, and therefore capable of sensing different signals. We do not want protocols specifically designed for signals with given statistical characteristics, so that a node should select the right protocol up to the current sensed signal. Conversely, we would like to have a transmission protocol totally unaware of the observed signal characteristics, but nevertheless able to adapt to them.

Pursuing this objective, my research activity brought to the following original contributions:

- an analysis of the statistical distribution of real world WSN signals that legitimates the use of CS in actual wireless sensor networks;

- the design and performance evaluation of an effective and flexible framework integrating CS to achieve distributed sampling, data gathering and recovery of signals from actual WSN deployments;

This part of my research is thoroughly described and discussed in Chapter 2, whose organization is explained in Section 1.3.

## 1.2 Distributed Optimization

In networking applications, the performance of a Delay Tolerant Network is a global measure that depends on decisions (i.e., protocol rules) and variables (i.e., protocol parameters) at each network node. Hence, the optimization of any given network protocol can be described as a global optimization problem which is governed by the local actions taken by each node. As an example, the message delivery delay and the energy consumption under the gossip protocol [18, 19] depend on the message forwarding probabilities which can be locally and independently calculated by each node. To further complicate matters, local (but globally optimal) decisions at different nodes are not independent and the optimal configuration is in general heterogeneous and depends on the specific scenario, as different nodes have different roles in the network. Given this, it may be not possible to compute optimal protocol rules and parameters off-line prior to network deployment. In addition, the disconnected nature of DTNs calls for on-line and distributed approaches to optimization where, in practice, each node has access to local variables and rules which can only be set according to what occurs within its immediate surroundings (the visibility scope of the node).

The authors of [20] present a distributed solution to this problem for the case where the global optimization target $f$ can be expressed as sum of $M$ convex functions $f_i$ and each node $i$ only knows the corresponding function $f_i$, referred to as the *local objective function*. Many performance metrics of interest have this decomposition property. For example, this is the case of performance metrics related to nodes (e.g., energy consumption at each node) or to messages (e.g., delivery time, delivery probability, number of copies in the network). In either case, the metrics can naturally be expressed as a sum of local cost functions relative to each node. Convexity may be not guaranteed, but when this assumption does not hold the system converges in general to a sub-optimal but still desirable solution.

In the framework proposed in [20], and later extended in [21], nodes optimize their own local objective functions through a *sub-gradient method*, where they try to reach agreement on their local estimates by occasionally exchanging their local information and averaging it, like in a consensus problem [22, 23].

This approach, referred to as the *distributed sub-gradient method*, is particularly appealing in the context of Delay Tolerant Networks which are sparse and/or highly mobile wireless ad hoc networks where no continuous connectivity guarantee can be assumed, e.g., see [24] and [25]. The nature of such networks, in fact, intrinsically leads to the impossibility of

collecting, at low cost and at a single data processing point, the information needed to solve network optimization problems in a centralized fashion.

Within the framework proposed in [20, 21], however, the local estimate of each node is proven to converge to the optimal solution under certain assumptions and two of these appear to be particularly restrictive for practical use in DTN scenarios. Specifically, the node mobility process should have strict deterministic bounds on the inter-meeting times between nodes, see [20], or it should be memory-less, see [21]. Neither of these conditions is in general satisfied in a real network. Second, all nodes should update their estimates at the same time, but synchronicity is difficult to achieve in such a disconnected scenario. My research activity, in this context, has been spurred by the challenge of addressing and solving both these issues.

As positive outcomes, my work brought to the following original contributions:

- proof of convergence of the distributed sub-gradient method when a general class of mobility models with memory is considered;
- analysis and practical guidelines for the implementation of the distributed sub-gradient method under asynchronous operations.

Furthermore, the validity of the analysis carried out is demonstrated through an example. In particular, the proposed distributed optimization technique is effectively applied for the dissemination of dynamic content in a DTN such as a mobile social network.

This part of my research is thoroughly described and discussed in Chapter 3, whose organization is explained in Section 1.3.

## 1.3 Discussion and Organization of the Thesis

This thesis is divided in two parts, corresponding to the two main research topics that I addressed during my PhD.

Specifically, Chapter 2 presents and discusses the principal steps and outcomes of my research activity on Compressive Sensing for WSNs. This research activity has been carried out in collaboration with Dr. Giorgio Quer from the University of Padova. A different perspective on the same work (and some additional details which fall better within his area of expertise) can be found in [26].

In Section 2.1 a general overview on the Compressive Sensing technique is presented at first. In particular, Section 2.1.1 gives an overview on the mathematical background of CS, whilst 2.1.2 briefly discusses the algorithms that allow to actually implement this technique. Then, in Section 2.3, is carried out a preliminary investigation to understand how CS can be efficiently used for signals that are typical of WSNs. In detail, the achievable performance through standard processing techniques (e.g., Discrete Cosine Transform, DCT) are tested; in this context, the main objective is to understand the implications of the topological/connectivity structure of the networks on the CS performance. Sections 2.4 and 2.5 contain, instead, the main contribution of the research presented in the chapter. This encompasses the design and implementation of a recovery technique exploiting CS that iteratively adapts to the time varying characteristics of the signal of interest (in time and space) and that does so starting from incomplete observations of the signal itself. This of course provides benefits in terms of energy expenditure, network congestion and so forth. Finally, in Section 2.6 we discuss the obtained research results and their effects for an actual implementation of CS in real WSNs.

Chapter 3, instead, describes my work on Distributed Subgradient Methods for Delay Tolerant Networks. This research activity has been carried out during my PhD visiting period at the *Institut national de recherche en informatique et automatique* (INRIA), Sophia Antipolis, France, under the supervision of Dr. Giovanni Neglia, MAESTRO team.

This chapter is organized as follows. In Section 3.1 the distributed sub-gradient method is reviewed, whilst Section 3.2 shows how to apply it to DTNs and motivates the presented work. As original contributions, Section 3.3 extends the theoretical results in [20] and [21] to more general network mobility models and in Section 3.4, the distributed sub-gradient method's framework is extended to cope with asynchronous node operations. Hence, in

Section 3.5, a possible DTN application exploiting the distributed sub-gradient method is illustrated. Finally, in Section 3.6 we discuss how the presented analysis can be used for the distributed optimization of practical network protocols.

## 1.A List of Publications

**Publications on Compressive Sensing for Wireless Sensor Networks**

- QUER G., **MASIERO R.**, MUNARETTO D., ROSSI M., WIDMER J., ZORZI M. (2009). *On the interplay between routing and signal representation for Compressive Sensing in wireless sensor networks. In: Information Theory and Applications Workshop*, 2009. San Diego, CA, 8-13 Feb. 2009, p. 206-215, ISBN/ISSN: 978-1-4244-3990-4

- **MASIERO R.**, QUER G., ROSSI M., ZORZI M. (2009). *A Bayesian analysis of Compressive Sensing data recovery in Wireless Sensor Networks*. In: Ultra Modern Telecommunications & Workshops, 2009. ICUMT '09. International Conference on. St. Petersburg, 12-14 Oct. 2009, p. 1-6, ISBN/ISSN: 978-1-4244-3942-3

- **MASIERO R.**, QUER G., MUNARETTO D., ROSSI M., WIDMER J., ZORZI M. (2009). *Data Acquisition through Joint Compressive Sensing and Principal Component Analysis*. In: Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE. Honolulu, HI, Nov. 30 2009-Dec. 4 2009, p. 1-6, ISBN/ISSN: 978-1-4244-4148-8 / 1930-529X

- QUER G., ZORDAN D., **MASIERO R.**, ZORZI M., ROSSI M. (2010). *WSN-Control: Signal Reconstruction through Compressive Sensing in Wireless Sensor Networks*. In: LCN 2010, IEEE Proceedings of. Denver, CO, US, 11-14 Oct. 2010

- **MASIERO R.**, QUER G., PILLONETTO G., ROSSI M., ZORZI M. (2011). *Sampling and Recovery with Compressive Sensing in Real Wireless Sensor Networks*. In: under submission to IEEE Transactions on Wireless Communications

- QUER G., **MASIERO R.**, ROSSI M., ZORZI M. (2011). *SCoRe1: Sensing Compression and Recovery through ON-line Estimation for Wireless Sensor Networks*. In: under submission to IEEE Transactions on Wireless Communications

**Publications on Distributed Sub-gradient Method for Delay Tolerant Networks**

- **MASIERO R.**, NEGLIA G. (2011). *Distributed Subgradient Methods for Delay Tolerant Networks*. In: The IEEE International Conference on Computer Communications (IEEE INFOCOM 2011), accepted for publication. China, Shanghai, April, 10-15

- **MASIERO R.**, NEGLIA G. (2010). *Distributed Sub-gradient Methods for Delay Tolerant Networks*. In: available on-line at http://hal.inria.fr/inria-00506485/en/, Research Report RR-7345

- **MASIERO R.**, ROSSI M., NEGLIA G. (2011). *Distributed Sub-gradient Methods for Delay Tolerant Networks*. In: under submission to IEEE Journal on Selected Areas in Communications

**Other Publications**

- **MASIERO R.**, MUNARETTO D., ROSSI M., WIDMER J., ZORZI M. (2009). *A Note on the Buffer Overlap Among Nodes Performing Random Network Coding in Wireless Ad Hoc Networks. In: Vehicular Technology Conference*, 2009. VTC Spring 2009. IEEE 69th. Barcelona, 26-29 April 2009, p. 1-5, ISBN/ISSN: 978-1-4244-2517-4 / 1550-2252

Chapter 2

# Compressive Sensing for Wireless Sensor Networks

## Contents

In this chapter the principal steps and outcomes of my research activity on Compressive Sensing (CS) for Wireless Sensor Networks (WSNs) are presented and discussed. To properly introduce them, an overview of Compressive Sensing is presented in Section 2.1.

The research activity reported in this chapter has been carried out in collaboration with Dr. Giorgio Quer from the University of Padova, thus a different perspective on the same work (and some additional details which fall better within his area of expertise) can be found in [26].

As previously mentioned, the overall objective of our research project is to design an algorithm for signal reconstruction based on CS. This algorithm is presented in Section 2.5 and is composed of several functional blocks: 1) the reconstruction engine, 2) a quality checker and 3) a control logic. The reconstruction engine is responsible for the approximation of the original input signal starting from its incomplete measurements; the quality checker continuously monitors the reconstruction quality and the control logic adapts the data gathering protocol so as to keep the reconstruction error below a minimum target level. However, before arriving to the aforementioned framework definition we need to address different issues.

In particular, as first step, we have to evaluate the potential benefits brought about by CS in distributed networks for an ideal protocol, by only taking the network topology into account. Here, our purpose is to assess the potential energy savings of CS over different topologies and to understand the implications of the topological/connectivity structure of the network on its performance. In parallel with this study, we also look at different standard processing techniques (e.g., Discrete Cosine Transform, DCT), to understand which is the best way to process real WSN signals so that they can take fully advantage of CS. Both these preliminary studies are reviewed and discussed in Section 2.3.

Moreover, as fundamental contribution of our research, we have to provide a sound

theoretical justification of the effectiveness of CS recovery when exploited in our proposed monitoring framework. In this analysis, we describe CS under a Bayesian perspective and study the statistics of real WSN signals. This is done in 2.4. A comparison of the CS performance against state-of-the-art signal reconstruction schemes is presented in Section 2.5, right after the description of our framework called SCoRe1 (Sensing, Compression an Recovery through ON-line Estimation), proposed to iterative monitor WSN signals.

## 2.1 Overview on Compressive Sensing

Compressive Sensing is a recent method to represent compressible signals with significantly fewer samples than required by the Nyquist Theorem. Reconstruction of the original data is possible with high probability through dedicated non-linear recovery algorithms without loss of information in the absence of noise and with excellent accuracy when observations are noisy [27]. A general introduction to this technique can be found in [15, 28–30].

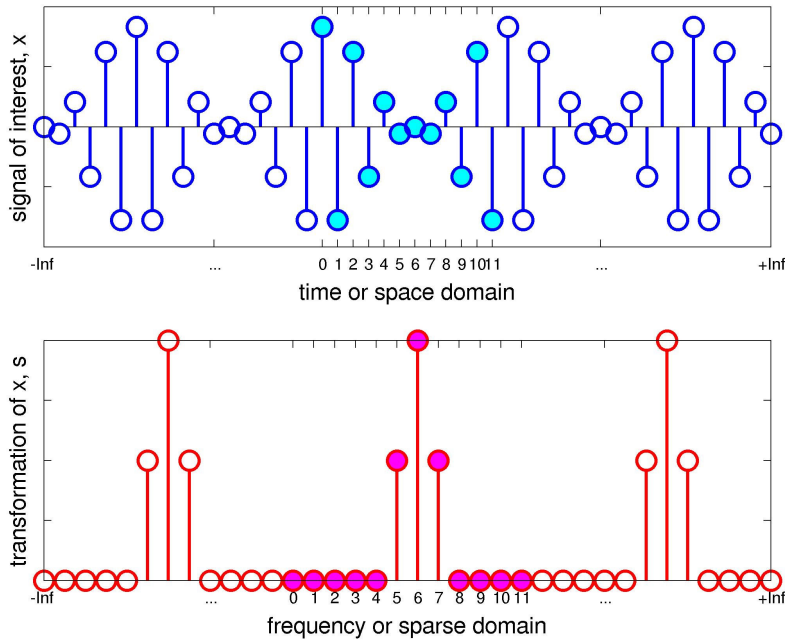



**Figure 2.1.** *Example to illustrate the Shannon-Nyquist Theorem. Top graph: considered signal,* $\mathbf{x}$ *(color-filled circles) and its periodic repetition. Bottom graph: period* $\mathbf{s}$ *(color-filled circles) of the frequency response computed from the periodic repetition of* $\mathbf{x}$*.*

Before proceeding further, we present here a simple example to explain the potential and novelty of CS. Most tutorial papers on this technique do the following claim: CS allows us to perfectly recover a given signal under-sampling it as respect to the classical Nyquist-Shannon Sampling Theorem [31]. The example we are going to discuss, clarifies better this appealing feature of CS. First, let us recall the classical Nyquist-Shannon Sampling Theorem, that we rephrase as

**Theorem 1** (Sampling Theorem)**.** *If a signal* $\mathbf{x}$ *contains no frequencies higher than* $B$ *Hertz,* $\mathbf{x}$ *can be completely determined by an its sampled version* $\mathbf{y}$ *made of a series of points spaced at most* $1/(2B)$ *seconds apart.*

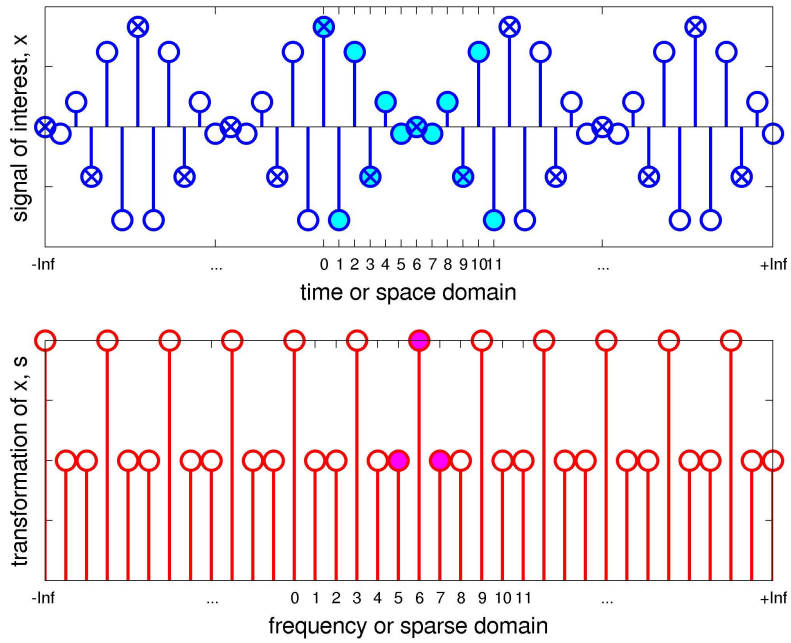



**Figure 2.2.** *Example to illustrate the Shannon-Nyquist Theorem. Top graph: sampled version* $\mathbf{y}$ *(crosses) of the considered signal,* $\mathbf{x}$ *(color-filled circles) and its periodic repetition. Bottom graph: period* $\mathbf{s}$ *(color-filled circles) of the frequency response computed from the sampled periodic repetition of* $\mathbf{x}$*.*

We can illustrate Theorem 1 through Figures 2.1 and 2.2. In Figure 2.1 we represent with color-filled circles a signal of interest $\mathbf{x}$, made of $N = 12$ elements (top graph). These elements are equally spaced, being $T = 0.25$ seconds apart one from another. As commonly done in signal processing, an equivalent discrete representation in frequency of $\mathbf{x}$ can be obtained by just pretending to repeat $\mathbf{x}$ infinitely often in time, thus building a periodic signal. Since we have 12 samples, and two consecutive samples are separated by $0.25$ seconds, $\mathbf{x}$ can be thought of as the period of a periodic signal of period $T_p = 3$ seconds. Therefore, in frequency the considered signal turns out to be periodic of period $F_p = 1/T = 4$ Hertz, and its components spaced $F = 1/T_p = 1/3$ Hertz apart (Figure 2.1, bottom graph). Note that in this example the bandwidth of $\mathbf{x}$ is $B = 2/3$. Thus, according to the Sampling Theorem we can handle the signal $\mathbf{x}$ (e.g., for storage or transmission) considering only $4$ samples out of the $12$ which composed it, i.e., we can considered, instead of $\mathbf{x}$, its sampled version $\mathbf{y}$ that is represented in Figure 2.2 (top graph) with crosses. In detail, the elements of $\mathbf{y}$ are spaced $T_s = 3/4$ seconds apart and $T_s \leq 1/(2B)$; thus, in accordance to Theorem 1, $\mathbf{y}$ completely determines $\mathbf{x}$. To further validate the bijective correspondence between $\mathbf{x}$ and $\mathbf{y}$ in this case, the bottom graph of Figure 2.2 shows that we have not signal distortion in frequency. In

fact, because the condition of the Sampling Theorem is not violated, we observe not aliasing (i.e., disturbing superpositions of far apart frequency signal components) in the frequency response of $\mathbf{y}$, which is simply a periodic repetition of the frequency response of $\mathbf{x}$. In summary, in this first considered example the Nyquist-Shannon Theorem alone guarantees us that we can perfectly recover $\mathbf{x}$ from $\mathbf{y}$.

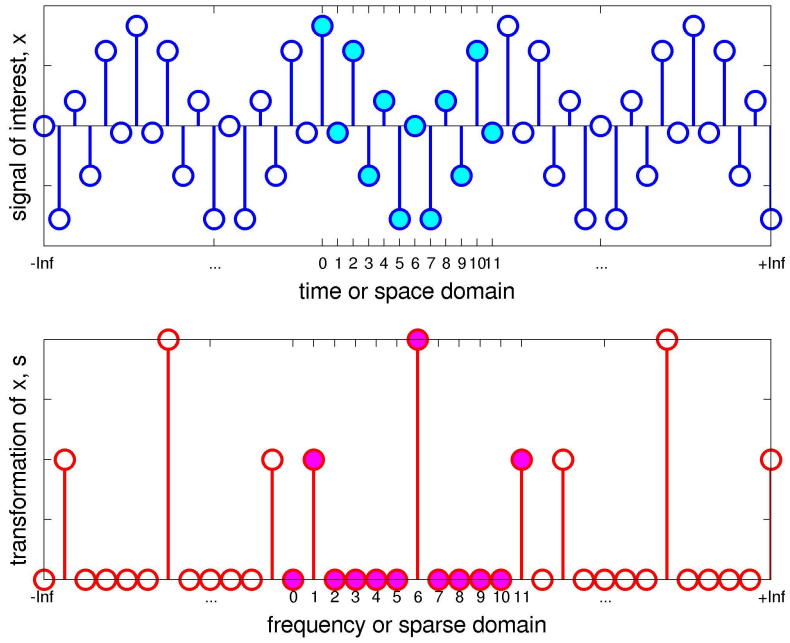



**Figure 2.3.** *Example to illustrate the potential and novelty of CS. Top graph: considered signal,* $\mathbf{x}$ *(color-filled circles) and its periodic repetition. Bottom graph: period* $\mathbf{s}$ *(color-filled circles) of the frequency response computed from the periodic repetition of* $\mathbf{x}$*.*

However, if we now consider the example in Figure 2.3, we can easily figure out that in this case we cannot reduce the number of samples of the original signal $\mathbf{x}$ (top graph) according to the classical theory. That is because, even if we pick just one sample every two, we will violate the fundamental condition of the Sampling Theorem: in fact, the signal bandwidth is now $B = 10/3$ and $2T = 1/2$ is greater than $1/(2B)$. Therefore, the Nyquist-Shannon theory teaches us that we should use all the $12$ elements of $\mathbf{x}$ to handle it. Differently, CS theory allow us to do better: namely, since the original signal $\mathbf{x}$ has a sparse representation $\mathbf{s}$ with only $M = 3$ significant components, $\mathbf{x}$ can be recovered from an its compressed version $\mathbf{y}$ made of $L > M \log N \simeq 7.45$ elements only; further, for larger $N$ we can achieve a bigger gain as respect to the classical sampling theory.

Clarified with this simple example the appeal of Compressive Sensing, we are going to present in Section 2.1.1 a mathematical overview on CS, then in Section 2.1.2 we briefly discuss about the key concepts of the algorithms that solve the convex optimization problem at the core of this technique.

### 2.1.1 Mathematical Background

For the sake of exposition, in this section we consider signals representable through one dimensional vectors[1] $\mathbf{x}$ in $\mathbb{R}^N$, where $N$ is the vector length. We assume that these vectors are such that there exists a transformation under which they are sparse. Specifically, there must exist an invertible *transformation matrix* $\mathbf{\Psi}$ of size $N \times N$ such that we can write

$$\mathbf{x} = \mathbf{\Psi}\mathbf{s} \tag{2.1}$$

and $\mathbf{s}$ is sparse. We say that a vector $\mathbf{s}$ is $M$-sparse if it has at most $M$ non-zero entries, with $M < N$. From a practical point of view, $\mathbf{s}$ is said to be $M$-sparse when it has only $M$ significant components, while the other $N - M$ are negligible with respect to the average energy per component, defined as $E_s = \frac{1}{N}\|\mathbf{s}\|_{\ell_2}$ whit $\|\cdot\|_{\ell_2}$ the $\ell_2$-norm of a vector, i.e., for a given vector $\mathbf{a}$ of $N$ elements, $\|\mathbf{a}\|_{\ell_2} = \sqrt{\sum_{i=1}^{N} a_i^2}$. Note that, if we know the matrix $\mathbf{\Psi}$, it is equivalent to have (or calculate a good approximation of) either of the two vectors $\mathbf{x}$ or $\mathbf{s}$, as due to Equation (2.1) there is an one-to-one mapping between them.

The compression of $\mathbf{x}$ entails a linear combination[2] of its elements through a further *measurement matrix* $\mathbf{\Phi}$ of size $L \times N$, with $L < N$. The compressed version of $\mathbf{x}$ is thus obtained as

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x}\ . \tag{2.2}$$

Now, using (2.1) we can write

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{s} \stackrel{def}{=} \mathbf{A}\mathbf{s}\ . \tag{2.3}$$

In general this system is both ill-posed and ill-conditioned as the number of equations $L$ is smaller than the number of variables $N$ and small variations of the input signal $\mathbf{s}$ can produce large variations of the output $\mathbf{y}$, respectively. However, if $\mathbf{s}$ is sparse, it has been shown

[1] In this chapter, all the real valued vectors are assumed to be column vectors unless otherwise specified.

[2] Note that according to the introduced formalism, a sampling operation can be viewed as a special case of linear combination. In this case, the matrix $\mathbf{\Phi}$ would be with a single one for each row, at most a single one for each column and the remaining elements set to zero.

that (2.3) can be inverted with high probability through the use of special optimization techniques [16, 32, 33]. These allow us to retrieve $\mathbf{s}$, whereas the original signal $\mathbf{x}$ is found by inverting (2.1), i.e., $\mathbf{s} = \mathbf{\Psi}^{-1}\mathbf{x}$.

Furthermore, when the matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are orthonormal, we can define the following quantity

$$\mu(\mathbf{\Phi}, \mathbf{\Psi}) = \sqrt{N} \max_{1 \leq i,j \leq N} |\langle \boldsymbol{\phi}_i, \boldsymbol{\psi}_j \rangle| \ , \tag{2.4}$$

where for any two column vectors $\mathbf{a}$ and $\mathbf{b}$ of the same length, we define $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T\mathbf{b}$. In Equation (2.4), $\boldsymbol{\phi}_i$ is the $i-$th column of the matrices $\mathbf{\Phi}$, whilst $\boldsymbol{\psi}_j$ is the $j-$th column of $\mathbf{\Psi}$. $\mu(\mathbf{\Phi}, \mathbf{\Psi})$ is called *coherence* of the matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$; it can assume values in the interval $[1, \sqrt{N}]$ and plays an important role in the CS theory because it bounds the minimum number of projections (i.e., the dimension of $\mathbf{y}$) required to recover the sparse signal $\mathbf{s}$ (and therefore the original signal $\mathbf{x}$). In detail, we have that $L$ must be greater or equal than $C\mu^2(\mathbf{\Phi}, \mathbf{\Psi})M \log N$ with $C$ a properly chosen constant value [16, 28].

Next, we illustrate the reconstruction process. Given a solution $\mathbf{s}_p$ of (2.3) such that $\mathbf{A}\mathbf{s}_p = \mathbf{y}$ and given the null space of matrix $\mathbf{A}$, $\mathcal{N}(\mathbf{A})$ of dimension $N - L$, any vector $\mathbf{s}' = \mathbf{s}_p + \mathbf{s}_\perp$, where $\mathbf{s}_\perp \in \mathcal{N}(\mathbf{A})$, is also a solution of (2.3). However, in [16] it is proved that: A1) if any set of $T \leq 2M$ columns of the matrix $\mathbf{A}$ approximatively behaves as an orthonormal system and A2) if $\mathbf{s}$ is $M$-sparse with $M$ smaller than a given threshold, then the original $\mathbf{s}$ is the sparsest admissible solution of (2.3). The solution that we find in this way, that we call $\widehat{\mathbf{s}}$, is equal to the original $\mathbf{s}$ if Assumptions A1 and A2 hold. Otherwise, there will be a reconstruction error that decreases for increasing $L$. Of course, when $L = N$ and $\mathbf{A}$ is full rank, the only solution of this system is $\mathbf{s}$ and it can be obtained through standard matrix inversion.

A generalization of the CS technique for 2D signals is detailed in the Appendix 2.A.

### 2.1.2 Algorithms for CS

In the previous section we have seen that at the core of Compressive Sensing, in order to reconstruct the original signal $\mathbf{x}$, there is the problem of inverting the ill-posed system defined by Equation (2.3). Under the assumption that $\mathbf{s}$ has a certain degree of sparsity, inverting (2.3) is equivalent to solving the following

**Problem 1** ($\ell_0$-norm minimization)**.**

$$\widehat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \|\mathbf{s}\|_{\ell_0} \quad \textit{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{s} \ , \tag{2.5}$$

*where* $\|\cdot\|_{\ell_0}$ *is the* $\ell_0$*-norm of a vector, i.e., for a given vector* $\mathbf{a}$ *of* $N$ *elements, the norm computed*[3] *as* $\|\mathbf{a}\|_{\ell_0} = \sum_{i=1}^{N} \mathbb{1}\{a_i \neq 0\}$.

Unfortunately, solving (2.5) is NP-hard. However, under specific assumptions on the matrix $\mathbf{A}$, Problem 1 has been proven [15] to be also equivalent to the following convex minimization problem

**Problem 2** ($\ell_1$-norm minimization)**.**

$$\widehat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \|\mathbf{s}\|_{\ell_1} \quad \textit{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{s} \;, \tag{2.6}$$

*where* $\|\cdot\|_{\ell_1}$ *is the* $\ell_1$*-norm of a vector, i.e., for a given vector* $\mathbf{a}$ *of* $N$ *elements,* $\|\mathbf{a}\|_{\ell_1} = \sum_{i=1}^{N} |a_i|$.

As a matter of fact, there is a wide literature that aims to solve both Problem 1 and Problem 2 through "relaxed" versions of them: in this context, a standard approach that attempts to reconstruct $\mathbf{s}$ from (2.3) can be formalized as [33]

**Problem 3** (Standard minimization approach)**.**

$$\widehat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} f(\mathbf{s}) \quad \textit{subject to} \quad \|\mathbf{A}\mathbf{s} - \mathbf{y}\|_{\ell_2} \leq \epsilon \;. \tag{2.7}$$

In Problem 3, $\epsilon^2$ can also be interpreted as an estimated upper bound on the noise power affecting the measurements $\mathbf{y}$; the choice of the regularization (convex) function $f(\cdot)$, instead, depends on prior assumptions about the input $\mathbf{s}$: in particular, if $\mathbf{s}$ is (approximately) sparse, an appropriate function is the $\ell_1$-norm, as advocate by the CS theory and extensively discussed in Section 2.4. Note that if in Problem 3 we set $\epsilon = 0$ and choose $f(\cdot) = \|\cdot\|_{\ell_0}$ or $f(\cdot) = \|\cdot\|_{\ell_1}$ , by solving (2.7) we actually solve Problem 1 or Problem 2, respectively.

As particular case of Problem 3, in this section we will describe especially a method to solve the following

**Problem 4** (Quadratically constrained $\ell_1$-norm minimization [33])**.**

$$\widehat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \|\mathbf{s}\|_{\ell_1} \quad \textit{subject to} \quad \|\mathbf{A}\mathbf{s} - \mathbf{y}\|_{\ell_2} \leq \epsilon \;. \tag{2.8}$$

Solvers based on the solution of Problem 4, in fact, have been extensively used for the study presented in this chapter, see Sections 2.3 and 2.5. The algorithm proposed in [33] to solve Problem 4 is called NESTA and is based on the Nesterov minimization method [34], that we

[3] For any value $x \in \mathbb{R}$, the function $\mathbb{1}\{x \neq 0\}$ is zero if $x = 0$ and 1 otherwise.

describe in the following. Subsequently, we discuss the extension of this method to non-smooth functions and finally we explain how it is applied to CS, thus obtaining NESTA.

More details on this topic can be found in [33], where it is possible to find a nice discussion and performance comparisons among (2.8) and other two common optimization approaches that solve the sparse reconstruction problem. In particular, these approaches are

**Problem 5** (Basis Pursuit Denoising Problem [35])**.**

$$\widehat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}}\, \lambda\|\mathbf{s}\|_{\ell_1} + \frac{1}{2}\|\mathbf{A}\mathbf{s} - \mathbf{y}\|_{\ell_2}^2 \,, \tag{2.9}$$

and

**Problem 6** (Lasso [36])**.**

$$\widehat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}}\, \|\mathbf{A}\mathbf{s} - \mathbf{y}\|_{\ell_2} \quad \textit{subject to} \quad \|\mathbf{s}\|_{\ell_1} \leq \tau \,. \tag{2.10}$$

In Problems 4–6, $\epsilon$, $\lambda$ and $\tau$ are optimization parameters that must be properly tuned in the corresponding algorithms; in any case, standard optimization theory [37] shows that since $\epsilon$, $\lambda$ and $\tau$ obey some special relationships, Problems 4–6 are practically equivalent.

**Nesterov minimization:** this method solves convex optimization problems of the type

$$\min_{\mathbf{x}\in\mathcal{Q}_p} f(\mathbf{x}) \,, \tag{2.11}$$

where the convex function to minimize, $f(\mathbf{x}) : \mathcal{Q}_p \to \mathbb{R}$, is defined in the convex set $\mathcal{Q}_p \subseteq \mathbb{R}^N$, e.g., of the form

$$\mathcal{Q}_p = \{\mathbf{x} : \mathbf{b} = \mathbf{Q}\mathbf{x}\} \,, \tag{2.12}$$

where $\mathbf{Q}$ is an $L \times N$ matrix, with $L \leq N$, and $\mathbf{b} \in \mathbb{R}^L$ is a given constant vector. Moreover, the function $f(\mathbf{x})$ must be smooth, i.e., it must be differentiable and its gradient must be Lipschitz: for any pair $\mathbf{x}_1 \in \operatorname{dom} f$ and $\mathbf{x}_2 \in \operatorname{dom} f$, it must hold

$$\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\|_{\ell_2} \leq C\|\mathbf{x}_1 - \mathbf{x}_2\|_{\ell_2} \,, \tag{2.13}$$

where $C > 0$ is a constant [34]. The algorithm proposed by Nesterov to solve (2.11) is listed in Table 2.1 and discussed in the following:

0. Initialize $\mathbf{x}_0$ to an allowable value. A possible initialization choice for $\mathbf{x}_0$ is $\mathbf{x}_0 = \mathbf{Q}^T\mathbf{b}$. Set $t = 0$.

**0.** Initialize $\mathbf{x}_0$.

**For** $t \geq 0$,

**1.** Compute $\nabla f(\mathbf{x}_t)$.

**2.** Compute $\mathbf{r}_{t+1}$:

$$\mathbf{r}_{t+1} = \underset{\mathbf{x}\in\mathcal{Q}_p}{\operatorname{argmin}} \left\{ p(\mathbf{x}, \mathbf{x}_t) + \langle \nabla f(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t \rangle \right\}.$$

**3.** Compute $\mathbf{z}_{t+1}$:

$$\mathbf{z}_{t+1} = \underset{\mathbf{x}\in\mathcal{Q}_p}{\operatorname{argmin}} \left\{ p(\mathbf{x}, \mathbf{x}_0) + \sum_{i=0}^{t} \alpha_i \langle \nabla f(\mathbf{x}_i), \mathbf{x} - \mathbf{x}_i \rangle \right\}.$$

**4.** Update $\mathbf{x}_{t+1}$:

$$\mathbf{x}_{t+1} = \tau_t \mathbf{z}_{t+1} + (1 - \tau_t)\mathbf{r}_{t+1} \ .$$

**5.** Stop if a given criterion is satisfied.

**Table 2.1.** *The Nesterov minimization algorithm for smooth functions.*

1. Computation of the gradient of $f(\mathbf{x}_t)$.

2. Computation of $\mathbf{r}_{t+1}$: $\mathbf{r}_{t+1}$ is a first sequence of vectors that converges towards the minimum of $f(\mathbf{x})$. The first term $p(\mathbf{x}, \mathbf{x}_t)$ is a proximity function (also referred to as *penalty function*) weighing more those points that are farther away from the current solution $\mathbf{x}_t$. A common choice is

$$p(\mathbf{x}, \mathbf{x}_t) = \frac{C}{2} \|\mathbf{x} - \mathbf{x}_t\|_{\ell_2}^2 \ . \tag{2.14}$$

   The second term corresponds to a gradient descent minimization with step $|\mathbf{x} - \mathbf{x}_t|$. Note that the step size is controlled by the first term, which penalizes large deviations from $\mathbf{x}_t$.

3. Computation of $\mathbf{z}_{t+1}$: $\mathbf{z}_{t+1}$ is a second sequence of vectors that also converges to the minimum of $f(\mathbf{x})$. The first term is equal to (2.14) but with $\mathbf{x}_0$ in place of $\mathbf{x}_t$. The second term corresponds to a gradient descent minimization accounting for all previous partial solutions $\mathbf{x}_i$, $i \leq t$.

4. The solution is updated as a weighted average of $\mathbf{r}_t$ and $\mathbf{z}_t$, using a suitable combination coefficient $\tau_t$.

5. A possible stopping criterion, adopted also in [33], is the following. Let $\overline{f}(\cdot)$ be the average of $f(\cdot)$ during the last ten iterations, namely

$$\overline{f}(\mathbf{x}_t) = \frac{1}{\min\{10,t\}} \sum_{i=1}^{\min\{10,t\}} f(\mathbf{x}_{t-i}) \; . \tag{2.15}$$

The algorithm is terminated when

$$\Delta f = \frac{|f(\mathbf{x}_t) - \overline{f}(\mathbf{x}_t)|}{\overline{f}(\mathbf{x}_t)} < \delta \; . \tag{2.16}$$

The coefficients $\alpha_t$, $\tau_t$ must be chosen to guarantee convergence, see [38]. The constant $\delta$ can be set arbitrarily upon the desired accuracy; typical values are $\delta \in \{10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$, see [33].

**Application of Nesterov minimization to CS:** reference [38] extended the Nesterov algorithms to *non-smooth* functions, showing that this extension is possible when these functions can be re-written as a maximization problem. Subsequently, with the NESTA algorithm [33], the theory of [38] has been applied to CS. In detail, (2.8) is re-written as

$$\min_{\mathbf{s} \in \mathcal{Q}'_p} \|\mathbf{s}\|_{\ell_1} \; , \tag{2.17}$$

where $\mathcal{Q}'_p$ is the convex set defined as

$$\mathcal{Q}'_p = \left\{ \mathbf{s} : \|\mathbf{y} - \mathbf{A}\mathbf{s}\|_{\ell_2} \leq \epsilon \right\} , \tag{2.18}$$

where $\mathbf{s} \in \mathbb{R}^N$ is a sparse vector with only $M$ significant elements with $M \ll N$, $\epsilon \geq 0$ is a small number and $\mathbf{A}$ is an $L \times N$ and real matrix having linearly independent rows ($M \leq L \leq N$). In [33], $\|\mathbf{s}\|_{\ell_1}$ is re-written as a maximization problem, i.e.,

$$\|\mathbf{s}\|_{\ell_1} = \max_{\mathbf{u} \in Q_d} \langle \mathbf{u}, \mathbf{s} \rangle \; , \tag{2.19}$$

where $Q_d \subseteq \mathbb{R}^N$ is the unit sphere defined as

$$Q_d = \{ \mathbf{u} : \|\mathbf{u}\|_\infty \leq 1 \} \; . \tag{2.20}$$

Hence, $\|\mathbf{s}\|_{\ell_1}$ is approximated by the smooth function

$$\|\mathbf{s}\|_{\ell_1} \simeq f_\mu(\mathbf{s}) = \max_{\mathbf{u} \in Q_d} \left\{ \langle \mathbf{u}, \mathbf{s} \rangle - \frac{\mu}{2} \|\mathbf{u}\|^2_{\ell_2} \right\} , \tag{2.21}$$

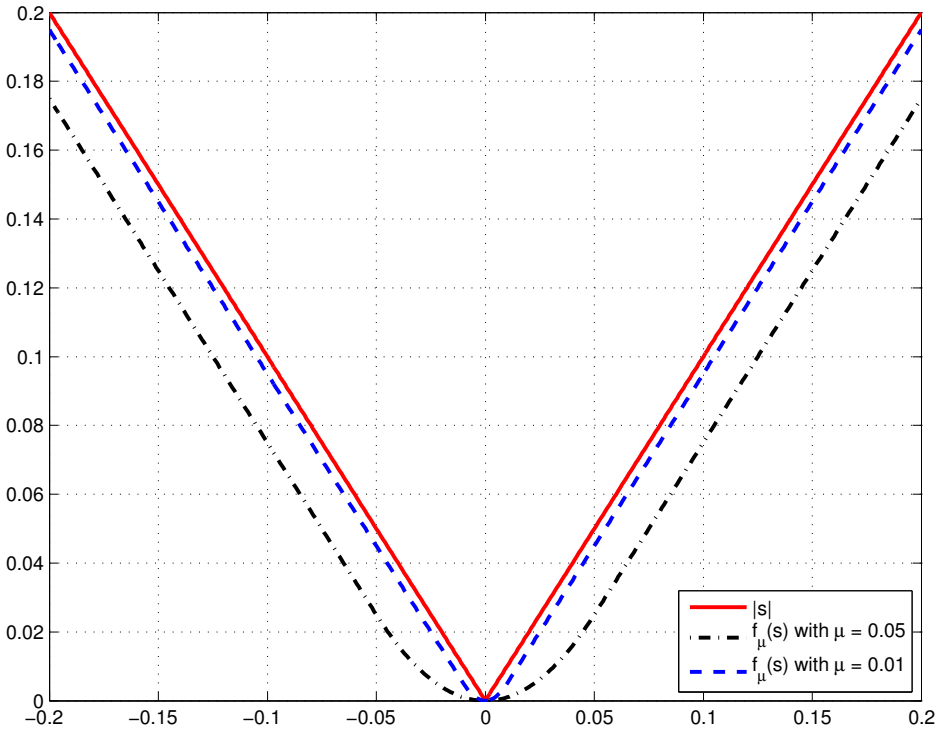



**Figure 2.4.** *The Huber function. Example of approximation of* $|s|$, $s \in \mathbb{R}$, *with* $\mu = 0.05$ *and* $\mu = 0.01$.

where $f_\mu(\mathbf{s})$ is generally known as the "Huber function". In Figure 2.4 is shown how $f_\mu(\mathbf{s})$ can approximate the $\ell_1$-norm depending on the value set for the parameter $\mu$ (in this figure we illustrate the case of the $\ell_1$-norm in 1-dimension): roughly speaking, we can obtain better approximations of the $\ell_1$-norm by lowering $\mu$.

It can be shown that $\nabla f_\mu(\mathbf{s})$ is Lipschitz with constant $C = 1/\mu$ and thus the Nesterov optimization algorithm can be applied to such function. In conclusion, the NESTA method of [33] amounts to solving

$$\min_{\mathbf{s}\in\mathcal{Q}'_p} \max_{\mathbf{u}\in Q_d} \left\{ \langle \mathbf{u}, \mathbf{s} \rangle - \frac{\mu}{2} \|\mathbf{u}\|_{\ell_2}^2 \right\} . \tag{2.22}$$

Note that (2.22) can now be tackled using the algorithm in Table 2.1, where the inner maximization problem (2.21) can be solved in linear time through the sequential evaluation of the elements of $\mathbf{u}$. In fact, defining $\widehat{\mathbf{u}}$ as

$$\widehat{\mathbf{u}} = \operatorname*{argmax}_{\mathbf{u}\in Q_d} \left\{ \langle \mathbf{u}, \mathbf{s} \rangle - \frac{\mu}{2} \|\mathbf{u}\|_{\ell_2}^2 \right\} \tag{2.23}$$

we have

$$\widehat{u}_i = \begin{cases} s_i/\mu & \text{for } |s_i| \leq \mu \\ +1 & \text{for } |s_i| > \mu \text{ and } s_i > 0 \\ -1 & \text{for } |s_i| > \mu \text{ and } s_i < 0 \end{cases} , \; i = 1, \ldots, N \; . \tag{2.24}$$

## 2.2 Application in WSNs

Quite a few number of papers have been written about the mathematical foundations of CS and its application to image processing, e.g., [15, 16, 29]. However, when my research activity moved its first steps, also the study of CS for networking problems was still in its infancy. In this section, we summarize the work related to the research activity presented in this chapter by discussing contributions that deal with the application of CS to data gathering in wireless networks.

An early contribution is [39], where the authors use Compressive Sensing and propose a distributed communication scheme for the energy efficient estimation of the data in a wireless sensor network. Properly chosen random projections of this data are used for reconstruction at the sink. The goal of this scheme is to use CS in a WSN for improving the performance of data gathering. The authors consider a multi-hop communication, but in-network data processing and compression are not used and data packets are transmitted directly to the sink. This requires phase synchronization among nodes. An extended version of this work can be found in [40].

[41] proposes an early application involving CS for network monitoring. The considered simulation scenario is a network where a small set of nodes fails. The goal is to correctly identify these nodes through the transmission of random projections (i.e., linear combinations) indicating the status of the nodes. However, these random projections are obtained by means of a pre-distribution phase (via simple gossiping algorithms), which is very expensive in terms of number of transmissions.

[42] also addresses the problem of gathering data in distributed WSNs through multi-hop routing. In detail, tree topologies are exploited for data gathering and routing, and the Wavelet transformation [43] is used for data compression. Even though CS is presented as one of the possible methods for data compression, the authors do not investigate the impact of the network topology and that of the routing scheme on the compression process.

Another interesting application for network monitoring exploiting CS is presented in [44], where the aim is to efficiently monitor communication metrics, such as loss or delay, over a set of end-to-end network paths by observing a subset of them. The topology is given a priori and the algorithm works in three steps: 1) compression, 2) non linear estimations and 3) suitable path selection. This last step in particular allows the selection of the best measurements for CS recovery, and therefore highly impacts the overall performance of the

algorithm.

In [45] and [46] an approach to distributed coding and compression in sensor networks based on CS is presented. The authors advocate the need to exploit the data both temporally and spatially. The projections of the signal measurements are performed at each source node, taking into account only the temporal correlation of the generated information. Thus, it is possible to design the best approximation of the collection of measurements for each node, since the projections can contain all the elements of this set. The spatial correlation is then exploited at the sink by means of suitable decoders through a joint sparsity model that well characterizes the different types of signals of interest.

A further related line of research is that of real and complex Network Coding [47, 48]. These papers highlight the analogies between Network Coding (NC) and Compressive Sensing from the viewpoint of distributed data processing and routing rules. An important difference between NC and CS is that CS works in real fields whereas NC exploits algebraic operations over Galois fields. This leads to practical issues, such as round-off errors that arise when dealing with real numbers, which are treated in [47].

Finally, it is also worth to mention the paper [49], which focuses on image recovery and compares classical CS recovery assuming random projections against an alternative method, where the projections are obtained through Principal Component Analysis (PCA). We anticipate here that also in our research we addressed both CS and PCA; nevertheless, the perspective of our study is very different from the one adopted in [49], as we used these two techniques in combination, by exploiting PCA to obtain good sparsification bases for the signal and CS to recover the signal given these bases.

## 2.3 Preliminary Studies

In the previous sections we have introduced Compressive Sensing and given insights about its promises for fully distributed compression in Wireless Sensor Networks. We have seen that, in theory, CS allows us to approximate the readings from a sensor field with excellent accuracy, while collecting only a small fraction of them at a data gathering point. However, the conditions under which CS performs well are not necessarily met in practice.

CS requires a suitable transformation that makes the signal sparse in its domain. Also, the transformation of the data given by the routing protocol and network topology and the sparse representation of the signal have to be incoherent (see Section 2.1.1), which is not straightforward to achieve in real networks.

In this section we present an overview of the preliminary studies carried out to address the data gathering problem in WSNs, where routing is used in conjunction with CS to transport random projections of the data. In these first studies, we considered the signals of interest mainly as "static pictures", according to the perspective presented in [40] (in Sections 2.4 and 2.5, instead, we will consider actually dynamic processes). We report analysis of both synthetic and real data sets and result comparisons against different techniques. In doing so, we present a number of popular transformations and we find that, with real data sets, none of them are able to sparsify the data while being at the same time incoherent with respect to the routing matrix. The obtained performance is thus not as good as expected, but such preliminary studies gave us the proper basis to look for the right approach: namely, a suitable method to build a transformation with good sparsification and incoherence properties for data gathering in static WSNs, as explained in Section 2.3.5.

### 2.3.1 Considered Signals and Transformations

In this section we discuss the signals that we considered for the performance evaluation in our preliminary studies.

First, we investigate synthetic signals that are sparse by construction under the DCT transformation. For these signals the degree of sparseness can be precisely controlled. As expected, when they are sufficiently sparse CS achieves substantial gains compared to plain routing schemes.

Successively, we select a number of static signals from real sensor networks measuring different physical phenomena. With such signals, we can much better characterize the per-

formance expected for actual WSN deployments. The problem with real signals, however, is to find a good transformation that sparsifies them in some domain. This issue is discussed at the end of the section.

**Synthetic signals.** Here, for the signal of interest we use a matrix $\mathbf{X}$ that we build starting from a sparse and discrete 2D signal $\mathbf{S}$ in the frequency (DCT) domain. In this context, we refer to the element $(i, j)$ of $\mathbf{X}$ or $\mathbf{S}$ as $x(i, j)$ or $s(i, j)$, respectively. According to the introduced formalism, $\mathbf{S}$ is obtained through the following steps:

1. Let $K$ be defined as $K = \sqrt{N}$, where $N$ is the number of values of the 2D signal. We build a preliminary signal $\mathbf{S}_1$ of size $K \times K$ having all frequencies (i.e., all entries in the matrix) with amplitude $s_1(i, j)$, where $s_1(i, j)$ is picked uniformly at random in the interval $[0.5, 1.5]$, $\forall\, i, j = 1, 2, \ldots, K$.

2. We define a frequency mask as a 2D function that is one for entries in position $(i, j)$ where $i + j \leq \mathrm{th}_{\mathrm{low}}$ or $i + j > \mathrm{th}_{\mathrm{high}}$ and zero otherwise. $\mathrm{th}_{\mathrm{low}}$ and $\mathrm{th}_{\mathrm{high}}$ are two thresholds in the value range $\{1, 2, \ldots, K\}$. This function is defined as

$$\texttt{triang}(i, j) \stackrel{def}{=} \begin{cases} 1 & \text{if } i + j \leq \mathrm{th}_{\mathrm{low}} \text{ or } i + j > \mathrm{th}_{\mathrm{high}} \\ 0 & \text{otherwise .} \end{cases} \tag{2.25}$$

3. We obtain a second signal $\mathbf{S}_2$ of size $K \times K$, whose entries $s_2(i, j)$ are calculated as

$$s_2(i, j) = s_1(i, j)\texttt{triang}(i, j) \ . \tag{2.26}$$

4. We finally obtain $\mathbf{S}$ as follows: if $s_2(i, j) = 0$ then $s(i, j) = \xi$ where $\xi \in [0, 0.01]$ is a constant. If instead $s_2(i, j) > 0$, $s(i, j) = \xi$ with probability $p_d$ and $s(i, j) = s_2(i, j)$ otherwise. The parameter $p_d$ represents the fraction of entries that are on average deleted from $\mathbf{S}_2$. The case $\xi > 0$ is accounted for to mimic non ideal signals, where the significant components lie within specific regions according to (2.25) and some *noise floor* is also present outside these regions. In this case, with CS we would like to only retrieve the significant values, while ignoring the noise.

Therefore, the signal $\mathbf{S}$ is obtained by first applying a frequency mask, which helps to assess the reconstruction performance for low-frequency, mid-frequency, and high-frequency signals. In addition, we delete some randomly picked frequencies according to a given

probability $p_d$. This is a simple method to control the characteristics of the signal in the DCT domain (i.e., the sparsity of the signal and its dominant frequency components) and allows to understand the effects of the signal structure on the performance of CS. For the results in Section 2.3.4 synthetic signals are mapped into matrices $\mathbf{X}$ of size $20 \times 20$, which is consistent with the network topology in Section 2.3.2 with $N = 400$ nodes.

**Real Signals.** We also used real signals from different environmental phenomena, considering what is likely to be of interest for a realistic wireless sensor network in terms of size of the network (i.e., number of spatial samples) and type of phenomenon to sense. For the sensor network, we considered the topology in Section 2.3.2 with $N = 400$ sensor nodes.

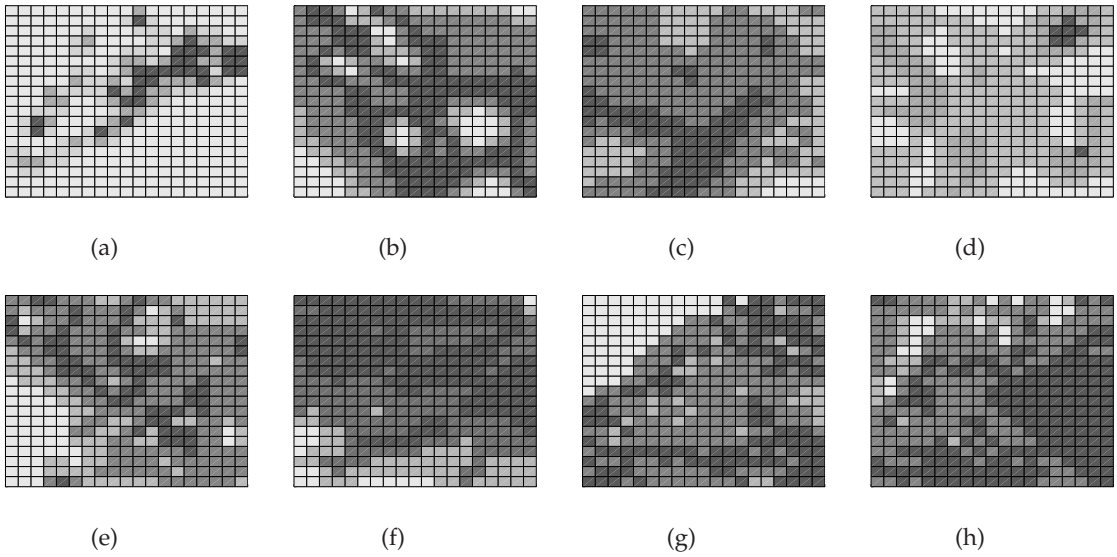



**Figure 2.5.** *Real signals: (a) Wi-Fi strength from MIT, (b) Wi-Fi strength from Stevens Institute of Technology, (c) Ambient temperature from EPFL SensorScope WSN, (d) Solar radiation from EPFL SensorScope WSN, (e) Rainfall in Texas, (f) Temperature of the ocean in California, (g) Level of pollution in Benelux and (h) in northern Italy.*

The following real signals were utilized:

S1. Two signals representing the Wi-Fi strength of the access points in the MIT campus (Cambridge, MA) [50] and in the Stevens Institute of Technology (Hoboken, NJ) [51].

S2. Two sets of measurements from the EPFL SensorScope WSN [52], representing ambient temperature and solar radiation.

S3. Two data readings, one from the Tropical Rainfall Measuring Mission [53] concerning

rain fall in Texas, and one on the temperature of the ocean off the coast of California [54].

S4. Two signals on the level of pollution in two European regions, namely, Benelux and Northern Italy [55].

These signals were quantized into five levels and rescaled in grids of $20 \times 20$ pixels. The assumption of measuring quantized signals was made as we think this is likely to be the case in actual WSN deployments, where the devices, due to communication, energy constraints or accuracy of the on-board sensor, can only sense or communicate the physical phenomena of interest according to a few discrete levels. In addition, for many signals of interest a quantized representation suffices to fully capture the needed information about the sensed phenomenon. The eight sample signals, quantized and rescaled as discussed above, are shown in Figure 2.5.

**Transformations.** By construction, for the above synthetic signals the DCT is the right sparsification method. These signals were in fact created sparse in the DCT domain. An effective utilization of CS for real signals requires a good sparsification approach. It is not trivial, however, to determine which approach is best for a given class of signals. Here, we consider four different transformations, which are commonly used in the image processing literature:

T1. *DCT:* this is the standard 2D discrete cosine transformation, see Appendix 2.A for further details.

T2. *Haar Wavelet:* the Haar Wavelet is recognized as the first known Wavelet and is a good Wavelet transformation for the sparsification of piece-wise constant signals as the ones in S1–S4, see [56].

T3. *Horz-diff:* this is a transformation that we propose here to exploit the spatial correlation of our signals. First, the 2D signal matrix $\mathbf{X}$ is written in vector form as follows:

$$\begin{aligned}\texttt{svec}(\mathbf{X}) = (&x(1,1), x(1,2), \ldots, x(1,k), x(2,K), x(2,K-1), \ldots \\ &\ldots, x(2,1), x(3,1), x(3,2), \ldots, x(3,K), x(4,K), x(4,K-1), \ldots \\ &\ldots, x(4,1), \ldots, x(K,K))^T . \end{aligned} \tag{2.27}$$

At this point we obtained the sparse vector $\mathbf{s}$ from $\texttt{svec}(\mathbf{X})$ by pair-wise subtraction of its elements.

T4. *HorzVer-diff:* according to this transformation the input signal $\mathbf{X}$ is processed by: 1) pair-wise subtraction of the elements along the columns of $\mathbf{X}$ and then 2) pair-wise subtraction of the elements of the resulting matrix, along its rows.

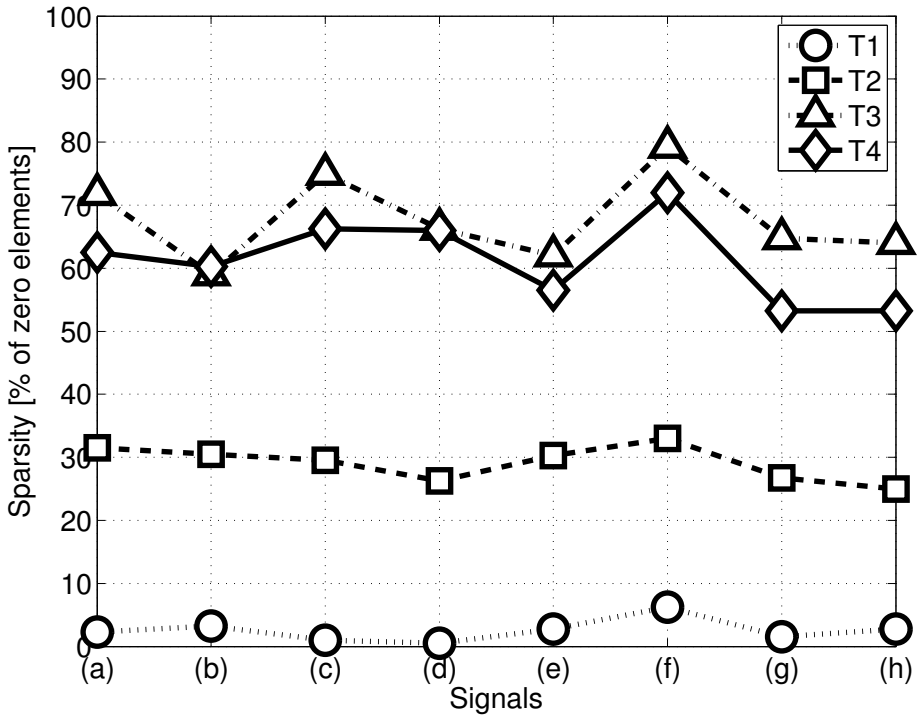



**Figure 2.6.** *Degree of sparsity for transformations T1–T4. The plot shows the percentage of zero elements of vector* $\mathbf{s}$ *after using transformations T1–T4.*

In Figure 2.6 we show the degree of sparseness achievable using the above transformations T1–T4 with the considered real signals (a)–(h). Notably, DCT (T1) and Haar Wavelet (T2) are not effective, whereas T3 and T4 perform best.

DCT and Wavelet transformations in this case have poor performance as, even though the sampled input signals $\mathbf{X}$ are quite large ($N = 400$ data points) for typical sensor deployments (where each node gathers a single data point), their size is still too small for T1 and T2 to perform satisfactorily. In this case, in fact, $N$ is related to the sampling rate of the transformations, which is also related to the bandwidth of the transformed signal. Due to this, a small $N$ implies that the components at high frequencies are likely to be non-negligible. T3 and T4 perform best since they exploit the characteristics of piece-wise constant signals, even if the sparsity obtained is not sufficient for CS to work properly. Since standard techniques as T1–T4 are not satisfactory, a more fundamental approach, i.e., via estimation of

the correlation of $\mathbf{X}$ and Karhunen-Loève expansion, has been envisioned at this point. We will discuss properly this line of research in Section 2.4.

### 2.3.2 Network Model

The concern of our study is about data gathering in 2D WSNs. Hence, for the rest of this section we consider sensor grids of $N$ nodes as follows. We consider $N$ nodes to be deployed in a square area with side length $L$. This area is split into a grid with $N$ square cells and we place each of the $N$ nodes uniformly within a given cell so that each cell contains exactly one node. For the transmission range $R$ of the nodes we adopt a unit disk model, i.e., nodes can only communicate with all other nodes placed at a distance less than or equal to $R$.[4] We use $R = \sqrt{5}L/\sqrt{N}$ as this guarantees that the structure is fully connected under any deployment of the nodes. A further node, the data gathering point or sink node, is placed in the center of the deployment area. We consider geographic routing to forward the data towards the sink, where each node considers as its next hop the node within range that provides the largest geographical advancement towards the sink. In Figure 2.7, we show an example topology; as per the above construction process, each cell has a node and the network is always connected. The tree in this figure is obtained through the above geographic routing approach, and is used by the data aggregation protocols to route data towards the sink.

According to this network scenario, the input signal is a square matrix $\mathbf{X}$ with $N$ elements, where element $x(i, j)$ is the value sampled by the sensor placed in cell $(i, j)$ of the sensor grid.

Despite its simplicity and the assumption that each cell contains a sensor node, this scenario captures the characteristic features (multi-hop routing and all to one transmission paradigm) of actual WSN deployments and allows to study the interplay between data gathering and compressive sensing. Actual WSN deployments will be considered in Sections 2.4 and 2.5.

### 2.3.3 Data Gathering Protocols

As pointed out in Section 2.2, there is a well studied line of research on the application of CS to data gathering in wireless networks. Previous studies however adapted the routing technique or the data transmission phase so as to take full advantage of CS. What we do here

[4]The unit disk graph model is used here for simplicity of explanation and topology representation. However, the presented methodology can be readily applied to more realistic propagation models, e.g., fading channels.

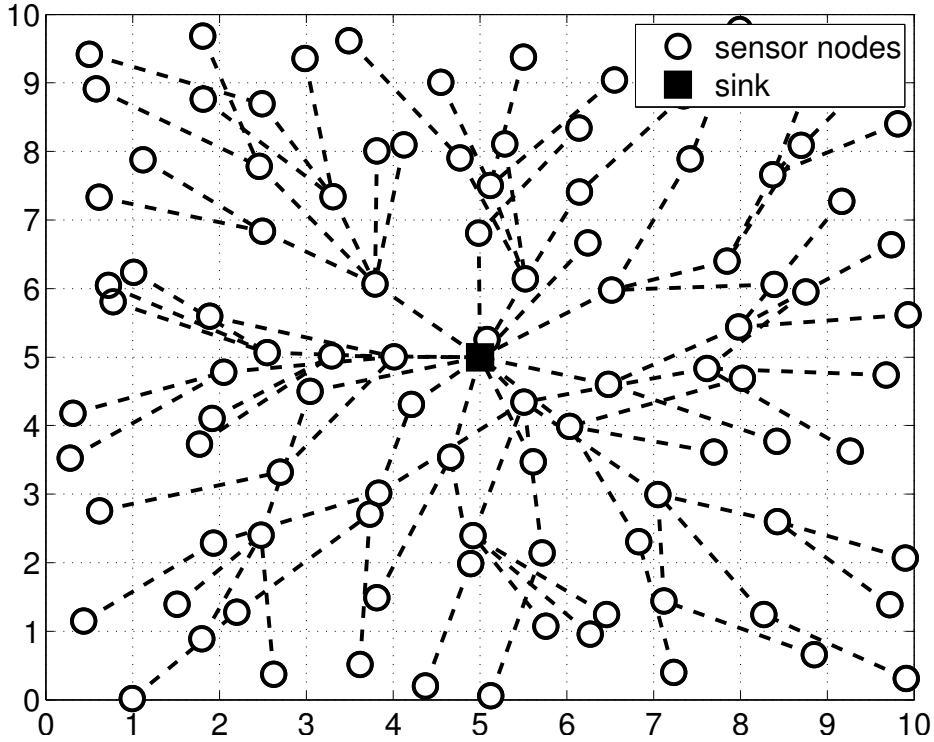



**Figure 2.7.** *Example of the considered multi-hop topology.*

is different as we pick a distributed WSN and consider the usual data gathering paradigm where sensors forward the packet(s) they receive along shortest paths towards the sink. This occurs in a completely unsynchronized and distributed manner, without knowledge about the correlation structure of the data and without knowing how it is processed at the sink through CS. Thus, our aim is to assess whether CS provides performance benefits with respect to standard schemes even in such distributed and unsynchronized network scenarios.

In what follows we present two schemes: the first is a standard geographical routing protocol, whereas the second is the same protocol in terms of routing, but it exploits CS for data recovery at the sink. We then characterize the structure of the $\Phi$ matrix (see Section 2.1) which is determined by the routing policy.

**Data gathering protocols.** To simplify the investigation and to pinpoint the fundamental performance trade-offs, in this first study we neglect channel access considerations (i.e., collisions, transmission times, etc.). Also, we assume a unit cost for each packet transmission and we ignore processing overhead at the nodes, as it is expected to be cheap compared to the cost of packet transmission.

P1. *Random sampling (RS):* this is the simplest protocol that we consider. In this case, each node becomes a source with probability $P_T = M/N$, which was varied in the simulations to obtain tradeoff curves for an increasing transmission overhead. On average, $M$ nodes transmit a packet containing their own sensor reading. Each packet is routed to the sink following the path that minimizes the number of transmissions (as defined by our geographical routing approach). Along this path, the packet is not processed but simply forwarded. The cost of delivering a single packet to the sink is given by the number of hops that connect the originating node to the data gathering point. The signal is reconstructed by interpolation of the collected values according to the method in [57].

P2. *Random sampling with CS (RS-CS):* this protocol is similar to RS. As above each node becomes a source with probability $P_T = M/N$. Again, each of these source nodes transmits a packet containing the reading of its own sensor. As this packet travels towards the sink, we combine the value contained therein with that of any other node that is encountered along the path. Specifically, let $x_i^m$ with $i = 1, 2, ..., \ell_m$ be the readings of the sensors along the path from node $m$ to the sink, where $x_1^m$ is the reading of the node itself and $\ell_m$ is the length of the path. Node $m$ sends a packet containing the value $y_1^m = \alpha_1^m x_1^m$ as well as the combination coefficient $\alpha_1^m$, where $\alpha_1^m$ is a value chosen uniformly at random either from $(0, 1]$ or from the set $\{-1, +1\}$.[5] The next node along the path will update the transmitted value and send out $y_2^m = y_1^m + \alpha_2^m x_2^m$ where $\alpha_2^m$ is again a random value. Also the coefficient $\alpha_2^m$ is included in the data packet along with $\alpha_1^m$. We proceed with these random combinations, where in general node $i + 1$ sends out

$$y_{i+1}^m = y_i^m + \alpha_{i+1}^m x_{i+1}^m \,, \tag{2.28}$$

until the packet finally reaches the sink. The sink extracts $y_{\ell_m}^m = \sum_{i=1}^{\ell_m} \alpha_i^m x_i^m$, together with coefficients that were used along the route. As explained below, these coefficients are the non-null elements of the $m$-th row of matrix $\mathbf{\Phi}$, referred to as $\overline{\boldsymbol{\varphi}}^m$. Note that some optimizations are possible. First, if we know in advance the network topology, we can assign combination coefficients at setup time to all nodes, rather than including them in the packets. We can further use the same pseudo-random number generator at the nodes and the sink and synchronize the seeds. However, all of this goes beyond

[5] The implications of the selection of the set to use are discussed in Section 2.3.4.

the scope of this study and we do not focus on how to optimize the control overhead of CS.

A few observations are in order. When we use CS at the sink, we receive packets carrying more valuable information than in the plain forwarding case. The received values are *linear random* combinations of the readings of several sensor nodes. For example, considering RS-CS, when the sink receives the $m$-th packet it can build a system of the form

$$\mathbf{y} \stackrel{def}{=} \begin{pmatrix} y^1_{\ell_1} \\ y^2_{\ell_2} \\ \vdots \\ y^m_{\ell_m} \end{pmatrix} = \begin{pmatrix} \overline{\varphi}^1 \\ \overline{\varphi}^2 \\ \vdots \\ \overline{\varphi}^m \end{pmatrix} \texttt{vec}(\mathbf{X}) = \mathbf{\Phi}\texttt{vec}(\mathbf{X}) \ , \tag{2.29}$$

where the $y^r_{\ell_r}$ with $r = 1, 2, \ldots, m$ are the combined values that were received by the sink in the packet that traversed the $r$th path, $\mathbf{X}$ is the input 2D signal, $\texttt{vec}(\mathbf{X})$ is defined as

$$\texttt{vec}(\mathbf{X}) \stackrel{\text{def}}{=} \begin{pmatrix} x(1,1) \\ \vdots \\ x(1,k) \\ x(2,1) \\ \vdots \\ x(2,k) \\ \vdots \\ x(k,1) \\ \vdots \\ x(k,k) \end{pmatrix} \tag{2.30}$$

and $\mathbf{\Phi}$ is an $m \times N$ matrix whose generic row $r$, $\overline{\varphi}^r$, contains the vector of coefficients $\alpha$ included in the packet. Note that, in general, some of the elements of $\overline{\varphi}^r$ might be equal to zero. Specifically, the node in cell $(i, j)$ of the 2D grid can only contribute to entry $(i-1)K+j$ of vector $\overline{\varphi}^r$, as we can figure out from the ordering shown in Equation (2.30). Thus, the combination coefficient in position $(i-1)K+j$ of $\overline{\varphi}^r$, with $i, j = 1, 2, \ldots, K$, is non-zero if and only if node $(i, j)$ was included in the path followed by the $r$-th packet and is set to zero otherwise.[6] Hence, matrix $\mathbf{\Phi}$ highly depends on the *network topology* and on the

[6]Given this, we see that setting an entire column of the matrix to zero, say column $c = (i-1)K+j$ for given $i$ and $j$, means that we completely ignore the contribution of the node placed in cell $(i, j)$. This happens when none of the $m$ received packets passes through this node while being routed to the sink.

*selected routing rules* as each of its rows will have non-zero elements only in those positions representing nodes that were included in the path followed by the corresponding packet.

Note that (2.29) is a system of linear equations that is in general ill-posed (as $m \leq M$ and $M$ is expected to be smaller than $N$). At the sink, we know vector $\mathbf{y}$ and matrix $\mathbf{\Phi}$ and we need to find the 2D input signal $\mathbf{X}$. We can now use the derivations in the Appendix 2.A and rewrite $\mathbf{y} = \mathbf{A}\texttt{vec}(\mathbf{S})$ which is solved for $\texttt{vec}(\mathbf{S})$ using the standard Compressive Sensing tools for the 1D case explained[7] in Section 2.1, thus finding the sparsest $\texttt{vec}(\mathbf{S})$ that verifies the system, referred to here as $\widehat{\mathbf{S}}$. $\widehat{\mathbf{S}}$ is finally used to reconstruct $\mathbf{X}$, i.e., $\widehat{\mathbf{X}} = \mathbf{\Psi}\widehat{\mathbf{S}}\mathbf{\Psi}^T$ (see also Equation (2.71) in the Appendix 2.A).

**Characterization of the routing matrix $\mathbf{\Phi}$.** According to our network model, the nodes that transmit their packet to the sink are chosen at random. As said above, every row $\overline{\boldsymbol{\varphi}}^j$ of $\mathbf{\Phi}$ represents a path from a given sensor to the sink and each forwarding node in this path contributes with a non zero coefficient. We characterize the sparsity $\nu_j$ of $\overline{\boldsymbol{\varphi}}^j$ counting the number of elements in this row that differ from zero: $\nu_j = \sum_{i=1}^{N} \mathbb{1}\{\overline{\varphi}_i^j \neq 0\}$, where $\overline{\varphi}_i^j$ is the $i$-th entry of vector $\overline{\boldsymbol{\varphi}}^j$ and $\mathbb{1}\{E\}$ is the indicator function, which is 1 when event $E$ is true and zero otherwise. $\nu_j$ is the cost, in terms of number of transmissions, for sending the $j$-th packet to the sink. With the network scenario in Section 2.3.2 it is easy to see that, for any source node in the network, the number of transmissions required for its packet to reach the sink is $O(\sqrt{N})$. Hence, the total cost for the transmission of $M$ packets is $O(M\sqrt{N})$. As an example, for a network with $N = 400$ nodes the cost of delivering a packet to the sink is $\simeq 4.5$ transmissions, which is close to $\sqrt{N}/4$. The sparsity of $\overline{\boldsymbol{\varphi}}^j$ directly translates into the sparsity of $\mathbf{\Phi}$ that, in turn, affects the *coherence* between the matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ (see Section 2.1.1). In the literature, the concept of coherence (or its dual, called *incoherence*) between these two matrices is directly related to the effectiveness of the CS recovery phase and is well defined when they are orthonormal. Specifically, the routing matrix $\mathbf{\Phi}$ and $\mathbf{\Psi}$ must be incoherent for CS to work properly [15].

In our settings, however, $\mathbf{\Phi}$ is built on the fly according to the routing topology, whereas $\mathbf{\Psi}$ is obtained according to any of the transformations T1–T4 that we discussed in Section 2.3.1. In the literature the concept of coherence is not defined for non-orthogonal matrices. However, according to the rationale in [15, 45] a quantity that is strictly related to the

[7] To be in complete accordance with the formalism introduced in Section 2.1, we just have to replace $\texttt{vec}(\mathbf{X})$ and $\texttt{vec}(\mathbf{S})$ with $\mathbf{x}$ and $\mathbf{s}$, respectively

incoherence can be computed as follows. Roughly speaking, incoherence between two matrices means that none of the elements of one matrix has a sparse representation in terms of the columns of the other matrix (if used as a basis). Put differently, two matrices are highly coherent when each element of the first can be represented linearly combining a small number of columns of the second. Hence, to characterize the incoherence we first project each row of $\mathbf{\Phi}$ into the space generated by the columns of $\mathbf{\Psi}$. After this, we take the sparsest projections obtained in this space as an indication of the incoherence. Formally, we have:

$$\boldsymbol{\zeta}^j = (\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\mathbf{\Psi}^T\left(\overline{\boldsymbol{\varphi}}^j\right)^T \ , \tag{2.31}$$

where $\overline{\boldsymbol{\varphi}}^j$ is the $j$-th row of $\mathbf{\Phi}$ and $\boldsymbol{\zeta}^j$ is the (column) vector of coefficients corresponding to its projection on the space generated by the columns of $\mathbf{\Psi}$. A measure of the incoherence is then obtained as

$$I(\mathbf{\Phi},\mathbf{\Psi}) = \min_{j=1,\ldots,N}\left[\sum_{i=1}^{N}\mathbb{1}\{\zeta_i^j \neq 0\}\right] \in \left[1, N\right] \ , \tag{2.32}$$

where $\zeta_i^j$ is the $i$-th entry of vector $\boldsymbol{\zeta}^j$.

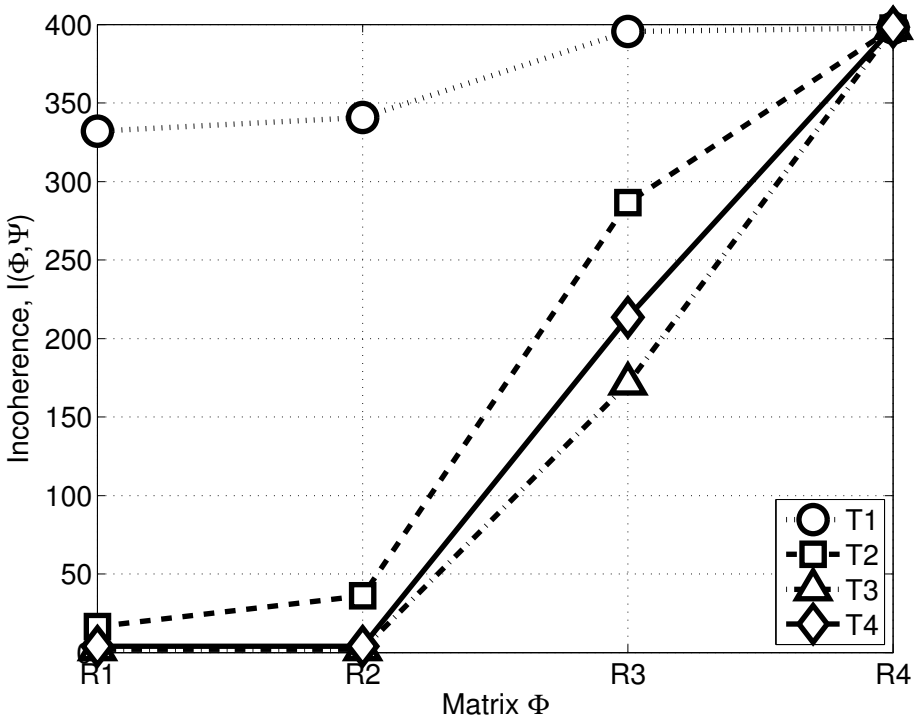



**Figure 2.8.** *Incoherence $I(\mathbf{\Phi},\mathbf{\Psi})$ between the routing matrix $\mathbf{\Phi}$, cases R1–R4, and the transformation matrix $\mathbf{\Psi}$, transformations T1–T4. The maximum value for $I(\mathbf{\Phi},\mathbf{\Psi})$ equals the number of nodes in the network, $N = 400$.*

In Figure 2.8 we show the incoherence, obtained from (2.32), for the four transformation methods T1–T4 and for the following matrices $\mathbf{\Phi}$:

R1) $\mathbf{\Phi}$ is built according to the CS routing protocol that we explained above, picking random coefficients in $\{-1, +1\}$;

R2) $\mathbf{\Phi}$ is built as in case R1, picking random coefficients in $(0, 1]$;

R3) $\mathbf{\Phi}$ has all coefficients randomly picked in $\{-1, +1\}$;

R4) $\mathbf{\Phi}$ has coefficients uniformly and randomly picked in $(0, 1]$.

As can be deduced from the results of [41], cases R3 and R4 are near optimal in terms of projections of the measurements and can be built through a pre-distribution of the data (that in a multi-hop WSN is in general demanding in terms of number of transmissions).

From this plot we see that the DCT transformation (T1) has a high incoherence with respect to all of the considered routing matrices. The remaining transformations T2–T4 all perform similarly and give satisfactory performance only for cases R3 and R4, whereas for random projections obtained through the actual routing scheme they are highly coherent to $\mathbf{\Phi}$. This has strong negative implications on the CS recovery performance and will be discussed in the following section.

### 2.3.4 Results

In this section we discuss the results of our preliminary study on CS in WSN. These results have been obtained by simulating the RS and RS-CS data gathering schemes for synthetic and real signals. The metric of interest is the reconstruction quality at the sink, which is defined as follows. Given a 2D input signal $\mathbf{X}$, a matrix $\mathbf{\Phi}$ and a vector $\mathbf{y}$ (containing the received values that are linear combinations of the sensor readings in the network) we have that $\mathbf{y} = \mathbf{\Phi}\mathtt{vec}(\mathbf{X})$. This system, that in general is ill-posed (as $M \leq N$), is solved for $\mathtt{vec}(\mathbf{X}) = \mathtt{vec}(\mathbf{\Psi S \Psi}^T)$ either through norm one [15] or smoothed zero norm [32] minimization. These methods efficiently find the sparsest $\mathbf{S}$, referred to as $\widehat{\mathbf{S}}$, that verifies the previous system.[8] If $\widehat{\mathbf{X}} = \mathbf{\Psi}\widehat{\mathbf{S}}\mathbf{\Psi}^T$ is the solution found for this system and $\mathbf{X}$ is the true input signal, the *reconstruction error* is defined as

$$\varepsilon = \frac{\|\mathtt{vec}(\mathbf{X}) - \mathtt{vec}(\widehat{\mathbf{X}})\|_2}{\|\mathtt{vec}(\mathbf{X})\|_2} \ . \tag{2.33}$$

[8] We found that these two methods are nearly equivalent in terms of quality of the solution, although the zero norm is simplest and faster. This might be important for practical implementations.

**Results for Synthetic signals**

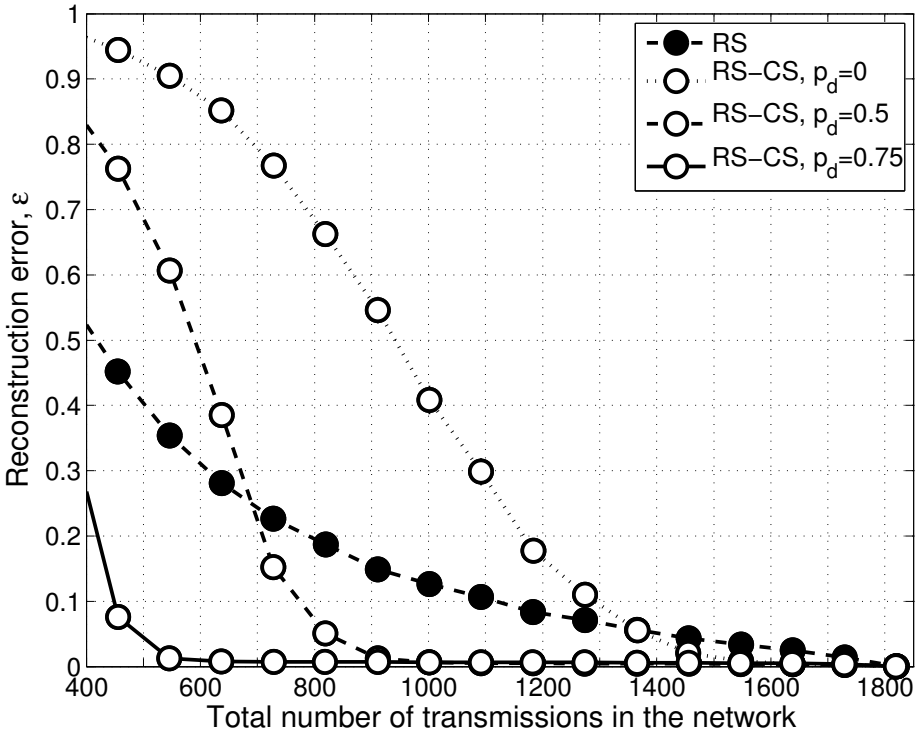



**Figure 2.9.** *Reconstruction quality $\varepsilon$ as a function of the total number of packets transmitted in the network: comparison between RS and RS-CS for synthetic signals and different values of $p_d$.*

In Figure 2.9 we show the reconstruction error $\varepsilon$ as a function of the total number of packets sent in the network for RS and RS-CS. For this plot we considered a low-pass signal with $\mathrm{th}_{\mathrm{low}} = \sqrt{N}/2 + 1$ and $\mathrm{th}_{\mathrm{high}} = \sqrt{N}$, with $N = 400$. Also, we considered three values of $p_d \in \{0, 0.5, 0.75\}$ so as to vary the sparseness of the signal. As a first observation, random sampling performs nicely for low-pass signals. Nevertheless, a perfect reconstruction of the sensed signal at the sink requires the transmission of a large number of packets (up to $1800$). When the signal is sufficiently sparse ($p_d \geq 0.5$) CS outperforms standard data gathering schemes, requiring less than half the packet transmissions (about $900$) to achieve the same recovery performance. We noticed that values of $\varepsilon$ larger than $0.3$ always led to very inaccurate reconstructions of the original signal. Figure 2.9 was obtained using $L_1$ minimization and combination coefficients in the set $\{-1, +1\}$. However, we obtained similar performance using smoothed $L_0$ norm and/or coefficients in the set $(0, 1]$. Note that using the set $\{-1, +1\}$ allows for reduced overhead as, in practical implementations, a single bit suffices to transmit each coefficient.

For high-pass signals the performance of CS is unvaried for the same degree of sparseness. This is expected as CS recovery operates in the frequency domain and is only affected by the number of non-zero frequency components and not by their position. Clearly, RS with the considered interpolation technique is not appropriate for high-pass signals, in which case it shows poor recovery performance.

As a consequence, CS-RS shows good recovery performance for synthetic signals as, by construction, the DCT transformation effectively sparsifies the signal and this transformation is incoherent with respect to the routing matrix $\mathbf{\Phi}$ (see Figure 2.8).

Further results on CS applied to synthetic signals can be found in both Appendices 2.B and 2.C. In the Appendix 2.B, we present performance evaluations of CS varying the sparsity of the observed signal; in the Appendix 2.C, instead, we evaluate CS-RS against different protocols than those presented in this section.

**Results for Real Signals**

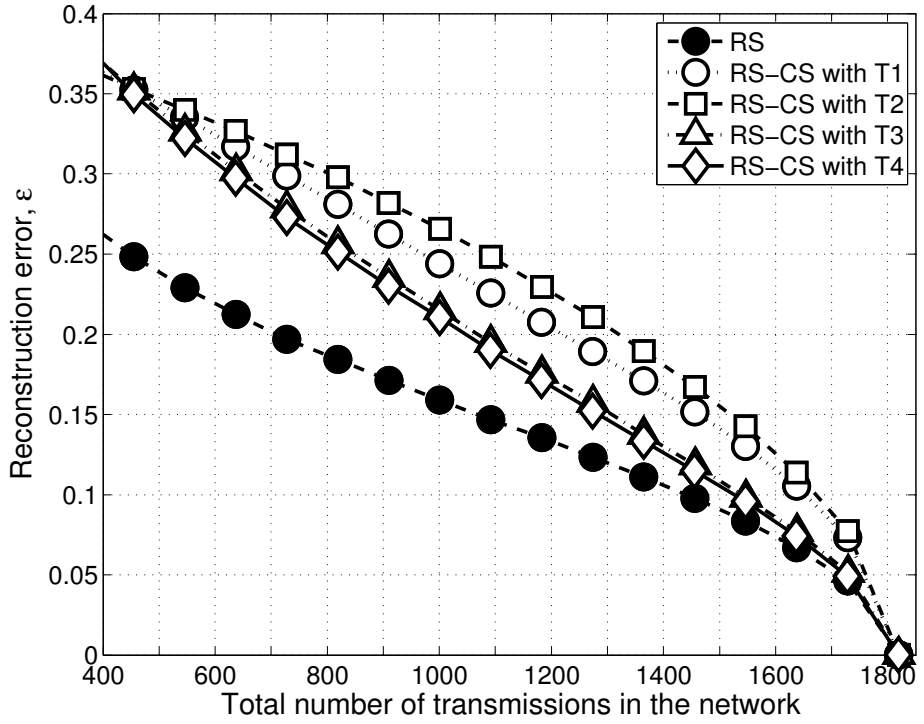



**Figure 2.10.** *Reconstruction error $\varepsilon$ vs total number of packets transmitted in the network: comparison between RS and RS-CS (for transformations T1–T4) for the real signals in Section 2.3.1.*

In Figure 2.10 we show the reconstruction error $\varepsilon$ as a function of the total number of packets sent in the network for RS and RS-CS. The sensed signals belong to the data sets

presented in Section 2.3.1. In this case, differently from the case of synthetic signals, RS-CS does not outperform RS, even though the performance of the two methods is very close. The reason for this is twofold. First, the considered transformations T1–T4 sparsify the real signals only up to 70% (see Section 2.3.1). This is mainly due to the characteristics of the signals and to the small size of the sample set. Second, the transformations with the best performance in terms of sparsification have a high coherence with respect to the routing matrix of CS-RS. Hence, while the sparsification performance may suffice, matrix $\mathbf{\Phi}$ (routing) does not have the required properties in terms of coherence for CS to perform satisfactorily.

In fact, for good recovery performance CS needs a good transformation in terms of sparsification. Also, transformation and routing matrices must be incoherent. From Figures 2.6, 2.8 and 2.10 we see that transformations T3 and T4 are the most suitable to sparsify the considered real signals and this allows them to perform better than T1 and T2 (even though they perform poorly in terms of incoherence, see Section 2.3.3). In addition, although T2 can sparsify real signals better than T1 (Figure 2.6), the latter performs better than T2 in terms of transmission cost *vs* error reconstruction (Figure 2.10), since it has better incoherence properties $I(\mathbf{\Phi}, \mathbf{\Psi})$ (Figure 2.8).

Finally, in Figure 2.11 we accounted for a pre-distribution phase of the data so that matrix $\mathbf{\Phi}$ is as close as possible to that of case R4 of Section 2.3.3 (we verified that case R3 gives similar performance). In this case, CS-RS outperforms RS as T1 and T2 provide a sparse representation of the signal and the routing matrix is sufficiently incoherent with respect to these transformations. However, this pre-distribution phase (which is similar to that proposed in [41]) has a high transmission cost, which is ignored in Figure 2.11.

### 2.3.5 Discussion on the Preliminary Studies

In these preliminary studies on Compressive Sensing applied to WSNs, we tested the behavior of this technique when used jointly with a routing scheme for recovering two types of signals: synthetic ones and real sensor data. We found out that for synthetic signals the reconstruction at the sink node is enhanced when applying CS, whereas the application of CS for real sensor data is not straightforward. Thus, the research on this subject has moved forward to further investigate which signal representation and routing allows CS to outperform random sampling in realistic WSN deployments. To this end, we jointly investigated

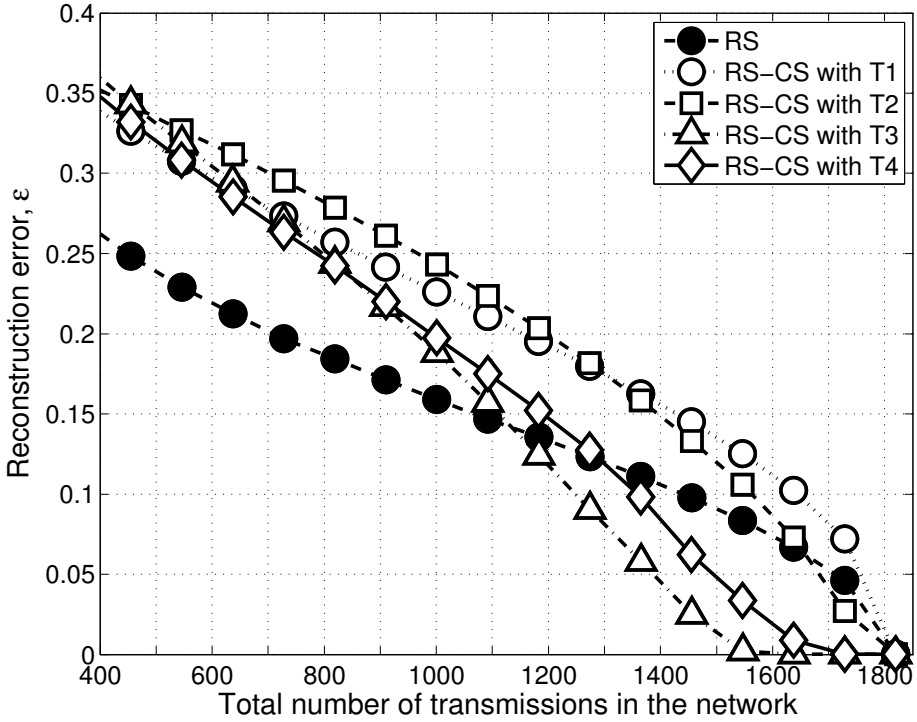



**Figure 2.11.** *Reconstruction error $\varepsilon$ vs total number of packets transmitted in the network: comparison between RS and RS-CS (for transformations T1–T4) when a pre-distribution of the data is allowed so that the routing matrix $\mathbf{\Phi}$ approaches that of case R4 of Section 2.3.3.*

the design of the two matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$, since the sparsity requirements and the incoherence between routing and signal representation have to be met.

As a result, we found that a good approach for applying CS in WSNs is to use CS in conjunction with a well-known statistical technique, called Principal Component Analysis (PCA). In detail, according to this approach, we propose to choose:

- $\mathbf{\Phi}$ as an $L \times N$ matrix with zero-elements but a single one in each row and at most a single one in each column;
- $\mathbf{\Psi}$ as an $N \times N$ orthonormal matrix whose columns are the unitary eigenvectors of the correlation matrix of the signal of interest $\mathbf{X}$, placed according to the decreasing order of the corresponding eigenvalues.

We note here that the above selection of $\mathbf{\Phi}$ has two advantages: 1) the matrix is orthonormal as generally required by CS, see e.g., [16] and 2) this type of routing matrix can be easily obtained through realistic routing schemes and does not require additional transmission overhead, as for implementing a data pre-distribution phase as envisioned in Section 2.3.4.

Furthermore, the joint use of $\mathbf{\Phi}$ and $\mathbf{\Psi}$ as proposed above, leads to good performance in terms of coherence [16, 28]. Roughly speaking, this is due to: 1) the orthonormality of both matrices and 2) their different structure (i.e., $\mathbf{\Phi}$ is very sparse, whilst $\mathbf{\Psi}$ is not).

In the following Sections 2.4 and 2.5 we describe in detail our proposed method for the application of CS to WSN data monitoring.

## 2.4 Signal Model and Real Signal Analysis

In the following we present and discuss our solution for exploiting CS in Wireless Sensor Network. As partially anticipated in Section 2.3.5, this approach jointly exploits Compressive Sensing and Principal Component Analysis to reconstruct real world (possibly non-stationary) signals through the collection of a small number of samples at a data gathering point. Our method is hereby characterized according to the Bayesian theory, that provides a general framework for data modeling [58,59]. As matter of fact, the Bayesian framework has been addressed in the recent literature to develop efficient and auto-tunable algorithms for CS [60]. However, previous work dealing with CS from a Bayesian perspective has mainly been focused on the theoretical derivation of CS and its usefulness in the image processing field. With the present study, instead, we provide empirical evidence of the effectiveness of CS in actual WSN monitoring scenarios.

In detail, the next sections are structured as follows. In Section 2.4.1, we describe the mathematical tools needed for implementing our data recovery framework. In doing so, we first briefly review the PCA theory; then we recall the CS mathematics of Section 2.1.1, adapting the formalism introduced therein for static signals to the case of signals that can vary over time; finally, we explain how to jointly exploit CS and PCA within our monitoring framework. This framework is properly presented in Section 2.4.2, along with a description of the signal model upon which is based. The considered real signals, gathered from actual WSN deployments, are described in Section 2.4.3, whilst, as original contribution, in Section 2.4.4 we infer the statistical distribution of the principal components of these real world signals. As concluding remark of the section, we will see how the presented study allows us to legitimate the use of CS in real world WSN. In particular, in Section 2.4.5 we shown that, according to the framework of Bayesian estimation, the CS recovery mechanism is equivalent to optimal maximum *a posteriori* (MAP) recovery. Moreover, we introduce here a simple example of protocol to give an insight into the advantages achievable by exploiting CS for signal monitoring. In Section 2.5, instead, is presented the actual signal monitoring framework based on CS that we propose as practical outcome of our research.

### 2.4.1 Mathematical tools

In this section we first review basic tools from PCA and CS and we subsequently illustrate a framework which jointly exploits these two techniques.

**Principal Component Analysis**

The Karhunen-Loève expansion is the theoretical basis for PCA. It is a method to represent through the best $M$-term approximation a generic $N$-dimensional signal, where $N > M$, given that we have full knowledge of its correlation structure. In practical cases, i.e., when the correlation structure of the signals is not known a priori, the Karhunen-Loève expansion can be approximated thanks to PCA [61], which relies on the on-line estimation of the signal correlation matrix. We assume to collect measurements according to a fixed sampling rate at discrete times $k = 1, 2, \ldots, K$. In detail, let $\mathbf{x}^{(k)} \in \mathbb{R}^N$ be the vector of measurements, at a given time $k$, from a WSN with $N$ nodes. $\mathbf{x}^{(k)}$ can be viewed as a single sample of a stationary vector process $\mathbf{x}$. The sample mean vector $\overline{\mathbf{x}}$ and the sample covariance matrix $\widehat{\mathbf{\Sigma}}$ of $\mathbf{x}^{(k)}$ are defined as:

$$\overline{\mathbf{x}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}^{(k)} \ , \ \widehat{\mathbf{\Sigma}} = \frac{1}{K}\sum_{k=1}^{K}(\mathbf{x}^{(k)} - \overline{\mathbf{x}})(\mathbf{x}^{(k)} - \overline{\mathbf{x}})^T \ . \tag{2.34}$$

Given the above equations, let us consider the orthonormal matrix $\mathbf{U}$ whose columns are the unitary eigenvectors of $\widehat{\mathbf{\Sigma}}$, placed according to the decreasing order of the corresponding eigenvalues. It is now possible to project a given measurement $\mathbf{x}^{(k)}$ onto the vector space spanned by the columns of $\mathbf{U}$. Therefore, let us define $\mathbf{s}^{(k)} \stackrel{def}{=} \mathbf{U}^T(\mathbf{x}^{(k)} - \overline{\mathbf{x}})$. If the instances $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(K)}$ of the process $\mathbf{x}$ are temporally correlated, then only a fraction of the elements of $\mathbf{s}^{(k)}$ can be sufficient to collect the overall energy of $\mathbf{x}^{(k)} - \overline{\mathbf{x}}$. In other words, each sample $\mathbf{x}^{(k)}$ can be very well approximated in an $M$-dimensional space by just accounting for $M < N$ coefficients. According to the previous arguments we can write each sample $\mathbf{x}^{(k)}$ as:

$$\mathbf{x}^{(k)} = \overline{\mathbf{x}} + \mathbf{U}\mathbf{s}^{(k)} \ , \tag{2.35}$$

where the $N$-dimensional vector $\mathbf{s}^{(k)}$ can be seen as an $M$-sparse vector, namely, a vector with at most $M < N$ non-zero entries. Note that the set $\{\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \cdots, \mathbf{s}^{(K)}\}$ can also be viewed as a set of samples of a random vector process $\mathbf{s}$. In summary, thanks to PCA, each original point $\mathbf{x}^{(k)} \in \mathbb{R}^N$ can be transformed into a point $\mathbf{s}^{(k)}$, that can be considered $M$-sparse. The actual value of $M$, and therefore the sparseness of $\mathbf{s}$, depends on the actual level of correlation among the collected samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(K)}$.

**Compressive Sensing**

As above, we consider signals representable through one dimensional vectors $\mathbf{x}^{(k)} \in \mathbb{R}^N$, containing the sensor readings at time $k$ of a WSN with $N$ nodes. Referring to Section 2.1.1, therefore, we just have to rewrite Equation (2.1) as

$$\mathbf{x}^{(k)} = \mathbf{\Psi}\mathbf{s}^{(k)} \ , \tag{2.36}$$

and Equation (2.2) as

$$\mathbf{y}^{(k)} = \mathbf{\Phi}\mathbf{x}^{(k)} \ . \tag{2.37}$$

Here, $\mathbf{\Phi}$ is still referred to as *routing matrix* as in Section 2.3 because it captures the way in which our sensor data is gathered and transmitted to the sink. However, as anticipated in Section 2.3.5, for the remainder of this thesis (if not otherwise specified) $\mathbf{\Phi}$ will be considered as an $L \times N$ matrix with zero-elements but a single one in each row and at most a single one in each column (i.e., $\mathbf{y}^{(k)}$ is a sampled version of $\mathbf{x}^{(k)}$). We recall here the two main advantages of this selection: from one hand, the matrix $\mathbf{\Phi}$ results orthonormal as generally required by CS, see e.g., [16]; form the other, this type of routing matrix can be easily obtained through realistic routing schemes.

Finally, using (2.36) and (2.37) we can rewrite Equation (2.3) as

$$\mathbf{y}^{(k)} = \mathbf{\Phi}\mathbf{x}^{(k)} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{s}^{(k)} \stackrel{def}{=} \mathbf{A}\mathbf{s}^{(k)} \ . \tag{2.38}$$

**Joint CS and PCA**

Here we propose a technique that jointly exploits PCA and CS to reconstruct a signal $\mathbf{x}^{(k)}$ at each time $k$. Assume that the signal is correlated both in time and in space, but that in general it is non-stationary. This means that the statistics that we have to use in our solution (i.e., in Equation (2.34), the sample mean $\overline{\mathbf{x}}$ and the covariance matrix $\widehat{\mathbf{\Sigma}}$) must be learned at runtime and might not be valid throughout the entire time frame in which we want to reconstruct the signal. We should also make the following assumption, that will be justified in the next sections:

1. at each time $k$ we have perfect knowledge of the previous $K$ process samples, namely we perfectly know the set $\mathcal{X}_K^{(k)} = \left\{\mathbf{x}^{(k-1)}, \mathbf{x}^{(k-2)}, \cdots, \mathbf{x}^{(k-K)}\right\}$, referred to in what follows as training set[9];

2. there is a strong temporal correlation between $\mathbf{x}^{(k)}$ and the set $\mathcal{X}_K^{(k)}$ that will be explicated in the next section via a Bayesian network. The size $K$ of the training set is

[9] In Section 2.5 we present a practical scheme that does not need this assumption in order to work.

chosen according to the temporal correlation of the observed phenomena to validate this assumption.

Using PCA, from Equation (2.35) at each time $k$ we can map our signal $\mathbf{x}^{(k)}$ into a sparse vector $\mathbf{s}^{(k)}$. The matrix $\mathbf{U}$ and the average $\overline{\mathbf{x}}$ can be thought of as computed iteratively from the set $\mathcal{X}_K^{(k)}$, at each time sample $k$. Accordingly, at time $k$ we indicate matrix $\mathbf{U}$ as $\mathbf{U}^{(k)}$ and we refer to the temporal mean and covariance of $\mathcal{X}_K^{(k)}$ as $\overline{\mathbf{x}}^{(k)}$ and $\widehat{\mathbf{\Sigma}}^{(k)}$, respectively[10]. Hence, we can write:

$$\mathbf{x}^{(k)} - \overline{\mathbf{x}}^{(k)} = \mathbf{U}^{(k)}\mathbf{s}^{(k)} \ . \tag{2.39}$$

Now, using Equations (2.37) and (2.39), we can write:

$$\mathbf{y}^{(k)} - \mathbf{\Phi}^{(k)}\overline{\mathbf{x}}^{(k)} = \mathbf{\Phi}^{(k)}(\mathbf{x}^{(k)} - \overline{\mathbf{x}}^{(k)}) = \mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)} \ , \tag{2.40}$$

where with the symbol $\mathbf{\Phi}^{(k)}$ we make explicit that also the routing matrix $\mathbf{\Phi}$ can change over time. The form of Equation (2.40) is similar to that of (2.38) with $\mathbf{A} = \mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}$. The original signal $\mathbf{x}^{(k)}$ is approximated as follows: 1) finding a good estimate[11] of $\mathbf{s}^{(k)}$, namely $\widehat{\mathbf{s}}^{(k)}$, using the techniques in [16] or [32] and 2) applying the following calculation:

$$\widehat{\mathbf{x}}^{(k)} = \overline{\mathbf{x}}^{(k)} + \mathbf{U}^{(k)}\widehat{\mathbf{s}}^{(k)} \ . \tag{2.41}$$

### 2.4.2 Monitoring Framework and Sparse Signal Models

In this section we describe a model to represent a broad range of environmental signals that can be gathered from a Wireless Sensor Network. The aim is to analyze the stochastic properties of these signals, in order to select the most appropriate sampling, compression and recovery techniques to minimize the number of transmitting nodes while keeping a certain level of reconstruction accuracy, as detailed in Section 2.4.5.

We have chosen to represent the variables involved with a Bayesian Network (BN) [62], i.e., a Directed Acyclic Graph (DAG) where nodes represent random variables and arrows represent conditional dependencies among them. From the DAG it is always possible to determine the conditional independence between two variables, applying a set of rules known as *d-separation* rules, e.g., see [63] for a detailed description about BNs properties. In this section, we propose two graphical models which illustrate the perspective we adopted in Section 2.4.4 and Section 2.4.5, respectively:

[10] See Equation 2.34.

[11] Here we refer to a good estimate of $\mathbf{s}^{(k)}$ as $\widehat{\mathbf{s}}^{(k)}$ such that $\|\mathbf{s}^{(k)} - \widehat{\mathbf{s}}^{(k)}\|_2 \leq \epsilon$. Note that by keeping $\epsilon$ arbitrarily small, Assumption 1 above is very accurate.

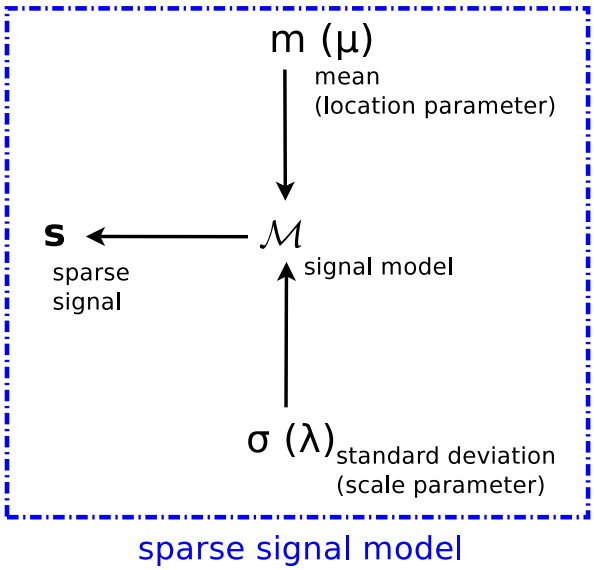



**Figure 2.12.** *Bayesian network used to model the probability distribution of the innovation signal* $\mathbf{s}$.

1. Figure 2.12 represents a stochastic model for the signal $\mathbf{s}$;
2. Figure 2.13 is a BN which links together all the variables involved in our analysis, highlighting those required to define the monitoring framework.

In detail, with Figure 2.12 we introduce a Bayesian model to describe the statistical properties of the elements of $\mathbf{s}^{(k)}$. Given the realizations of the signal $\mathbf{s}^{(k)}$ at time $k = 1, \ldots, K$, we use a Bayesian estimation method, described in Section 2.4.4, to infer a suitable model $\mathcal{M}$ along with the best-fitting values of its parameters. In particular, for a Gaussian model the parameters to infer are the mean value $m$ of each component and the standard deviation $\sigma$, whereas for a Laplacian model are the location parameter $\mu$ and the scale parameter $\lambda$, respectively. This modeling approach is exploited in Section 2.4.4 to determine which stochastic model, chosen among a set of plausible ones, better describes the signal $\mathbf{s}^{(k)}$.

Figure 2.13, instead, depicts the whole considered framework that involves the following variables for each time sample $k$: the training set $\mathcal{X}^{(k)}$, the WSN signal $\mathbf{x}^{(k)}$, its compressed version $\mathbf{y}^{(k)}$, obtained sampling $\mathbf{x}^{(k)}$ according to matrix $\mathbf{\Phi}^{(k)}$ as in Equation (2.37), the invertible matrix $\mathbf{\Psi}^{(k)}$, obtained through PCA, and the sparse representation $\mathbf{s}^{(k)}$, introduced in Equation (2.35). From the results presented in Section 2.4.4, it turns out that $\mathbf{s}^{(k)}$ is well approximated by a Laplacian distribution. Analyzing the DAG in Figure 2.13, based on the *d-separation* rules, we can make the following observations:

- **data gathering**: the WSN signal $\mathbf{x}^{(k)}$ is independent of the stochastic sampling matrix

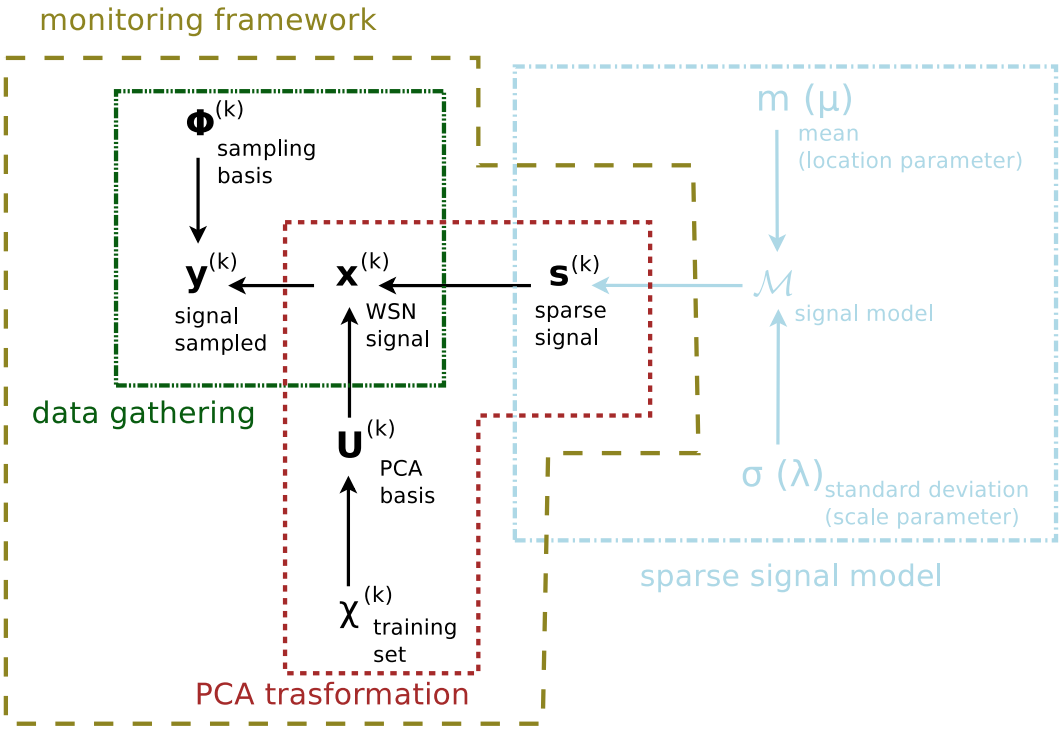



**Figure 2.13.** *Bayesian network used to model the considered real signals. In the scheme we highlight the monitoring framework at each time sample $k$.*

$\mathbf{\Phi}^{(k)}$, whose nature is better described in Section 2.5, but the observation of $\mathbf{y}^{(k)}$ reveals a link between these two variables;

- **PCA transformation**: this is the core of our model, that describes how the system learns the statistics of the signal of interest $\mathbf{x}^{(k)}$. According to the dynamic system framework, $\mathbf{\Psi}^{(k)}$ can be seen as the state of a system, since it summarizes at each instant $k$ all the past history of the system, represented by the set $\mathcal{X}^{(k)}$. The system input is the signal $\mathbf{s}^{(k)}$, that can be seen as a Laplacian or Gaussian innovation process. This type of priors on the signal induces estimators that use, respectively, the $\ell_1$ and $\ell_2$-norm of the signal as regularization terms. Hence, such priors are often used in the literature in view of the connection with powerful shrinkage methods such as ridge regression and LASSO, as well as for the many important features characterizing them, e.g., see Section 3.4 in [64] for a thorough discussion. Note also that the observation of the WSN signal $\mathbf{x}^{(k)}$ has a twofold effect: the former is the creation of a deterministic dependence between the PCA basis $\mathbf{\Psi}^{(k)}$ and the sparse signal $\mathbf{s}^{(k)}$, that otherwise are independent; the latter is the separation of $\mathbf{\Psi}^{(k)}$ and $\mathbf{s}^{(k)}$ from the data gathering variables, i.e., they become independent of $\mathbf{y}^{(k)}$ and $\mathbf{\Phi}^{(k)}$;

- **sparse signal model**: we observe that the priors assigned to the variable $\mathcal{M}$ and to the corresponding parameters $\mu$ (resp. $m$) and $\lambda$ (resp. $\sigma$) are non informative, except for the non-negativity of the variance. Here the observation of the sparse signal $\mathbf{s}^{(k)}$ separates the sparse signal model from the monitoring framework, i.e., after observing the signal $\mathbf{s}^{(k)}$, the variable $\mathcal{M}$ and the corresponding parameters $\mu$ (resp. $m$) and $\lambda$ (resp. $\sigma$) will no longer be dependent on the variables of the monitoring framework, so they can be analyzed separately as we do in Section 2.4.4.

In the next section we will describe the real world signals that will be used to develop a statistical analysis on the principal component distribution and from which we obtain a set of realizations for the signal $\mathbf{s}^{(k)}$. In Section 2.4.4 we will show that the Laplacian is a good model to represent the principal components of typical WSN data. In turn, this provides a justification for using CS in WSNs, as detailed in Section 2.4.5.

### 2.4.3 Description of Considered Signals and WSNs

The ultimate aim of WSN deployments is to monitor the evolution of a certain physical phenomenon over time. Examples of applications that require such infrastructure include monitoring for security, health-care or scientific purposes. Many different types of signals can be sensed, processed and stored, e.g., the motion of objects and beings, the heart beats, or environmental signals like the values of temperature and humidity, indoor or outdoor. Very often the density of sensor network deployments is very high and therefore sensor observations are strongly correlated in the space domain. Furthermore, the physics itself of the observed signals makes consecutive observations of a sensor node to be also temporally correlated.

The spatial and temporal correlation represents a huge potential that can be exploited designing collaborative protocols for the nodes constituting a WSN. In this perspective, we can think of reducing the energy consumption of the network by tuning the overall number of transmissions required to monitor the evolution of a given phenomenon over time. The appeal of the techniques presented in Section 2.4.1 follows from the fact that CS enables us to significantly reduce the number of samples needed to estimate a signal of interest with a certain level of quality. Clearly, the effectiveness of CS is subject to the knowledge of a transformation basis for which the observed signals result sparse.

In this section we illustrate the WSNs and the gathered signals that will be used in Section 2.4.4 to test, using the Bayesian framework presented in Section 2.4.2, whether CS and

PCA are effective for real signals, i.e., whether the real signal transformed by the PCA matrix is actually sparse.

**Networks.** In addition to our own experimental network deployed on the ground floor of the Department of Information Engineering at the University of Padova, we consider other three WSNs whose sensor reading databases are available on-line, and a further deployment called Sense&Sensitivity, whose data has been kindly provided to the authors by Dr. Thomas Watteyne of the Dust Networks, Incorporation. A brief technical overview of each of these five experimental network scenarios follows.

**W1** WSN testbed of the Department of Information Engineering (DEI) at the University of Padova, collecting data from $68$ TmoteSky wireless sensor nodes [65], [66]. The node hardware features an IEEE 802.15.4 Chipcon wireless transceiver working at $2.4$ GHz and allowing a maximum data rate of $250$ Kbps. These sensors have a TI MSP430 micro-controller with $10$ Kbytes of RAM and $48$ Kbytes of internal FLASH;

**W2** LUCE (Lausanne Urban Canopy Experiment) WSN testbed at the Ecole Polytechnique Fédérale de Lausanne (EPFL), [52]. This measurement system exploits $100$ SensorScope weather sensors which have been deployed across the EPFL campus. The node hardware is based on a TinyNode module equipped with a Xemics XE1205 radio transceiver operating in the 433, 868 and 915 MHz license-free ISM (Industry Scientific and Medical) frequency bands. Also these sensors have a TI MSP430 micro-controller;

**W3** St-Bernard WSN testbed at EPFL, [67]. This experimental WSN deployment is made of $23$ SensorScope stations deployed at the Grand St. Bernard pass at 2400 m, between Switzerland and Italy. See point W2 for a brief description of the related hardware;

**W4** CitySense WSN testbed, developed by Harvard University and BBN Technologies, [68]. CitySense is an urban scale deployment that will consist of $100$ wireless sensor nodes equipped with an ALIX 2d2 single-board computer. The transmitting interface is reconfigurable by the user and by default it operates in 802.11b/g ad hoc mode at $2.4$ GHz. Nowadays this WSN deployment counts about twenty nodes;

**W5** The Sense&Sensitivity [69] testbed is a WSN of $86$ WSN430 nodes, which embed Texas technology: a MSP430 micro-controller and a CC1100 radio chip operating in the ISM band (from 315 to 915 MHz).

| W1 (DEI WSN) | | | | | | |
|---|---|---|---|---|---|---|
| | **# of nodes** | **frame length** | **# of frames** | **starting time (G.M.T)** | **stopping time (G.M.T)** | **signals** |
| **Campaign A** | 37 | 5 min | 783 | 13/03/2009, 09:05:22 | 16/03/2009, 18:20:28 | S1 S2 S3 S4 S6 |
| **Campaign B** | 45 | 5 min | 756 | 19/03/2009, 10:00:34 | 22/03/2009, 17:02:54 | S1 S2 S3 S4 S6 |
| **Campaign C** | 31 | 5 min | 571 | 24/03/2009, 11:05:10 | 26/03/2009, 10:15:42 | S1 S2 S3 S4 S6 |
| **W2 (EPFL LUCE WSN)** | | | | | | |
| | **# of nodes** | **frame length** | **# of frames** | **starting time (G.M.T)** | **stopping time (G.M.T)** | **signals** |
| **Campaign A** | 85 | 5 min | 865 | 12/01/2007, 15:09:26 | 15/01/2007, 15:13:26 | S1 S2 S5 |
| **Campaign B** | 72 | 5 min | 841 | 06/05/2007, 16:09:26 | 09/05/2007, 14:13:26 | S1 S2 S5 |
| **Campaign C** | 83 | 30 min | 772 | 02/02/2007, 17:09:26 | 18/02/2009, 19:09:26 | S6 S7 |
| **W3 (EPFL St Bernard WSN)** | | | | | | |
| | **# of nodes** | **frame length** | **# of frames** | **starting time (G.M.T)** | **stopping time (G.M.T)** | **signals** |
| **Campaign A** | 23 | 5 min | 742 | 03/10/2007, 12:35:37 | 06/10/2007, 02:35:37 | S1 S2 S5 |
| **Campaign B** | 22 | 5 min | 756 | 19/10/2007, 12:35:37 | 22/10/2007, 03:35:37 | S1 S2 S5 |
| **Campaign C** | 22 | 30 min | 778 | 02/10/2007, 07:06:05 | 19/10/2007, 12:06:50 | S6 S7 |
| **W4 (CitySense WSN)** | | | | | | |
| | **# of nodes** | **frame length** | **# of frames** | **starting time (G.M.T)** | **stopping time (G.M.T)** | **signals** |
| **Campaign A** | 8 | 60 min | 887 | 14/10/2009, 14:01:57 | 21/11/2009, 00:01:57 | S1 |
| **Campaign B** | 8 | 60 min | 888 | 14/10/2009, 13:00:01 | 21/11/2009, 00:00:01 | S5 |
| **W5 (Sense&Sensitivity WSN)** | | | | | | |
| | **# of nodes** | **frame length** | **# of frames** | **starting time (G.M.T)** | **stopping time (G.M.T)** | **signals** |
| **Campaign A** | 77 | 15 min | 65 | 26/08/2008, 14:46:46 | 27/08/2008, 07:31:07 | S1 S3 S4 S6 |

**Table 2.2.** *Details of the considered WSN and gathered signals.*

**Signals.** From the above WSNs, we gathered seven different types of signals: **S1)** temperature; **S2)** humidity; **S3-S4)** luminosity in two different ranges ($320-730$ and $320-1100$ nm, respectively); **S5)** wind direction; **S6)** voltage and **S7)** current. Concerning the signals gathered from our testbed W1, we collected measurements from all nodes every $5$ minutes for $3$ days. We repeated the data collection for three different measurement campaigns, choosing different days of the week. Regarding the data collection from WSNs W2–W5, we studied the raw data available on-line with the aim of identifying a portion of data that could be used as a suitable benchmark for our research purposes. This task has turned out to be really challenging due to packet losses, device failures and battery consumption that are very common and frequent in currently available technology. For the acquisition of the signals we divided the time axis in frames (or time slots) such that each of the working nodes was able to produce a new sensed data per frame. Details of the signals extracted from the records of W1–W5, and organized in different campaigns, are reported schematically in Table 2.2.

### 2.4.4 Sparsity Analysis of Real Signal Principal Components

In this section we aim to infer the statistical distribution of the vector random process $\mathbf{s}$ from the samples $\{\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots, \mathbf{s}^{(T)}\}$ which are obtained from the above WSN signals. The parameter $T$ is the duration (number of time samples) of each monitoring campaign in Table 2.2.

From the theory [61] we know that signals in the PCA domain (in our case $\mathbf{s}$) have in general uncorrelated components. Also, in our particular case we experimentally verified that this assumption is good since $E[s_i s_j] \simeq E[s_i]E[s_j]$ for $i, j \in \{1, \ldots, N\}$ and $i \neq j$. In our analysis, we make a stronger assumption, i.e., we build our model of $\mathbf{s}$ considering statistical independence among its components, i.e., $p(s_1, \ldots, s_N) = \prod_{i=1}^{N} p(s_i)$. A further assumption that we make is to consider the components of $\mathbf{s}$ as stationary over the entire monitoring period[12]. The model developed following this approach leads to good results as shown in Section 2.5, and this allows us to validate these assumptions.

Owing to these assumptions, the problem of statistically characterizing $\mathbf{s}$ reduces to that of characterizing the random variables

$$s_i = \sum_{j=1}^{N} u_{ji}(x_j - \overline{x}_j) \ , \ i = 1, \ldots, N \ , \tag{2.42}$$

[12]Note that this model is able to follow also signals whose frequency content varies over time since the signal basis adapts to the data.

where the r.v. $u_{ji}$ is an element of matrix $\mathbf{U}$ in Equation (2.39) and the r.v. $x_j$ is an element of vector $\mathbf{x}$.

A statistical model for each $s_i$ can be determined through the Bayesian estimation procedure detailed below. Similarly to the approach adopted in [70], we rely upon two levels of inference.

**First level of inference.** Given a set of competitive models $\{\mathcal{M}_1, \cdots, \mathcal{M}_N\}$ for the observed phenomenon, each of them depending on the parameter vector $\boldsymbol{\theta}$, we fit each model $\mathcal{M}_i$ to the collected data denoted by $\mathcal{D}$, i.e., we find the $\boldsymbol{\theta}_{\text{MAP}}$ that maximizes the a posteriori probability density function (pdf)

$$p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_i) = \frac{p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}_i)p(\boldsymbol{\theta}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_i)} \ , \tag{2.43}$$

i.e.,

$$\boldsymbol{\theta}_{\text{MAP}} = \underset{\theta}{\text{argmax}}\, p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}_i) \ , \tag{2.44}$$

where $p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}_i)$ and $p(\boldsymbol{\theta}|\mathcal{M}_i)$ are known as the *likelihood* and the *prior* respectively, whilst the so called *evidence* $p(\mathcal{D}|\mathcal{M}_i)$ is just a normalization factor which plays a key role in the second level of inference.

**Second level of inference.** According to Bayesian theory, the most probable model is the one maximizing the posterior $p(\mathcal{M}_i|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{M}_i)p(\mathcal{M}_i)$. Hence, when the models $\mathcal{M}_i$ are equiprobable, they are ranked according to their evidence. In general, evaluating the evidence involves the computation of analytically intractable integrals. For this reason, we rank the different models according to a widely used approximation, the Bayesian Information Criterion (BIC) [71], that we define as:

$$\text{BIC}(\mathcal{M}_i) \stackrel{def}{=} \ln\left[p(\mathcal{D}|\boldsymbol{\theta}_{MAP}, \mathcal{M}_i)p(\boldsymbol{\theta}_{MAP}|\mathcal{M}_i)\right] - \frac{\ell_i}{2}\ln(T) \ , \tag{2.45}$$

where $\boldsymbol{\theta}_{MAP}$ is defined in (2.44), $\ell_i$ is the number of free parameters of model $\mathcal{M}_i$ and $T$ is the cardinality of the observed data set $\mathcal{D}$. Roughly speaking, the BIC provides insight in the selection of the best fitting model penalizing those models requiring more parameters.

According to the introduced formalism we consider $\{\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots, \mathbf{s}^{(T)}\}$ as the set of collected data $\mathcal{D}$; further, the observation of the experimental data gives empirical evidence for the selection of four statistical models $\mathcal{M}_i$ and corresponding parameter vectors $\boldsymbol{\theta}$:

$\mathcal{M}_1$ a Laplacian distribution with $\boldsymbol{\theta} = [\mu, \lambda]$, that we call $\mathcal{L}$;

$\mathcal{M}_2$ a Gaussian distribution with $\boldsymbol{\theta} = [m, \sigma^2]$, that we call $\mathcal{G}$;

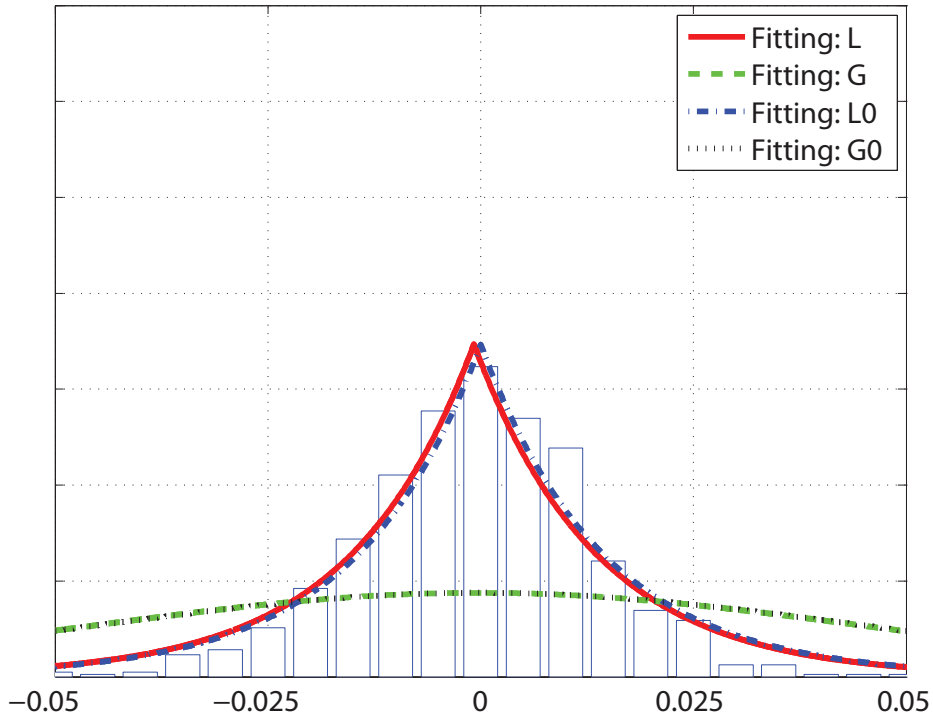



**Figure 2.14.** *Empirical distribution and model fitting for a principal component of signal S1, temperature.*

$\mathcal{M}_3$ a Laplacian distribution with $\mu = 0$ and $\theta = \lambda$, that we call $\mathcal{L}_0$;

$\mathcal{M}_4$ a Gaussian distribution with $m = 0$ and $\theta = \sigma^2$, that we call $\mathcal{G}_0$.

The space of models for each $\mathbf{s}_i$ is therefore described by the set $\{\mathcal{L}, \mathcal{G}, \mathcal{L}_0, \mathcal{G}_0\}$. In detail, for each signal $S1 - S7$ in the corresponding WSNs and campaigns of Table 2.2, we collected the $T + K$ signal samples $\left\{\mathbf{x}^{(1-K)}, \ldots, \mathbf{x}^{(-1)}, \mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots \mathbf{x}^{(T)}\right\}$ from which we computed $\left\{\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots, \mathbf{s}^{(T)}\right\}$ according to what explained in Section 2.4.1. Then, for each component $s_i, i = 1, \ldots, N$, and for each model $\mathcal{M}_i, i = 1, \ldots, 4$, we have estimated the parameters (i.e., the most probable *a posteriori*, $MAP$) that best fit the data according to (2.43). These estimations are related to the BN in Figure 2.12 and since we deal with Gaussian and Laplacian distributions, they have well known and closed form solutions [59]. In detail, for each component $s_i$:

$\mathcal{M}_1$ $\widehat{\mu} = \mu_{1/2}(s_i)$ and $\widehat{\lambda} = \frac{\sum_{k=1}^{T}\left|s_i^{(k)} - \widehat{\mu}\right|}{T}$, with $\mu_{1/2}(s_i)$ the median of the set $\left\{s_i^{(1)}, \ldots, s_i^{(T)}\right\}$;

$\mathcal{M}_2$ $\widehat{m} = \frac{\sum_{j=1}^{T} s_i^{(k)}}{T}$ and $\widehat{\sigma}^2 = \frac{\sum_{k=1}^{T}\left(s_i^{(k)} - \widehat{m}\right)^2}{T-1}$;

$\mathcal{M}_3$ $\widehat{\lambda} = \frac{\sum_{k=1}^{T}\left|s_i^{(k)}\right|}{T}$;

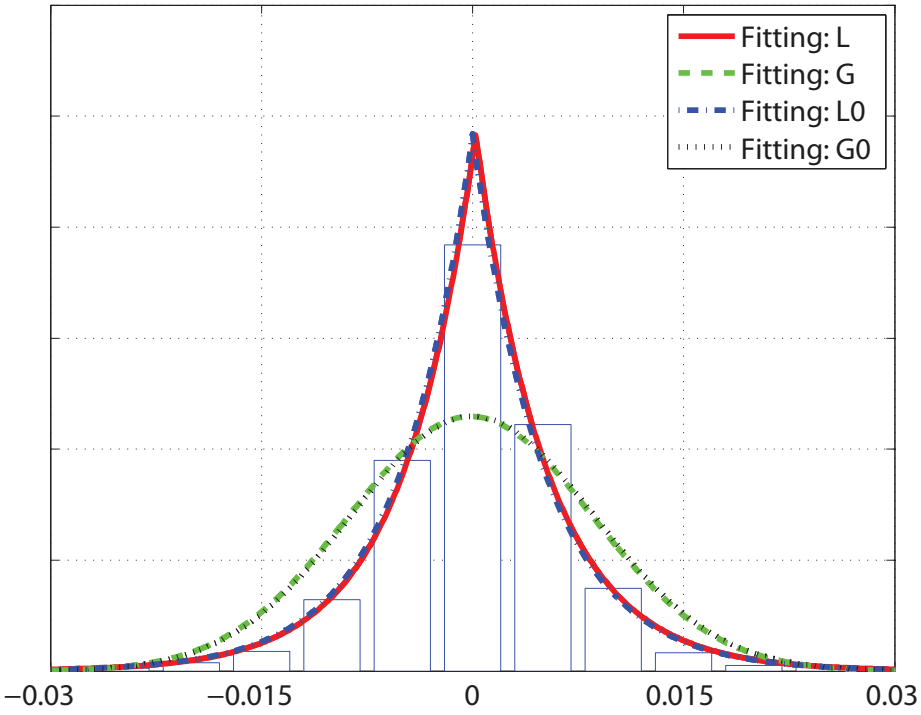



**Figure 2.15.** *Empirical distribution and model fitting for a principal component of signal S3, luminosity in the range* $320 - 730$ *nm.*

$\mathcal{M}_4$ $\widehat{\sigma}^2 = \frac{\sum_{k=1}^{T}\left(s_i^{(k)}\right)^2}{T}$.

Figures 2.14–2.15 show two examples of data fitting according to the aforementioned models; in these figures we plot the empirical distribution and the corresponding inferred statistical model for a generic principal component (but not the first one, as explained in the following) of the temperature (S1) and the luminosity (S3), respectively. Both these signals have been observed during the data collection of the campaign A, in the WSN testbed W1 (DEI). From the graphs in Figures 2.14–2.15 we see that the distribution of the principal components of our signals is well described by a Laplacian distribution. Formally, the best among the four considered models can be determined ranking them according to the Bayesian Information Criterion introduced in Equation (2.45). Since we assigned non informative priors to the model parameters, $p(\boldsymbol{\theta}_{\text{MAP}}|\mathcal{M}_i)$ is a constant for each $\mathcal{M}_i$ and therefore the BIC can be redefined as:

$$\text{BIC}(\mathcal{M}_i) \stackrel{def}{=} \ln p(\mathcal{D}|\boldsymbol{\theta}_{MAP}, \mathcal{M}_i) - \frac{\ell_i}{2}\ln(T)\ . \tag{2.46}$$

| | W1 (DEI WSN) | | | | W2 (EPFL LUCE WSN) | | | | W3 (EPFL St Bernard WSN) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\mathcal{L}$ | $\mathcal{G}$ | $\mathcal{L}_0$ | $\mathcal{G}_0$ | $\mathcal{L}$ | $\mathcal{G}$ | $\mathcal{L}_0$ | $\mathcal{G}_0$ | $\mathcal{L}$ | $\mathcal{G}$ | $\mathcal{L}_0$ | $\mathcal{G}_0$ |
| S1 (Temperature) | 1382.8 | 1042.1 | **1385.5** | 1044.9 | -36.1 | -195.3 | **-33.3** | -192.5 | -82.3 | -487.4 | **-79.3** | -484.7 |
| S2 (Humidity) | 1059.8 | 804.9 | **1062.4** | 807.60 | -992.3 | -1163.7 | **-989.5** | -1160.9 | -1473 | -1700.7 | **-1469.9** | -1697.8 |
| S3 (Light) | 2191.7 | 1690 | **2194.9** | 5078.3 | - | - | - | - | - | - | - | - |
| S4 (IR) | 1760.9 | 1154.5 | **1764.1** | 1157.4 | - | - | - | - | - | - | - | - |
| S5 (Wind) | - | - | - | - | -3694.9 | -4026.5 | **-3691.5** | -4023.6 | -3700.2 | -3850.3 | **-3697.3** | -3847.5 |
| S6 (Voltage) | 4656.9 | 3814.1 | **4660.1** | 3816.9 | 1854.1 | 1191.4 | **1856.3** | 1194.2 | 1617.8 | 1087.9 | **1619** | 1090.7 |
| S7 (Current) | - | - | - | - | -972.8 | -1520.3 | **-969.6** | -1517.3 | -1557.5 | -1877.2 | **-1554.2** | -1874.2 |
| | **W4 (CitySense WSN)** | | | | **W5 (Sense&Sensitivity WSN)** | | | | | | | |
| | $\mathcal{L}$ | $\mathcal{G}$ | $\mathcal{L}_0$ | $\mathcal{G}_0$ | $\mathcal{L}$ | $\mathcal{G}$ | $\mathcal{L}_0$ | $\mathcal{G}_0$ | | | | |
| S1 (Temperature) | -858.1 | -1094.6 | **-856.8** | -1091.9 | -127.7 | -176.1 | **-125.7** | -174.7 | | | | |
| S2 (Humidity) | - | - | - | - | - | - | - | - | | | | |
| S3 (Light) | - | - | - | - | -196.2 | -232.1 | **-194.2** | -230.6 | | | | |
| S4 (IR) | - | - | - | - | -184.4 | -227.5 | **-182.3** | -225.8 | | | | |
| S5 (Wind) | -4309.5 | -4384.2 | **-4306.4** | -4381.2 | - | - | - | - | | | | |
| S6 (Voltage) | - | - | - | - | 110 | 70.2 | **111.9** | 71.8 | | | | |
| S7 (Current) | - | - | - | - | - | - | - | - | | | | |

**Table 2.3.** *Bayesian Information Criterion (BIC) averaged over all Principal Components and relative campaigns, for each model $\mathcal{M}_1$–$\mathcal{M}_4$, for each testbed W1–W5 and each signal among S1–S7.*

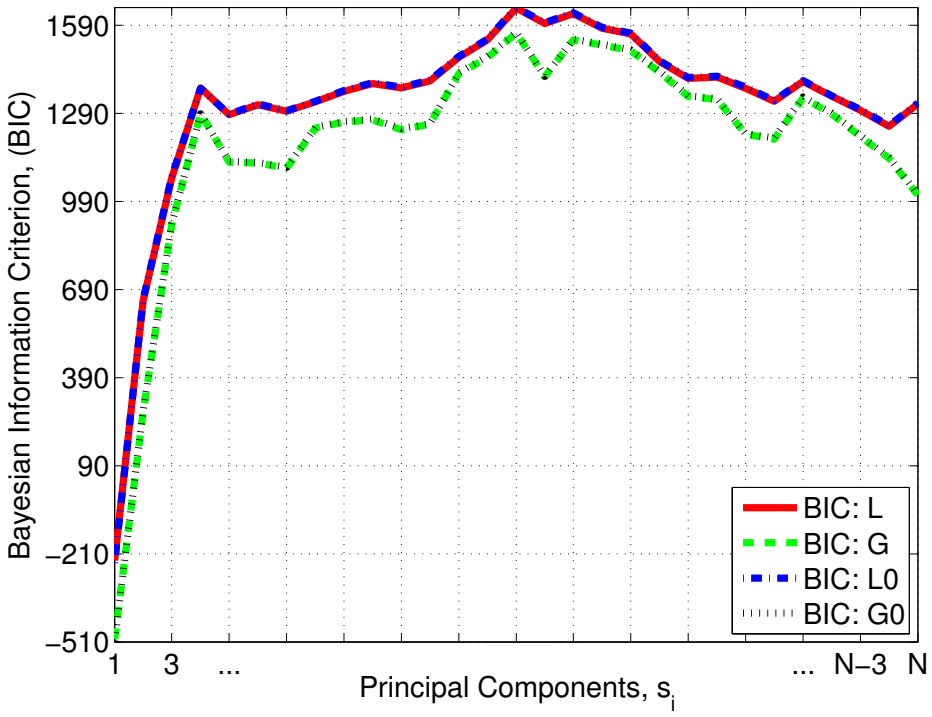



**Figure 2.16.** *Bayesian Information Criterion (BIC) per Principal Component, for each model $\mathcal{M}_1$–$\mathcal{M}_4$, WSN W1 (DEI), campaign A and signal S2, humidity.*

Figure 2.16 shows the BIC for the aforementioned humidity signal, for all its principal components and for all the considered models. From this figure we see that the Laplacian models better fit the data for all principal components $s_i$, $i = 1, 2, \ldots, N$. The average BIC for each model, for the different signals, campaigns and WSN testbeds, is shown in Table 2.3. The values of this table are computed averaging over the $N$ principal components. From these results we see that model $\mathcal{L}_0$ provides the best statistical description of the experimental data. In fact, the BIC metric is higher for Laplacian models in all cases; furthermore, $\mathcal{L}_0$ has a higher evidence with respect to $\mathcal{L}$, since it implies the utilization of a single parameter. As previously mentioned, the over-parameterization of the model is penalized according to the factor $T^{\frac{-\ell}{2}}$ (see Equation (2.46)). Based on the above results, we can conclude that the Laplacian model describes slightly better than the Gaussian one the real signal principal components obtained according to our proposed framework, for all the considered signals. Furthermore, it is worth noting that the first principal components (to be more precise, the first $K-1$ principal components[13] of the signal, where $K$ is the training set length) have

[13]Note that, according to Equation (2.41), the matrix $\mathbf{U}^{(k)}$ is obtained from the elements of the training set $\mathcal{X}^{(k)}$ minus their mean, i.e., from the set $\left\{\mathbf{x}^{(k-1)} - \overline{\mathbf{x}}^{(k)}, \mathbf{x}^{(k-2)} - \overline{\mathbf{x}}^{(k)}, \cdots, \mathbf{x}^{(k-K)} - \overline{\mathbf{x}}^{(k)}\right\}$ which spans a vector

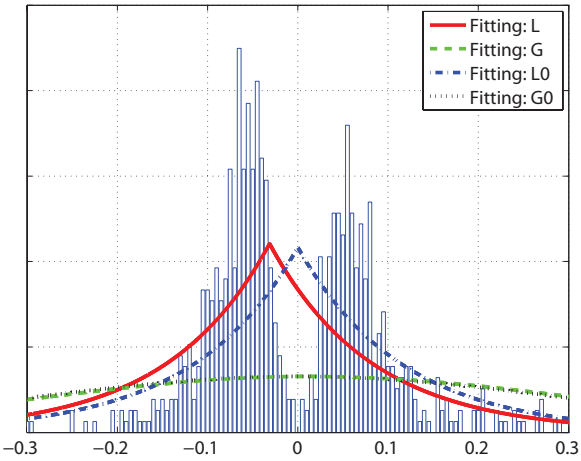



**Figure 2.17.** *Empirical distribution and model fitting for the first principal component of signal S1, temperature.*

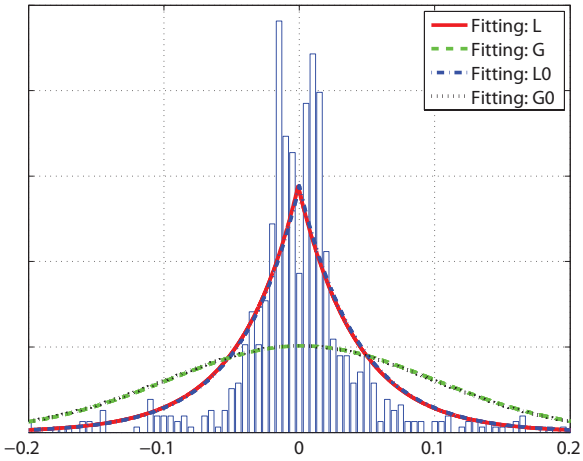



**Figure 2.18.** *Empirical distribution and model fitting for the first principal component of signal S3, luminosity in the range* $320 - 730$ *nm.*

different statistics from the remaining ones, in terms of both signal range dynamics and amplitude of the components. This is due to the fact that the first $K - 1$ components actually map the observed signal into the training set vector space, instead the remaining ones are random projections of the signals. The former capture the "core" of the signal $\mathbf{x}$, the latter allow to recover its details which can lie outside the linear span of the training data. In our simulations we set $K = 2$, in accordance to the rationale presented in the Appendix 2.D, so that only the first principal component shows a behavior different from the one illustrated in Figures 2.14–2.15 as reported in Figures 2.17–2.18. In any case, the Laplacian model still fits better the observed data compared to the Gaussian one.

### 2.4.5 Bayesian MAP Condition and CS Recovery for Real Signals

In the previous section we have seen that the Laplacian model is a good representation for the principal components of typical WSN signals. This legitimates the use of CS in WSNs when it is exploited according to the framework presented in Section 2.4.1; to support this claim we review in this section a Bayesian perspective that highlights the equivalence between the output of the CS reconstruction algorithm and the solution that maximizes the posterior probability in Equation (2.43).

Assume a sink is placed in the center of a WSN with $N$ sensor nodes and let our goal

space of dimension at most $K - 1$.

be to determine at each time $k$ all the $N$ sensor readings by just collecting at the sink a small fraction of them. To this end we exploit the joint CS and PCA scheme presented in Section 2.4.1. Equations (2.39)–(2.41) show that the considered framework does not depend on the particular topology considered; the only requirement is that the sensor nodes be ordered (e.g., based on the natural order of their IDs). Our monitoring application can be seen, at each time $k$, as an interpolation problem: from a sampled $M$-dimensional vector $\mathbf{y}^{(k)} = \mathbf{\Phi}\mathbf{x}^{(k)} \in \mathbb{R}^M$, we are interested in recovering, via interpolation, the signal $\mathbf{x}^{(k)} \in \mathbb{R}^N$. Typically (e.g., see [70]) this problem can be solved through a linear interpolation on a set $\mathcal{F}$ of $h$ basis functions $\mathbf{f}_i \in \mathbb{R}^N$, i.e., $\mathcal{F} = \{\mathbf{f}_1, \cdots, \mathbf{f}_h\}$. We can assume that the interpolated function has the form:

$$\mathbf{x}^{(k)} = \overline{\mathbf{x}}^{(k)} + \sum_{i=1}^{h} \theta_i \mathbf{f}_i \ . \tag{2.47}$$

In accordance to what explained in Section 2.4.1, at each time $k$ we can do the following associations: the columns of the PCA matrix $\mathbf{U}^{(k)}$ as the set of $h = N$ basis functions, i.e., $\mathcal{F} = \{\mathbf{f}_1, \cdots, \mathbf{f}_N\} = \{\mathbf{u}_1^{(k)}, \cdots, \mathbf{u}_N^{(k)}\} = \mathcal{U}^{(k)}$; the sparse vector $\mathbf{s}^{(k)} = (s_1^{(k)}, \cdots, s_N^{(k)})^T$ as the parameter vector $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_N)^T$. In this perspective the interpolated function has the form (see Equation (2.39))

$$\mathbf{x}^{(k)} - \overline{\mathbf{x}}^{(k)} = \sum_{i=1}^{N} s_i^{(k)} \mathbf{u}_i^{(k)} \ . \tag{2.48}$$

A Bayesian approach would estimate the most probable value of $\mathbf{s}^{(k)} = (s_1^{(k)}, \cdots, s_N^{(k)})^T$ by maximizing a posterior pdf of the form $p(\mathbf{s}^{(k)}|\mathbf{y}^{(k)}, \mathcal{U}^{(k)}, \mathcal{M})$, where $\mathcal{M}$ is a plausible model for the vector $\mathbf{s}^{(k)}$. To avoid confusion, it is important to note that in this section the interpretation of all the variables involved is slightly different from the one adopted in Section 2.4.4. In detail, now the vector $\mathbf{s}^{(k)}$ is seen as the parameter vector $\boldsymbol{\theta}$ in Equation (2.43), whilst the vector $\mathbf{y}^{(k)}$ represents the set $\mathcal{D}$ of collected data. Moreover, the observed phenomenon $\mathbf{x}^{(k)}$ is modeled through both a set $\mathcal{U}^{(k)}$ of basis functions (i.e., the columns of the matrix $\mathbf{U}^{(k)}$) and a model $\mathcal{M}$ for the parameter vector $\mathbf{s}^{(k)}$, according to the BN in Figure 2.13. In Equation (2.43) we indicated with the symbol $\mathcal{M}_i$ a possible model for the observed phenomenon: here that symbol is replaced with the couple $(\mathcal{U}^{(k)}, \mathcal{M})$, where $\mathcal{M}$ refers directly to $\mathbf{s}^{(k)}$. Using the symbol $\mathcal{M}$ to indicate a model for $\mathbf{s}^{(k)}$ (even if $\mathbf{s}^{(k)}$ is now interpreted as the parameter vector $\theta$) allows us to highlight the correspondence between the adoption of a particular model for $\mathbf{s}^{(k)}$ and the results of the study carried out in Section 2.4.4. This correspondence will become clear in the following.

As in [70], we assume also that $\mathcal{M}$ can be specified by a further parameter set $\boldsymbol{\alpha}$ (called

hyper-prior) related to $\mathbf{s}^{(k)}$, so that the posterior can be written as

$$p(\mathbf{s}^{(k)}|\mathbf{y}^{(k)},\mathcal{U}^{(k)},\mathcal{M}) = \int p(\mathbf{s}^{(k)}|\mathbf{y}^{(k)},\boldsymbol{\alpha},\mathcal{U}^{(k)},\mathcal{M})p(\boldsymbol{\alpha}|\mathbf{y}^{(k)},\mathcal{U}^{(k)},\mathcal{M})\,\mathrm{d}\boldsymbol{\alpha}\,.$$

If the hyper-prior can be inferred from the data and has non zero values $\widehat{\boldsymbol{\alpha}}$, maximizing the posterior corresponds to maximizing $p(\mathbf{s}^{(k)}|\mathbf{y}^{(k)},\widehat{\boldsymbol{\alpha}},\mathcal{U}^{(k)},\mathcal{M})$, that as shown in [70] corresponds to maximizing the following expression

$$\begin{aligned} p(\mathbf{s}^{(k)}|\mathbf{y}^{(k)},\mathcal{U}^{(k)},\mathcal{M}) &\propto p(\mathbf{s}^{(k)}|\mathbf{y}^{(k)},\widehat{\boldsymbol{\alpha}},\mathcal{U}^{(k)},\mathcal{M}) \\ &= \frac{p(\mathbf{y}^{(k)}|\mathbf{s}^{(k)},\mathcal{U}^{(k)})p(\mathbf{s}^{(k)}|\widehat{\boldsymbol{\alpha}},\mathcal{M})}{p(\mathbf{y}^{(k)}|\widehat{\boldsymbol{\alpha}},\mathcal{U}^{(k)},\mathcal{M})}\,, \end{aligned} \tag{2.49}$$

where $p(\mathbf{y}^{(k)}|\mathbf{s}^{(k)},\mathcal{U}^{(k)})$ and $p(\mathbf{s}^{(k)}|\widehat{\boldsymbol{\alpha}},\mathcal{M})$ are the likelihood function and the prior, respectively, while $p(\mathbf{y}^{(k)}|\widehat{\boldsymbol{\alpha}},\mathcal{U}^{(k)},\mathcal{M})$ is a normalization factor. The parameters $\widehat{\boldsymbol{\alpha}}$ are estimated maximizing the evidence $p(\mathbf{y}^{(k)}|\boldsymbol{\alpha},\mathcal{U}^{(k)},\mathcal{M})$, which is a function of $\boldsymbol{\alpha}$. Note that here the hyper-prior plays, in regard to $\mathbf{s}^{(k)}$, exactly the same role as the parameter vector $\boldsymbol{\theta}$ in the previous section, where $\mathbf{s}^{(k)}$ was interpreted as the collected data set $\mathcal{D}$ of the observed phenomenon; for example, if we choose $\mathcal{M} = \mathcal{L}_0$ for $\mathbf{s}^{(k)}$ then $\boldsymbol{\alpha} = \lambda$, i.e., the hyper-prior is the scale parameter of the Laplacian prior assigned to $\mathbf{s}^{(k)}$.

In Equation (2.48), without loss of generality we can assume that $\overline{\mathbf{x}}^{(k)} = 0$, thus the constraints on the relationship between $\mathbf{y}^{(k)}$ and $\mathbf{s}^{(k)}$ can be translated into a likelihood of the form (see Equation (2.40)):

$$p(\mathbf{y}^{(k)}|\mathbf{s}^{(k)},\mathcal{U}^{(k)}) = \delta(\mathbf{y}^{(k)}, \boldsymbol{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)})\,, \tag{2.50}$$

where $\delta(x,y)$ is 1 if $x = y$ and zero otherwise. In Section 2.4.4, we have seen that the statistics of vector $\mathbf{s}^{(k)}$ is well described by a Laplacian density function with location parameter $\mu$ equal to 0 ($\mathcal{L}_0$). This pdf is widely used in the literature [32, 60] to statistically model sparse random vectors and, owing to the assumption of statistical independence of the components of $\mathbf{s}^{(k)}$, we can write it in the form:

$$p(\mathbf{s}^{(k)}|\widehat{\boldsymbol{\alpha}},\mathcal{M}) = p(\mathbf{s}^{(k)}|\widehat{\lambda},\mathcal{L}_0) = \frac{e^{-\widehat{\lambda}\sum_{i=1}^{N}|s_i^{(k)}|}}{(2/\widehat{\lambda})^N}\,. \tag{2.51}$$

In this equation, all the components of $\mathbf{s}^{(k)}$ are assumed to be equally distributed. If (2.49) holds, we can obtain the following posterior:

$$\begin{aligned} p(\mathbf{s}^{(k)}|\mathbf{y}^{(k)},\mathcal{U}^{(k)},\mathcal{L}_0) &\propto p(\mathbf{s}^{(k)}|\mathbf{y}^{(k)},\widehat{\lambda},\mathcal{U}^{(k)},\mathcal{L}_0) \\ &\propto p(\mathbf{y}^{(k)}|\mathbf{s}^{(k)},\mathcal{U}^{(k)})p(\mathbf{s}^{(k)}|\widehat{\lambda},\mathcal{L}_0). \end{aligned} \tag{2.52}$$

Using (2.50)–(2.52), maximizing the posterior corresponds to solving the problem

$$\begin{aligned}
& \underset{\mathbf{s}^{(k)}}{\operatorname{argmax}}\, p(\mathbf{s}^{(k)}|\mathbf{y}^{(k)},\mathcal{U}^{(k)},\mathcal{L}_0) \\
= & \underset{\mathbf{s}^{(k)}}{\operatorname{argmax}}\, p(\mathbf{y}^{(k)}|\mathbf{s}^{(k)},\mathcal{U}^{(k)})p(\mathbf{s}^{(k)}|\widehat{\lambda},\mathcal{L}_0) \\
= & \underset{\mathbf{s}^{(k)}}{\operatorname{argmax}}\, \delta(\mathbf{y}^{(k)},\mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)})\frac{e^{-\widehat{\lambda}\sum_{i=1}^{N}|s_i^{(k)}|}}{(2/\widehat{\lambda})^N} \\
= & \underset{\mathbf{s}^{(k)}}{\operatorname{argmin}} \sum_{i=1}^{N}|s_i^{(k)}|, \text{ given that } \mathbf{y}^{(k)}=\mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)} \\
= & \underset{\mathbf{s}^{(k)}}{\operatorname{argmin}} \|\mathbf{s}^{(k)}\|_1, \text{ given that } \mathbf{y}^{(k)}=\mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)}\ ,
\end{aligned} \tag{2.53}$$

which is the convex optimization problem solved by the CS reconstruction algorithms (see [16] and [33]). In our approach, unlike in the classical CS problems, the sparsification matrix $\mathbf{U}^{(k)}$ is not fixed but varies over time adapting itself to the current data.

**Example of Application**

In light of the above results we are legitimate in using CS for WSN monitoring applications via data reconstruction. In particular, here we describe with illustrative purposes a naive solution that, despite its simplicity, provides useful insight into the advantages that can be obtained by exploiting the CS reconstruction algorithm. This solution is detailed and evaluated thoroughly in the Appendix 2.D.

We assume a data gathering technique that alternates between a *training phase* and a *monitoring phase*. During the former phase, all sensors transmit their data to a central unity (e.g., an application server) which estimates the signal statistics in terms of mean and covariance. This information is subsequently used to reconstruct the WSN signal in the latter phase, where only a small fraction of nodes transmit. Specifically, we assume that the training phase lasts $K_1$ time samples or collection rounds. During this period of time, all sensors transmit their data to the application server which therefore receives $K_1$ complete vectors with $N$ measurements[14]. During the monitoring phase, which lasts $K_2$ time samples, each sensor node transmits its data with probability $p_{\text{tx}} = L/N$, where $L < N$. The value of $L$, and therefore of $p_{\text{tx}} = L/N$, can be chosen by a central entity, based on some metric to optimize, and sent to the WSN nodes according to a feedback-like mechanism. Thus, for $K_2$ time samples each sensor transmits its data with this probability (i.e., on average only

[14]In case there is an error during the transmissions, so that not all $N$ measurements are received, the server will use as this training set the last complete training set received.

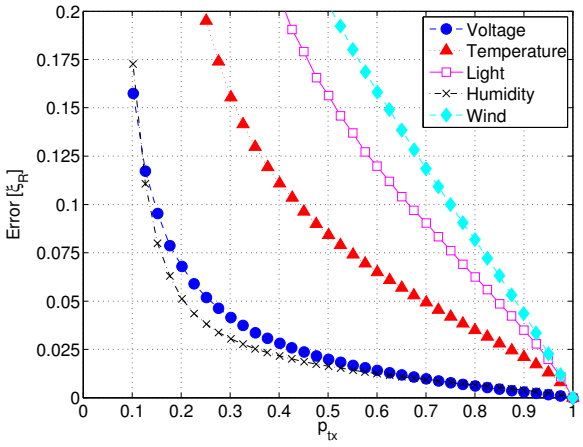



**Figure 2.19.** *Average reconstruction error for different types of signals.*

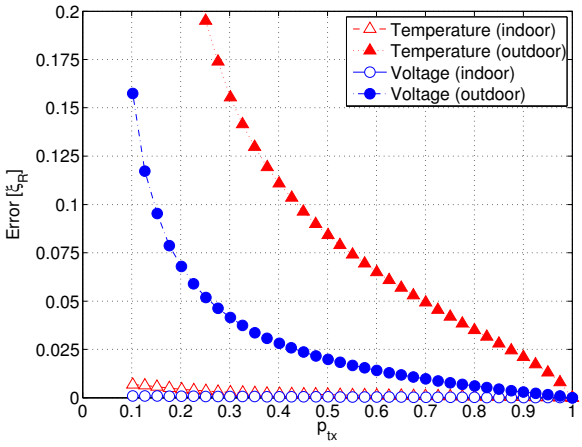



**Figure 2.20.** *Average reconstruction error for two types of signals. Comparison between signals gathered in indoor and outdoor environments, respectively.*

$L$ of the $N$ sensors transmit at each time sample). The interleaving between training and monitoring phases can be viewed as follows:

$$\ldots, \underbrace{\mathbf{y}^{(k)}, \ldots, \mathbf{y}^{(k+K_1-1)}}, \underbrace{\mathbf{y}^{(k+K_1)}, \ldots, \mathbf{y}^{(k+K_1+K_2-1)}}, \ldots$$

$$\begin{array}{cc} \text{training phase} & \text{monitoring phase} \\ p_{\text{tx}} = 1 & p_{\text{tx}} = L/N \\ \dim(\mathbf{y}^{(j)}) = N & E\left[\dim(\mathbf{y}^{(j)})\right] = L \end{array}$$

where $\dim(\mathbf{y}^{(j)})$ is the number of the components of $\mathbf{y}^{(j)}$ and $E[\dim(\mathbf{y}^{(j)})]$ is the expected value of $\dim(\mathbf{y}^{(j)})$.

With this simple implementation, the overall energy consumption of the network can be reduced by limiting the number of transmitting sensor nodes (low $p_{\text{tx}}$) during the monitoring phase. The entire signal of interest is then reconstructed at the application server using CS. As a first step, at each time $k$ we can associate to the vector $\mathbf{y}^{(k)}$ the corresponding matrix $\mathbf{\Phi}^{(k)}$, based on the IDs of the transmitting nodes, according to (2.37). Then we can exploit the samples of the last recorded training set to infer the reconstructed value of $\widehat{\mathbf{x}}^{(k)}$. The statistics necessary to build the sparsifying matrix $\mathbf{U}^{(k)}$ are derived from the samples of the recorded training set (i.e., mean and covariance matrix), and the CS recovery problem is solved via a convex optimization problem, see Section 2.1.2.

Figures 2.19–2.20 show the quality of the monitored signal reconstruction (at the applica-

tion server) $vs$ the transmission probability $p_{\text{tx}}$. The results are obtained implementing the simple above mechanism combining software simulation of protocol stack operation with the real measurements described in Section 2.4.3. The x-axis of the figures represents the $p_{\text{tx}}$ adopted during the monitoring phase, while the y-axis represents the average relative reconstruction error in the whole simulation ($k = 1, \ldots, K$), defined as

$$\overline{\xi}_R = \frac{1}{K} \sum_{k=1}^{K} \xi_R^{(k)} \ , \tag{2.54}$$

where $\xi_R^{(k)}$ is the relative reconstruction error at time $k$, i.e.,

$$\xi_R^{(k)} = \frac{\|\mathbf{x}^{(k)} - \widehat{\mathbf{x}}^{(k)}\|_2}{\|\mathbf{x}^{(k)}\|_2} \ . \tag{2.55}$$

In our simulations we set the length of the training phase to $K_1 = 2$ and the length of the monitoring phase to $K_2 = 4$, according to the rationale presented in the Appendix 2.D. In Figure 2.19 we plot the recovery performance achieved with different kinds of signals. We note that for highly correlated signals like voltage and humidity, the reconstruction error is sufficiently small, i.e., below $\overline{\xi}_R < 0.01$, for relatively small values of $p_{\text{tx}} \approx 0.6$. Instead for more unpredictable signals like luminosity and wind the error increases sharply with decreasing $p_{\text{tx}}$; in this case an error below $\overline{\xi}_R < 0.05$ is only achievable with $p_{\text{tx}} > 0.9$. In Figure 2.20 we make a comparison, instead, between signals of the same kind but measured in different environments, i.e., in indoor and outdoor environments. In detail, the indoor environment here considered is the WSN testbed W1 (DEI), where the nodes have been placed on the ground floor of the Information Engineering Department of the University of Padova, whilst take W2 (EPFL LUCE) as an outdoor WSN testbed since its nodes have been placed outside, throughout the EPFL campus. Still, the possibility of reducing the transmission cost, given a desired quality threshold, strongly depends on the signal statistics. In case of indoor signals (high correlation and low variation) we can have a reconstruction error below $\overline{\xi}_R < 0.01$ even with $p_{\text{tx}} \approx 0.1$, which leads to enormous saving in transmission energy. Conversely, with outdoor signals, whose lower correlation depends also on the wider extension of the WSN itself, we need $p_{\text{tx}} > 0.7$ to make the error $\overline{\xi}_R$ go below $0.05$. Even if the possible gains depend on the actual statistic of the observed signal, the proposed approach, despite its simplicity, adapts to the monitored process and allows us to achieve important savings in all the considered cases. This open an interesting perspective for an useful and effective exploitation of CS in WSN, which is explained in Section 2.5.

## 2.5 Application of CS in a Monitoring Framework for WSN

Here we present a lightweight and self-adapting framework called *SCoRe1* for the estimation of large data sets with high accuracy through the collection of a small number of sensor readings. Legitimated by the analysis carried out in Section 2.4, this framework is based on the joint use of CS and PCA to devise a scheme where the processing of the signal is only required at the sink, whereas data gathering and routing are independent of it. A detailed description of SCoRe1 is presented in Section 2.5.1. As mentioned at the beginning of this chapter, the main objective of our framework is to be very general, i.e., suitable to be implemented as protocol for a monitoring application independently of the observed signal. This requirement is very appealing when we think about a network of nodes equipped with different sensors, and therefore capable of sensing different signals. We do not want protocols specifically designed for signals with given statistical characteristics, so that a node should select the right protocol up to the current sensed signal. Conversely, we would like to have a transmission protocol totally unaware of the observed signal characteristics, but nevertheless able to adapt to them. Further, we stress that SCoRe1 is proposed for WSNs, but it can be readily applied to other types of network infrastructures that require the approximation of large and distributed datasets with spatial or temporal correlation.

It is worth to note that traditionally CS is exploited to jointly perform data compression and aggregation, see Section 2.1. Within our framework, instead, we use the CS recovery mechanism as an interpolation technique and therefore in the following we compare it against well-known data-fitting methods, presented in Section 2.5.2. In detail, we still refer to the Bayesian theory [58, 59, 72] as in the previous section, and consider that signals of interest can be approximated according to (i) a deterministic approach, i.e., through a proper fitting of the collected measurements, as the case of the spline method, and to (ii) a probabilistic approach, i.e., the signal is estimated from the collected measurements and some a priori statistical knowledge of the signal, as the case of CS or the Least Square Error (LSE) method. The integration of CS as interpolation technique into an actual network framework for signal monitoring can be regarded as one important contribution of our research, and in Section 2.5.3 we show that, within our framework, CS performs as good as or better than the other state-of-the-art techniques analyzed.

### 2.5.1 SCoRe1: Sensing, Compression an Recovery through ON-line Estimation

In Figure 2.21, a diagram shows the logic blocks which compose our iterative monitoring framework called SCoRe1 (Sensing, Compression an Recovery through ON-line Estimation).

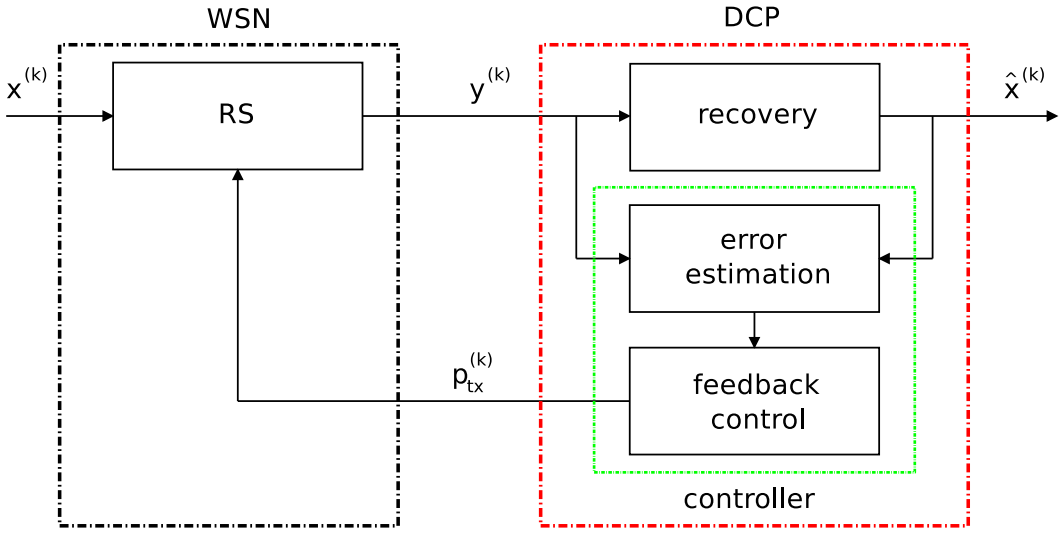



**Figure 2.21.** *Diagram of the proposed sensing, compression and recovery scheme. Note that the* Controller*, which includes the* Error estimator *and the* Feedback Control *blocks, is a characteristic of* SCoRe1 *and is not present in the other DC techniques.*

At each time $k$ the sink, which we call Data Collection Point (DCP), collects a compressed version $\mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{x}^{(k)}$, $\mathbf{y}^{(k)} \in \mathbb{R}^L$, of the original signal $\mathbf{x}^{(k)} \in \mathbb{R}^N$. As seen in Section 2.4, the routing matrix $\mathbf{\Phi}^{(k)} \in \mathbb{R}^{L\times N}$, with $L \leq N$, has one element equal to 1 per row and at most one element equal to 1 per column, indicating which nodes transmit their data sample to the DCP at time $k$, while all the other elements are equal to zero. Thus, the elements in $\mathbf{y}^{(k)} \in \mathbb{R}^L$ are a subset of those in $\mathbf{x}^{(k)}$ (spatial sampling). Note that reducing the number of nodes that transmit to the DCP is a key aspect as each sensor is supposed to be a tiny battery powered sensing unit with a finite amount of energy that determines its lifetime. At each time $k$ the transmitting nodes are chosen in a distributed way according to a simple Random Sampling (RS) technique to be executed in each node of the WSN, as we detail again shortly. The DCP can be the sink of the WSN or a remote server that is not battery powered so it does not have stringent energy requirements and has enough computational resources to execute signal recovery algorithms. The DCP is responsible for collecting the compressed data $\mathbf{y}^{(k)}$, sending a feedback to the WSN and recovering the original signal from $\mathbf{y}^{(k)}$.

In the following we provide a detailed description of all the blocks that form SCoRe1.

**Wireless Sensor Network (WSN)**

The geometry of the considered deployment is not important, i.e., the nodes can be placed arbitrarily in a given area. Our framework, in fact, is flexible and does not depend on a specific topology; the only requirement is that the sensor nodes can be ordered, e.g., based on their IDs. Multi-hops paths are taken into account for transmission energy computation by assigning a weight to each node proportionally to its distance from the sink. The actual WSN deployments considered in our study have been described in Section 2.4.3.

**Random Sampling (RS)**

The RS scheme is used to decide in a fully distributed way which sensors transmit their data to the sink and which remain silent, at any given time $k$. This scheme has been chosen because it allows us to have a simple and general solution that can easily adapt to different signal characteristics and changes. In detail, at each time $k$ each sensor node decides, with probability $p_{tx}^{(k)}$, whether to transmit its measurement to the DCP. This decision is made independently of the past and of the behavior of the other nodes, so there is no need for a large memory in each sensor, nor for further control packets within the network. The probability of transmission $p_{tx}^{(k)}$ can be fixed beforehand and kept constant, or can be varied as a function of the reconstruction error.

**Data Collection Point (DCP)**

The role of DCP is threefold: (1) it receives as input $\mathbf{y}^{(k)}$ and returns the reconstructed signal $\widehat{\mathbf{x}}^{(k)}$; (2) it adapts $p_{tx}^{(k)}$ and sends its new value to the sensor nodes; this is done to reduce the number of transmissions in the network while bounding the reconstruction error; (3) it provides the recovery block with a training set $\widehat{\mathcal{T}}_K^{(k)}$ for $\mathbf{x}^{(k)}$. This training set is used to infer the structure of the signal, which is then exploited by the signal recovery algorithm. $\widehat{\mathcal{T}}_K^{(k)}$, at each time sample $k$, is formed by the $K$ previously reconstructed signals $\widehat{\mathbf{x}}^{(j)}$ for $j < k$, so it can be written as $\widehat{\mathcal{T}}_K^{(k)} = \{\widehat{\mathbf{x}}^{(k-K)}, \ldots, \widehat{\mathbf{x}}^{(k-1)}\}$. Determine a proper value for $K$ is a delicate issue to deal with. The right choice of this parameter depends on the stationarity of $\mathbf{x}^{(k)}$ and so far it we have determined it only empirically, see Appendix 2.D.

**Recovery**

The recovery method adopted in our framework is based on a joint use of CS and PCA, as explained in Section 2.4.1. In detail, let us consider the training set provided by the DCP, namely $\widehat{\mathcal{T}}_K^{(k)} = \{\widehat{\mathbf{x}}^{(k-K)}, \ldots, \widehat{\mathbf{x}}^{(k-1)}\}$. This training set contains $K$ previously recovered signals: they can be initialized to $K$ entirely collected signals and then updated over time. Using PCA and substituting $\mathcal{X}_K^{(k)}$ with $\widehat{\mathcal{T}}_K^{(k)}$, we can repeat all the steps presented in Section 2.4.1 and obtain again Equation (2.41), i.e.,

$$\widehat{\mathbf{x}}^{(k)} = \overline{\mathbf{x}}^{(k)} + \mathbf{U}^{(k)}\widehat{\mathbf{s}}^{(k)} \ .$$

Now, the DCP can update $\widehat{\mathcal{T}}_K^{(k)}$ by substituting the oldest signal contained in it with the currently computed $\widehat{\mathbf{x}}^{(k)}$. Note, that here $\widehat{\mathcal{T}}_K^{(k)}$ is made of reconstructed signals whilst in Section 2.4 we assumed to know $\mathcal{X}_K^{(k)}$, i.e., the set made by the exact $K$ previous samples of the observed process at time $k$, and based on this we legitimate the use of CS with real WSN signals. Therefore, it is important to make $\widehat{\mathcal{T}}_K^{(k)}$ a good approximation for $\mathcal{X}_K^{(k)}$ (indeed, this means that we want a good reconstruction of $\mathbf{x}^{(k)}$ at each time sample $k$): this can be done by estimating and keeping bounded the reconstruction error; the last two blocks of our framework are in charge of this task.

**Error Estimation**

The reconstruction error that we want to estimate is again the quantity given by Equation (2.55), i.e.,

$$\xi_R^{(k)} = \frac{\|\mathbf{x}^{(k)} - \widehat{\mathbf{x}}^{(k)}\|_2}{\|\mathbf{x}^{(k)}\|_2} \ ,$$

where $\widehat{\mathbf{x}}^{(k)}$ is the signal reconstructed by the recovery block at time $k$. Note that at the DCP we do not have $\mathbf{x}^{(k)}$, but only $\mathbf{y}^{(j)} = \mathbf{\Phi}^{(j)}\mathbf{x}^{(j)}$ and $\widehat{\mathbf{x}}^{(j)}$, for $j \leq k$. Since the quantity $\xi_0^{(k)} = \|\mathbf{y}^{(k)} - \mathbf{\Phi}^{(k)}\widehat{\mathbf{x}}^{(k)}\|_2 / \|\mathbf{y}^{(k)}\|_2$ is always zero, due to the fact that the received samples are reconstructed perfectly, i.e., $\mathbf{\Phi}^{(k)}\widehat{\mathbf{x}}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{x}^{(k)}$, one might use some heuristics to calculate the error from the past samples. In our study we use the following formula:[15]

$$\xi^{(k)} = \left\| \begin{bmatrix} \mathbf{y}^{(k)} \\ \mathbf{y}^{(k-1)} \end{bmatrix} - \begin{bmatrix} \mathbf{\Phi}^{(k)}\widehat{\mathbf{x}}^{(k-1)} \\ \mathbf{\Phi}^{(k-1)}\widehat{\mathbf{x}}^{(k)} \end{bmatrix} \right\|_2 \cdot \left( \left\| \begin{bmatrix} \mathbf{y}^{(k)} \\ \mathbf{y}^{(k-1)} \end{bmatrix} \right\|_2 \right)^{-1} \ . \tag{2.56}$$

With this heuristic we compare the spatial samples collected at time $k$, i.e., $\mathbf{y}^{(k)}$, with the reconstructed values at time $k-1$, i.e., $\widehat{\mathbf{x}}^{(k-1)}$, sampled in the same points of the compressed

[15] We tried other heuristics and verified through extensive simulation that they perform similarly and worse than the one in (2.56). These are not listed here as they do not provide additional insights.

values at time $k$, i.e., $\mathbf{\Phi}^{(k)}\widehat{\mathbf{x}}^{(k-1)}$. Then we compare the same signals inverting $k$ and $k-1$. Note that $\xi^{(k)}$ does not only account for the reconstruction error but also for the signal variability. This introduces a further approximation to the error estimate, but on the other hand it allows the protocol to react faster if the signal changes abruptly and this is a desirable feature. In fact, if the signal significantly differs from time $k-1$ to time $k$, $\xi^{(k)}$ will be large and this will translate into a higher $p_{tx}^{(k+1)}$ (see below).

**Feedback Control**

This block calculates the new $p_{tx}^{(k+1)}$ for the next time $k+1$ and sends a broadcast message with this new value to the network nodes. The calculation of the new $p_{tx}$ is made according to a technique similar to TCP's congestion window adaptation, where $p_{tx}$ is exponentially increased in case the error is above a defined error threshold $\tau$ (to quickly bound the error) and is linearly decreased otherwise. In detail, we define the constants $C_1 \in [1, +\infty[$, $C_2 \in \{1, 2, \ldots, N\}$ and $p_{tx}^{\min}$, the minimum value allowed for the probability of transmission, and we calculate the new probability of transmission as:

$$p_{tx}^{(k+1)} = \begin{cases} \min\left\{p_{tx}^{(k)} C_1, 1\right\} & \text{if } \xi^{(k)} \geq \tau \\ \max\left\{p_{tx}^{(k)} - C_2/N, p_{tx}^{\min}\right\} & \text{if } \xi^{(k)} < \tau \ . \end{cases} \tag{2.57}$$

In the Appendix 2.E we report discussions and some performance evaluations to motivate our choices for the above blocks of SCoRe1. In the next section, instead, we present state-of-the-art data-fitting techniques that can be used as alternative to CS. Our main intent is to prove that Compressive Sensing, used within our iterative monitoring framework, can be successfully exploited for networking.

### 2.5.2 Data recovery from an incomplete measurement set

The recovery algorithm (see Figure 2.21) is executed at the DCP and, at any time $k$, it tries to recover the original signal $\mathbf{x}^{(k)} \in \mathbb{R}^N$ from its compressed version $\mathbf{y}^{(k)} \in \mathbb{R}^M$, with $M \leq N$. To this end, in the previous section we presented a mechanism that jointly exploit CS and PCA for signal interpolation in WSN and that from here on we will call **CS-PCA**. Obviously, many alternatives exist in literature, each one of those based on a particular signal model. Given a signal model, theoretical analysis can tell us which is the best, or the optimal, recovery mechanism to adopt, see Section 2.4. However, when we apply the chosen

mechanism to real signals, we can obtain unsatisfactory results if the chosen model does not capture well the reality. It is important to note that, generally, a signal model does not only capture the signal itself, but it can model how this signal appears when processed according to a well specified procedure. In Section 2.4, we have seen that the principal components of real signals, computed exploiting $K$ past sample of the signals themselves, are well modeled by a Laplacian distribution. With this statistical model the CS recovery algorithm results to be optimal.

In what follows, we first review well-known state-of-the-art interpolation techniques that, according to what seen in the previous sections, formally solve the following problem:

**Problem 7** (Interpolation Problem)**.** *Estimate* $\widehat{\mathbf{x}}^{(k)}$ *(such that* $\|\widehat{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)}\|_2 / \|\mathbf{x}^{(k)}\|_2 \simeq 0$*) knowing that* $\mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{x}^{(k)}$*, where* $\mathbf{y}^{(k)} \in \mathbb{R}^L$*,* $L \leq N$ *and* $\mathbf{\Phi}^{(k)}$ *is an* $[L \times N]$ *sampling matrix (i.e., all rows of* $\mathbf{\Phi}^{(k)}$ *contain exactly one element equal to* $1$ *and all columns of* $\mathbf{\Phi}^{(k)}$ *contain at most one element equal to* $1$*, whilst all the remaining elements are zero).*

All these technique are based on particular signal models, that we explicitly describe as well. At the end of this section, we detail how the presented recovery techniques can be implemented within the iterative monitoring framework of SCoRe1. In this manner they can be compared against to the proposed CS-PCA method. The performance results presented in Section 2.5.3, even if limited to the signals therein explained, will give insights on which of the analyzed technique is more suitable to be used with real signals (and therefore, which is the signal model among those considered that best describe the reality).

**Signal Models and Interpolation Techniques**

The *a priori* knowledge that we can have about the signal of interest $\mathbf{x}^{(k)}$ help us building a model for such signal. This knowledge can be deterministic, e.g., a description of the physical characteristics of the observed process, or probabilistic, e.g., the formulation of a probability distribution (called prior) to describe the possible realizations of $\mathbf{x}^{(k)}$ (or an equivalent representation of $\mathbf{x}^{(k)}$). In both cases, the acquired knowledge on the signal to recover can be obtained by observing or processing a set of representative realizations of the signals of interest (i.e., the set $\mathcal{X}_K^{(k)}$ or the training set $\widehat{\mathcal{T}}_K^{(k)}$). In summary, to compute $\widehat{\mathbf{x}}^{(k)}$ from $\mathbf{y}^{(k)}$, we need a model of $\mathbf{x}^{(k)}$ that can be built according to two different approaches: a *deterministic approach* or a *probabilistic approach*.

**Recovery Methods based on Deterministic Signal Models**

A possible way to think of $\mathbf{x}^{(k)} \in \mathbb{R}^N$ is as a signal whose elements depend on $d-$dimensional coordinates. To be more concrete, we can think of an environmental monitored signal collected from a WSN of $N$ nodes that we order freely (e.g., according to their IDs). Each node $i$, with $i = \{1, \ldots, N\}$, at time $k$ senses a value which is represented by element $x_i^{(k)}$ of vector $\mathbf{x}^{(k)}$. Since the considered node $i$ is deployed in a specific location of the network, it is also linked to a set of geographical coordinates (e.g., latitude and longitude, which can be represented with a $d = 2$ dimensional coordinate vector $\mathbf{c}^{(i)}$). $x_i^{(k)}$ represents the reading of the $i-$th network's node, which in turn is associated with a vector of $d$ coordinates $\mathbf{c}^{(i)}$, and therefore we can express $x_i^{(k)}$ as a function of $\mathbf{c}^{(i)}$, i.e., $x_i^{(k)}(\mathbf{c}^{(i)})$. A straightforward way to model $\mathbf{x}^{(k)}$ is by defining a proper function of the $d-$dimensional coordinate $\mathbf{c}$, $\phi(\mathbf{c})$ (e.g., the Green function) that satisfies regularity conditions (e.g., smoothness) inferred by "typical" realizations of the signal of interest $\mathbf{x}^{(k)}$ [73]. Thus, we can write each element $i$ of $\mathbf{x}^{(k)}$ as

$$x_i^{(k)}(\mathbf{c}^{(i)}) \simeq \sum_{j=1}^{L} \alpha_j \phi(\mathbf{c}^{(i)} - \mathbf{c}^{(j)}) \,, \tag{2.58}$$

where the function $\phi(\cdot)$ is used as a building block for $\mathbf{x}^{(k)}$ and $\alpha_j$ is the weight associated to $\phi(\cdot)$ centered in $\mathbf{c}^{(j)}$, with $j = 1, \ldots, L$, that is the $d-$dimensional coordinate corresponding to the physical placement of the node from which we received the $j-$th measurement.

The Biharmonic Spline Interpolation (**Spline**) [73] method solves Problem 7 exploiting the deterministic model in (2.58); the objective is to find a biharmonic function that passes through $L$ data points stored in the $L-$dimensional vector $\mathbf{y}^{(k)}$. In this context, the elements of both $\mathbf{y}^{(k)}$ and $\mathbf{x}^{(k)}$ are seen as a function of $d$ coordinates. Namely, to each element $j$ of the $L-$dimensional vector $\mathbf{y}^{(k)}$ is associated a $d-$dimensional index $\mathbf{c}^{(j)} = [c_1^{(j)}, \ldots, c_d^{(j)}]^T$. Similarly, to each element $i$ of the $N-$dimensional vector $\mathbf{x}^{(k)}$ is associated the $d-$dimensional index $\mathbf{c}^{(i)}$. In order to interpolate the $L$ points in $\mathbf{y}^{(k)}$ we require to satisfy for each element of $\mathbf{x}^{(k)}$ the smoothness condition[16] $\nabla^4 \widehat{x}^{(k)}(\mathbf{c}) = \sum_{j=1}^{L} \alpha_j \delta(\mathbf{c} - \mathbf{c}^{(j)})$ , given that, if $\mathbf{c} = \mathbf{c}^{(j)}$ then $\widehat{x}^{(k)}(\mathbf{c}^{(j)}) = y_j^{(k)}$, where $\mathbf{c}^{(j)}$ is the coordinate vector $\mathbf{c}^{(j)} = [c_1^{(j)}, \ldots, c_d^{(j)}]^T \in \mathbb{R}^d$ related to the reading $y_j^{(k)}$ (e.g., the geographical location of the reading $y_j^{(k)}$). The solution is proved

[16]Here, $\nabla^4$ is the biharmonic operator which allows to formalize regularity conditions on the fourth-order derivatives; $\delta(\cdot)$ is defined as $\delta(x) = 1$ if $x = 0$, $\delta(x) = 0$ otherwise.

to be:

$$\widehat{x}^{(k)}(\mathbf{c}) = \sum_{j=1}^{L} \alpha_j \phi_d(\mathbf{c} - \mathbf{c}^{(j)}) \,, \tag{2.59}$$

where $\phi_d(\cdot)$ is the Green function for the $d-$dimensional problem[17]. The constants $\alpha_1, \ldots, \alpha_L$ are found by solving the linear system $y_i^{(k)} = \sum_{l=1}^{L} \alpha_l \phi_d(\mathbf{c}^{(j)} - \mathbf{c}^{(l)})$, $\forall\, j \in \{1, \ldots, L\}$. To conclude, the solution $\widehat{\mathbf{x}}^{(k)} \in \mathbb{R}^N$ is the vector whose element $i$ is equal to $\widehat{x}^{(k)}(\mathbf{c}^{(i)})$, namely, the recovered value associated with the $d-$dimensional index $\mathbf{c}^{(i)}$.

An alternative way to determine a model for $\mathbf{x}^{(k)}$ allows us to abstract from the knowledge of where the signal sources are placed. Further, this second method is adaptable to the spatio-temporal correlation and structure of the signal. Observing that generally a physical phenomenon is correlated in time and that its spatial correlation can be considered as stationary over a given time period (e.g., from $k - K$ until $k$), a natural way to proceed is by assuming that $\mathbf{x}^{(k)}$ lies in the vector space spanned by the $K$ previous samples contained in $\mathcal{X}_K^{(k)}$ (or in $\widehat{\mathcal{T}}_K^{(k)}$ as seen in Section 2.5.1), i.e., in $\mathrm{span}\left\langle \mathcal{X}_K^{(k)} \right\rangle$. According to the formalism introduced in Section 2.4, let us refer to the temporal mean and covariance matrix of the elements in $\mathcal{X}_K^{(k)}$ as $\overline{\mathbf{x}}^{(k)}$ and $\widehat{\mathbf{\Sigma}}^{(k)}$, respectively. Let us consider also the ordered set $\mathcal{U}^{(k)} = \{\mathbf{u}_1^{(k)}, \ldots, \mathbf{u}_N^{(k)}\}$ of unitary eigenvectors of $\widehat{\mathbf{\Sigma}}^{(k)}$, placed according to the decreasing order of the corresponding eigenvalues. Let $\mathbf{U}_M^{(k)}$ be the $[N \times M]$ matrix whose columns are the first $M$ elements of $\mathcal{U}^{(k)}$. To build a model of $\mathbf{x}^{(k)}$ based on the assumption that this one lies in $\mathrm{span}\left\langle \mathcal{X}_K^{(k)} \right\rangle$, we can write:

$$\mathbf{x}^{(k)} \simeq \overline{\mathbf{x}}^{(k)} + \mathbf{V}^{(k)}\mathbf{s}^{(k)} = \overline{\mathbf{x}}^{(k)} + \mathbf{U}_M^{(k)}\mathbf{s}^{(k)} \,, \tag{2.60}$$

where, in general, $\mathbf{V}^{(k)}$ can be a whatsoever $[N \times M]$ matrix of orthonormal columns (obtained at time $k$ from the set $\{\mathbf{x}^{(k-K)} - \overline{\mathbf{x}}^{(k)}, \ldots, \mathbf{x}^{(k-1)} - \overline{\mathbf{x}}^{(k)}\}$, e.g., through the Gram-Schmidt process [74]), with $M \leq N$; here we set $\mathbf{V}^{(k)} = \mathbf{U}_M^{(k)}$ because given $M \leq N$, the best way to represent with $M$ components each element out of a set of $N-$dimensional elements is through PCA. In fact, from a geometric point of view, we can consider each sample $\mathbf{x}^{(k)}$, for all $k$, as a point in $\mathbb{R}^N$ and look as follows for the $M$-dimensional plane (with $M \leq N$) which provides the best fit to all the elements in $\mathcal{X}_K^{(k)}$, and therefore for all the vectors that lie in $\mathrm{span}\left\langle \mathcal{X}_K^{(k)} \right\rangle$, in terms of minimum Euclidean distance. The key point of PCA, is the Ky Fan Theorem [75].

[17] E.g., $\phi_1(\mathbf{c}) = |\mathbf{c}|^3$, $\phi_2(\mathbf{c}) = |\mathbf{c}|^2(\ln|\mathbf{c}| - 1)$ and $\phi_3(\mathbf{c}) = |\mathbf{c}|$.

**Theorem 2** (Ky Fan Theorem). *Let $\mathbf{\Sigma} \in \mathbb{R}^{N \times N}$ be a symmetric matrix, let $\lambda_1 \geq \cdots \geq \lambda_N$ be its eigenvalues and $\mathbf{u}_1, \ldots, \mathbf{u}_N$ the corresponding eigenvectors (which are assumed to be orthonormal, without loss of generality). Given $M$ orthonormal vectors $\mathbf{b}_1, \ldots, \mathbf{b}_M$ in $\mathbb{R}^N$, with $M \leq N$, it holds that*

$$\max_{\mathbf{b}_1,\ldots,\mathbf{b}_M} \sum_{j=1}^{M} \mathbf{b}_j^T \mathbf{\Sigma} \mathbf{b}_j = \sum_{j=1}^{M} \lambda_i \ , \tag{2.61}$$

*and the maximum is attained for $\mathbf{b}_i = \mathbf{u}_i$, $\forall i$.*

According to the Ky Fan Theorem, maximizing $\sum_{j=1}^{M} \mathbf{b}_j^T \widehat{\mathbf{\Sigma}}^{(k)} \mathbf{b}_j$ corresponds to finding the linear transformation $\mathcal{F}: \mathbb{R}^N \rightarrow \mathbb{R}^M$ that maximally preserves the information contained in the training set $\mathcal{X}_K^{(k)}$. In other words, this corresponds to maximize the variance of the $M$-dimensional (linear) approximation of each element in $\operatorname{span}\left\langle \mathcal{X}_K^{(k)} \right\rangle$ that, in turn, is strictly related to the information content of each signal in $\mathcal{X}_K^{(k)}$. Because of Theorem 2, the best $M$-dimensional approximation of any signal $\mathbf{x} \in \operatorname{span}\left\langle \mathcal{X}_K^{(k)} \right\rangle$ is given by [61]

$$\widehat{\mathbf{x}} = \overline{\mathbf{x}}^{(k)} + \mathbf{U}_M^{(k)} \left(\mathbf{U}_M^{(k)}\right)^T (\mathbf{x} - \overline{\mathbf{x}}^{(k)}) \ ,$$

where $\left(\mathbf{U}_M^{(k)}\right)^T (\mathbf{x} - \overline{\mathbf{x}}^{(k)})$ is the projection of $\mathbf{x} - \overline{\mathbf{x}}^{(k)}$ onto its best fitting $M$-dimensional plane. In summary, if the original point of interest $\mathbf{x}^{(k)} \in \operatorname{span}\left\langle \mathcal{X}_K^{(k)} \right\rangle$, we can transform it into a point $\mathbf{s}^{(k)} \in \mathbb{R}^M$ as follows:

$$\mathbf{s}^{(k)} \stackrel{def}{=} \left(\mathbf{U}_M^{(k)}\right)^T (\mathbf{x}^{(k)} - \overline{\mathbf{x}}^{(k)}) \ . \tag{2.62}$$

Multiplication of (2.62) by $\mathbf{U}_M^{(k)}$ and summation with the sample mean return the best approximation of the original vector, in accordance to (2.60).

To solve Problem 7 exploiting the model in Equation (2.60) we can simply use the Ordinary Least Square (OLS) method [72], thus we refer to this recovery solution as Deterministic Ordinary Least Square (**DOLS**). From $\mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)} \mathbf{x}^{(k)}$ and the assumption that Equation (2.60) holds, we can write

$$\mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)} (\overline{\mathbf{x}}^{(k)} + \mathbf{U}_M^{(k)} \mathbf{s}^{(k)}) \ . \tag{2.63}$$

The ordinary least square solution of Equation (2.63) is given by

$$\widehat{\mathbf{s}}^{(k)} = (\mathbf{\Phi}^{(k)} \mathbf{U}_M^{(k)})^{\dagger} (\mathbf{y}^{(k)} - \mathbf{\Phi}^{(k)} \overline{\mathbf{x}}^{(k)}) \tag{2.64}$$

and it allows us to estimate the signal $\mathbf{x}^{(k)}$ as $\widehat{\mathbf{x}}^{(k)} = \overline{\mathbf{x}}^{(k)} + \mathbf{U}_M^{(k)} \widehat{\mathbf{s}}^{(k)}$. In the above expression the symbol † indicates the Moore-Penrose pseudo-inverse matrix.

Recalling that in Equation (2.63) $\mathbf{y}^{(k)}$ is an $[L \times 1]$ vector whilst $\mathbf{s}^{(k)}$ is an $[M \times 1]$ with $M \leq L$, the systems (2.63) is in general overdetermined and may have no solutions (e.g., when all the $L$ measurements are linearly independent). In this case (2.64) minimizes $\|(\mathbf{y}^{(k)} - \mathbf{\Phi}^{(k)}\overline{\mathbf{x}}^{(k)}) - \mathbf{\Phi}^{(k)}\mathbf{U}_M^{(k)}\mathbf{s}^{(k)}\|_{\ell_2}$, obtaining $\widehat{\mathbf{s}}^{(k)}$ as the nearest (according to the Euclidean norm) possible vector to all the $L$ collected measurements. If $L = M$, instead, the Moore-Penrose pseudo-inverse coincides with the inverse matrix and $\widehat{\mathbf{s}}^{(k)}$ is uniquely determined.

**Recovery Methods based on Probabilistic Signal Models**

This alternative approach allows us to introduce an uncertainty in the model of $\mathbf{x}^{(k)}$ thus improving its effectiveness and robustness when exploited for interpolation, see Section 2.5.3.

Considering Equation (2.60), this can be reformulated as:

$$\mathbf{x}^{(k)} \simeq \overline{\mathbf{x}}^{(k)} + \mathbf{V}^{(k)}\mathbf{s}^{(k)} = \overline{\mathbf{x}}^{(k)} + \mathbf{U}^{(k)}\mathbf{s}^{(k)} \,, \tag{2.65}$$

where $\mathbf{V}^{(k)}$ is now an $[N \times N]$ matrix of orthonormal columns set equal to the PCA matrix $\mathbf{U}^{(k)}$ following the same rationale than above. Here, the cardinality of the model's parameters is $N$ (i.e., the dimension of vector $\mathbf{s}^{(k)}$), which is surely larger than or equal to the dimension of $\text{span}\left\langle \mathcal{X}_K^{(k)} \right\rangle$. The model in (2.65), therefore, allows us to account for the fact that $\mathbf{x}^{(k)}$ could not perfectly lie in $\text{span}\left\langle \mathcal{X}_K^{(k)} \right\rangle$. This kind of approach has been implicitly adopted also in Section 2.4.2 and, recalling Figure 2.13, we can see that we need further assumptions on the system input $\mathbf{s}^{(k)}$ to fully characterize the model in Equation (2.65), i.e., we have to assign a prior to $\mathbf{s}^{(k)}$. Practically, as already seen in Section 2.4, $\mathbf{s}^{(k)}$ is a vector random process that we can assume to be, e.g., a Gaussian multivariate process[18] or a Laplacian vector process with i.i.d. components.

When we assign a Laplacian prior to $\mathbf{s}^{(k)}$, we can solve Problem 7 through our proposed recovery CS-PCA that corresponds to minimize $\|\mathbf{s}^{(k)}\|_{\ell_1}$, given that $\mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{\Psi}\mathbf{s}^{(k)}$, as shown in Section 2.4.5.

Differently, when we assign a Gaussian prior to $\mathbf{s}^{(k)}$, we can solve Problem 7 again via the Ordinary Least Square Method; we refer to this recovery method as Probabilistic Ordinary Least Square Method (**POLS**). In this case, we just have to rewrite Equation (2.64) as

$$\widehat{\mathbf{s}}^{(k)} = (\mathbf{\Phi}^{(k)}\mathbf{U}^{(k)})^{\dagger}(\mathbf{y}^{(k)} - \mathbf{\Phi}^{(k)}\overline{\mathbf{x}}^{(k)}) \,. \tag{2.66}$$

[18]This is the standard way of dealing with such problems, which appeals to the Central Limit Theorem of probability theory [76].

In this equation, the dimension of $\mathbf{y}^{(k)}$, $L$, is less then the dimension of $\mathbf{s}^{(k)}$, which is $N$. Therefore, Equation (2.66) is the solution of an ill-posed system, which theoretically allows an infinite number of solutions. Nevertheless, a multivariate Gaussian prior on $\mathbf{s}^{(k)}$ with zero mean and independent components[19], i.e., $p(\mathbf{s}^{(k)}) \sim \mathcal{N}(0, \mathbf{\Sigma_s})$ where $\mathbf{\Sigma_s}$ is a diagonal matrix, helps us to choose, among all the possible solutions, the one estimated as[20]

$$\begin{aligned}
\widehat{\mathbf{s}}^{(k)} &= \underset{\mathbf{s}^{(k)}}{\operatorname{argmax}}\, p(\mathbf{s}^{(k)}|\mathbf{y}^{(k)}) = \underset{\mathbf{s}^{(k)}}{\operatorname{argmax}}\, p(\mathbf{y}^{(k)}|\mathbf{s}^{(k)})p(\mathbf{s}^{(k)}) \\
&= \underset{\mathbf{s}^{(k)}}{\operatorname{argmax}}\, \delta(\mathbf{y}^{(k)}, \mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)}) \frac{1}{(2\pi)^{\frac{L}{2}} \det(\mathbf{\Sigma_s})^{\frac{L}{2}}} \exp\left\{ -\frac{\|\mathbf{\Sigma_s}\mathbf{s}^{(k)}\|_2^2}{2} \right\} \\
&= \underset{\mathbf{s}^{(k)}}{\operatorname{argmin}}\, \|\mathbf{\Sigma_s}\mathbf{s}^{(k)}\|_2^2, \;\; \text{given that } \mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)}
\end{aligned} \tag{2.67}$$

that corresponds to the solution in Equation (2.66), namely the minimum of $\|\mathbf{s}^{(k)}\|_{\ell_2}$ given that $\mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)}$.

**Implementation of Signal Recovery Methods**

Each of the interpolation techniques explained above can be implemented at the data collection point of our monitoring framework, specifically in the recovery block (see Section 2.5.1 and Figure 2.21). As previously remarked, at each time sample $k$, we can think of $\mathbf{x}^{(k)}$ as an $N-$dimensional signal whose elements depend on coordinates in $d$ dimensions. If we measure $\mathbf{x}^{(k)}$ in $L$ different coordinate points, collecting the measurement set $\{y_1^{(k)}, \ldots, y_L^{(k)}\}$, for the recovery stage we can proceed as follows:

1) Biharmonic Spline (**Spline**)

a) compute $\alpha_1, \ldots, \alpha_L$ solving $y_j^{(k)}(\mathbf{c}^{(j)}) = \sum_{l=1}^{L} \alpha_l \phi_d(\mathbf{c}^{(j)} - \mathbf{c}^{(l)}) \quad \forall\, j \in 1, \ldots, L$ ;

b) estimate $x_i^{(k)}$ as $\widehat{x}_i^{(k)}(\mathbf{c}^{(i)}) = \sum_{j=1}^{L} \alpha_j \phi_d(\mathbf{c}^{(i)} - \mathbf{c}^{(j)}) \quad \forall\, i \in 1, \ldots, N$ .

Alternatively, if we assume to know the $K$ previous samples $\mathcal{X}_K^{(k)} = \{\mathbf{x}^{(k-K)}, \ldots, \mathbf{x}^{(k-1)}\}$ or the training set $\widehat{\mathcal{T}}_K^{(k)} = \{\widehat{\mathbf{x}}^{(k-K)}, \ldots, \widehat{\mathbf{x}}^{(k-1)}\}$, with $K \leq N$, we can abstract from the knowledge of the physical coordinates associated to $\mathbf{x}^{(k)}$. In this case we need to compute the PCA matrix $\mathbf{U}^{(k)}$ from $\mathcal{X}_K^{(k)}$ (or $\widehat{\mathcal{T}}_K^{(k)}$) as explained in Section 2.4.1. Then, knowing both

[19] Note that $\mathbf{s}^{(k)}$ can be assumed to have independent components if obtained through (2.62). If $\mathbf{s}^{(k)}$ is the vector of principal components of $\mathcal{X}_K^{(k)}$, these are known to be uncorrelated and therefore, under the assumption of gaussianity and zero mean, they are also independent.

[20] We recall here that, in the formulas (2.67) $\delta(\cdot)$ is a function defined as: $\delta(\mathbf{x}, \mathbf{y}) = 1$ if $\mathbf{x} = \mathbf{y}$, $\delta(\mathbf{x}, \mathbf{y}) = 0$ otherwise.

$\mathbf{U}^{(k)}$ and $\mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{x}^{(k)}$, for the DOLS method we set $M = K - 1$ and at each time $k$ we can estimate $\mathbf{x}^{(k)}$ according to:

2) Deterministic Ordinary Least Square (**DOLS**)

a) estimate $\mathbf{s}^{(k)}$ as $\widehat{\mathbf{s}}^{(k)} = (\mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}_{K-1})^{\dagger}(\mathbf{y}^{(k)} - \mathbf{\Phi}^{(k)}\overline{\mathbf{x}}^{(k)})$ ;

b) estimate $\mathbf{x}^{(k)}$ as $\widehat{\mathbf{x}}^{(k)} = \overline{\mathbf{x}}^{(k)} + \mathbf{U}^{(k)}_{K-1}\widehat{\mathbf{s}}^{(k)}$ .

Concerning the remaining recovery methods, instead, they can be implemented as follows:

3) Probabilistic Ordinary Least Square (**POLS**)

a) estimate $\mathbf{s}^{(k)}$ as $\widehat{\mathbf{s}}^{(k)} = (\mathbf{\Phi}^{(k)}\mathbf{U}^{(k)})^{\dagger}(\mathbf{y}^{(k)} - \mathbf{\Phi}^{(k)}\overline{\mathbf{x}}^{(k)})$ ;

b) estimate $\mathbf{x}^{(k)}$ as $\widehat{\mathbf{x}}^{(k)} = \overline{\mathbf{x}}^{(k)} + \mathbf{U}^{(k)}\widehat{\mathbf{s}}^{(k)}$ .

4) Joint CS and PCA (**CS-PCA**)

a) estimate $\mathbf{s}^{(k)}$ as $\widehat{\mathbf{s}}^{(k)} = \text{argmin}_{\mathbf{s}^{(k)}} \|\mathbf{s}^{(k)}\|_{\ell_1}$, given that $\mathbf{y}^{(k)} = \mathbf{\Phi}^{(k)}\mathbf{U}^{(k)}\mathbf{s}^{(k)}$ ;

b) estimate $\mathbf{x}^{(k)}$ as $\widehat{\mathbf{x}}^{(k)} = \overline{\mathbf{x}}^{(k)} + \mathbf{U}^{(k)}\widehat{\mathbf{s}}^{(k)}$ .

The performance of these four different reconstruction techniques is compared in the next section.

### 2.5.3 Performance Analysis

In this section we analyze the performance of the proposed monitoring framework when used in conjunction with the signal recovery methods of Section 2.5.2. First, we analyze the statistics of all the signals gathered from the WSN deployments W1-W5 described in Section 2.4.3 and we choose a relevant subset of them to perform our performance analysis[21]. Successively, we investigate the performance of the signal recovery methods.

**Signals:** We considered five different WSNs, each one of them sensing different types of signals for a total of 24 signals, see Table 2.2. For each signal $\mathbf{x}^{(k)} \in \mathbb{R}^N$, we calculate the average inter-node correlation $\rho_s(\mathbf{x}^{(k)})$, defined as the average correlation between the one

[21] We recall here that the proposed framework is flexible and does not depend on a specific network topology. Its only requirement is that the sensor nodes must be ordered according to some criterion, e.g., using their IDs. For this reason, it is expected that signals with similar statistical characteristics have similar performances.

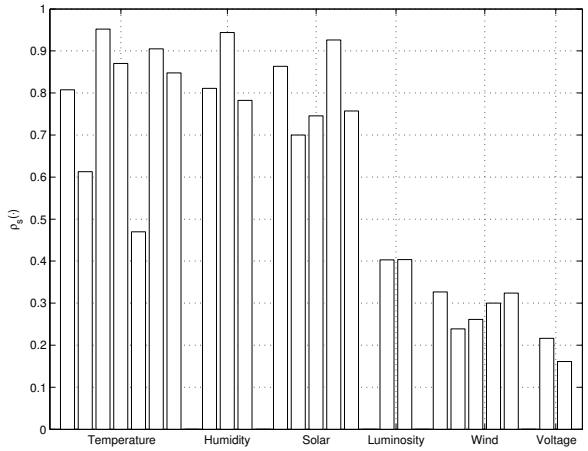



**Figure 2.22.** *Inter-node correlation for different signals gathered from the 5 different WSNs considered.*

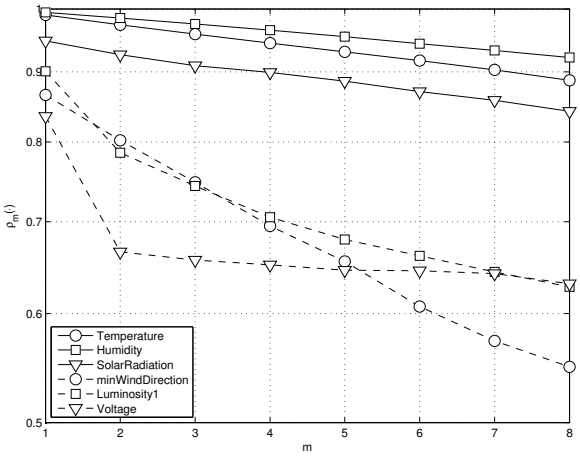



**Figure 2.23.** *Intra-node correlation for the signals chosen among all the signals considered in Figure 2.22.*

dimensional signal sensed by node $i$, $x_i(k)$, and the one sensed by node $j$, $x_j(k)$, for all the node pairs $i, j$:

$$\rho_s(\mathbf{x}^{(k)}) = \sum_{k=1}^{K} \frac{1}{K} \sum_{i=1}^{N} \sum_{j>i} \frac{\left(x_i^{(k)} - E[x_i]\right)\left(x_j^{(k)} - E[x_j]\right)}{((N^2-N)/2)\sigma_{x_i}\sigma_{x_j}} \,. \tag{2.68}$$

$\rho_s(\mathbf{x}^{(k)})$ gives us a measure of the expected sparsity of the principal components $\mathbf{s}^{(k)} \in \mathbb{R}^N$. If we calculate the principal components of a signal with maximum inter-node correlation, i.e., $\rho_s(\mathbf{x}^{(k)}) = 1$, we will obtain a signal $\mathbf{s}^{(k)}$ with only the first component different from zero. Conversely, if we calculate the principal components of a signal with minimum inter-node correlation $\rho_s = 0$, we will obtain a signal $\mathbf{s}^{(k)}$ with no negligible components (as respect to the overall energy of the signal).

In Figure 2.22 we depict the inter-nodes correlation for all the signals considered and we divide them according to the signal type, i.e, Temperature, Humidity, Solar Radiation, Luminosity, Wind and Voltage. We notice that the signals Temperature, Humidity and Solar Radiation have in average a high inter-node correlation ($\rho_s(\mathbf{x}^{(k)}) \simeq 0.7$), while indoor Luminosity, Wind Direction and Voltage have a lower inter node correlation ($\rho_s(\mathbf{x}^{(k)}) \simeq 0.25$). To further analyze these signals, we consider the intra-node correlation $\rho_m(\mathbf{x}^{(k)})$, that is the correlation of the one dimensional signal $x_i^{(k)}$ sensed by a single node with the same signal shifted by $m$ time samples, i.e., $x_i^{(k+m)}$, averaged for all the $N$ signals of $\mathbf{x}^{(k)} \in \mathbb{R}^N$. It is

defined as:

$$\rho_m(\mathbf{x}^{(k)}) = \sum_{i=1}^{N} \frac{1}{N} \frac{\sum_{k=1}^{K} \left(x_i^{(k)} - E[x_i]\right)\left(x_i^{(k+m)} - E[x_i]\right)}{K\sigma_{x_i}^2} \,. \tag{2.69}$$

For representation purposes, we choose one signal for each type, within the 24 signals depicted in Figure 2.22, and we represent for each chosen signal the temporal correlation $\rho_m(\mathbf{x}^{(k)})$, for $m = 1, \ldots, 8$ in Figure 2.23. We notice that Temperature, Humidity and Solar Radiation signals keep a high intra-node correlation even for $m = 8$ ($\rho_8(\mathbf{x}^{(k)}) \geq 0.85$), while for Luminosity and Wind signals the temporal correlation quickly decreases ($\rho_8(\mathbf{x}^{(k)}) \leq 0.65$). The Voltage signal has different characteristics, since, even though it has inter-node and intra-node correlation similar to Luminosity and Wind Direction, it is a nearly constant signal.

For our results, we used the signals gathered from the WSN testbed deployed on the ground floor of the Department of Information Engineering at the University of Padova [77] using $N = 68$ TmoteSky wireless nodes equipped with IEEE 802.15.4 compliant radio transceivers. We have chosen these signals because 1) they are representative of the whole signal set considered, and 2) we have full control on the WSN from which they have been gathered, so that we could easily collect meaningful traces for the performance evaluation of our proposed scheme. In particular, we consider 5 signals divided into classes accordingly to their statistical characteristics:

**C1)** two signals with high temporal and spatial correlation, i.e., the ambient temperature [°C] and the ambient humidity [%];

**C2)** two signals with lower correlation, i.e., the photo sensitivity [A/W] in the range $320 - 730$ nm and in the range $320 - 1100$ nm;

**C3)** the battery level [V] of the sensor nodes during the signal collection campaign.

Over time, each signal has been collected every 5 minutes. The results have been obtained from 100 independent simulation runs and by averaging the performance over all signals in each class.

**Performance of the Signal Recovery Methods:** In the following, we show performance curves for the different recovery techniques illustrated in Section 2.5.2: Biharmonic Spline (Spline), Deterministic Ordinary Least Square (DOLS), Probabilistic Ordinary Least Square

(POLS) and Joint CS and PCA (CS-PCA). Note that DOLS cannot be considered as an effective solution since it is affected by a numerical stability problem. Nevertheless, we considered it in view of its simplicity and low complexity.

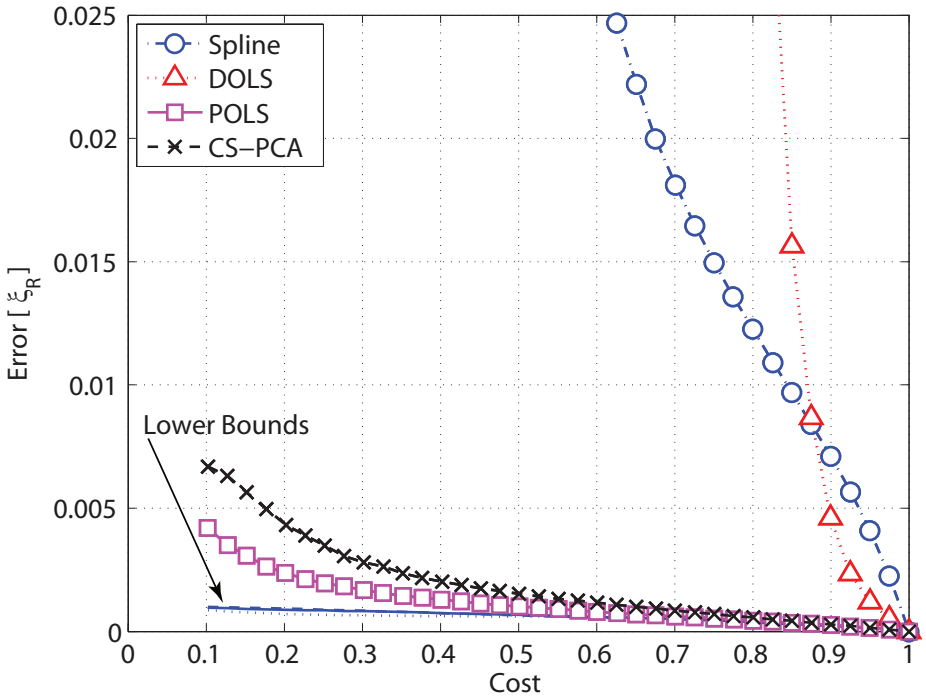



**Figure 2.24.** *Performance comparison of different recovery techniques within our iterative monitoring scheme, for signals in class C1, temperature and humidity.*

Along the x-axis of the figures presented in this section we have the normalized cost expressed as the average fraction of packet transmissions in the network per time sample, formally:

$$\text{Cost} = \frac{1}{T d_{\text{TOT}}} \sum_{k=1}^{T} \sum_{n=1}^{N} d_n \mathcal{I}^{(k)}(n) \,, \tag{2.70}$$

where $T$ is the number of considered time instants (i.e., the overall duration of the data collection), $N$ is the total number of nodes in the WSN, $d_n$ is the distance in terms of number of hops from node $n$ to the DCP, $d_{\text{TOT}} = \sum_{n=1}^{N} d_n$ and $\mathcal{I}^{(k)}(n)$ is an indicator function, with $\mathcal{I}^{(k)}(n) = 1$ if node $n$ transmits and $\mathcal{I}^{(k)}(n) = 0$ if node $n$ remains silent at time $k$. Note that a normalized cost equal to 1 corresponds to the case where all nodes transmit during all time instants $1, 2, \ldots, T$, which accounts for the maximum energy consumption for the network. Conversely, the normalized cost is zero when all nodes remain silent during all time instants. The y-axis shows the mean signal reconstruction error at the end of the recovery process,

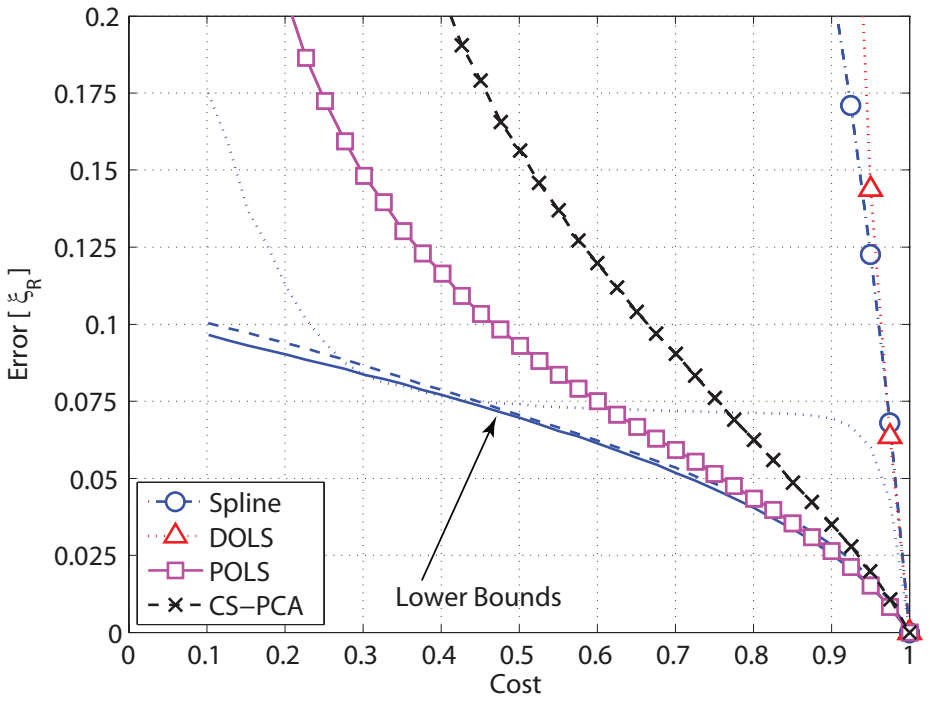



**Figure 2.25.** *Performance comparison of different recovery techniques within our iterative monitoring scheme, for signals in class C2, photo sensitivity in the range* $320 - 730$ *nm and in the range* $320 - 1100$ *nm.*

calculated accordingly to Equation (2.54), i.e.,

$$\overline{\xi}_R = \frac{1}{K}\sum_{k=1}^{K} \xi_R^{(k)} \quad \text{where} \quad \xi_R^{(k)} = \frac{\|\mathbf{x}^{(k)} - \widehat{\mathbf{x}}^{(k)}\|_2}{\|\mathbf{x}^{(k)}\|_2} \,.$$

In order to vary the metric Cost (x-axis) we modify the parameters of SCoRe1 as explained in the Appendix 2.E. Solid, dashed and dotted lines without marks represent lower bounds on the error recovery performance, which are obtained as exploiting a perfect knowledge of the past in the training set, i.e., considering $\mathcal{X}_K^{(k)}$ instead of $\widehat{\mathcal{T}}_K$.

In Figures 2.24 and 2.25 we can see that an imperfect knowledge of the training set severely impacts the recovery performance of DOLS. This is however not as dramatic for CS-PCA and POLS. It is also interesting to note that using $\widehat{\mathcal{T}}_K^{(k)}$ POLS outperforms CS-PCA, whilst with $\mathcal{X}_K^{(k)}$ CS-PCA and POLS perform equally good, also for the highly variable signals of class C2, see Figure 2.25. In fact, the introduction of a further error in the model, i.e., an uncertainty on the training set, makes the Gaussian prior for $\mathbf{s}^{(k)}$ more effective than the Laplacian one, in accordance to the Central Limit Theorem (e.g., see [76]). Nevertheless, both POLS and CS-PCA remain valid solutions for a monitoring application framework,

since the performance loss from the ideal case, which assumes perfect knowledge of $\mathcal{X}_K^{(k)}$, to the one that exploits $\widehat{\mathcal{T}}_K^{(k)}$ is sufficiently small. Concerning spline, this method allows to reach good performance only above a transmission probability of 0.8; furthermore, the use of Spline as recovery technique within the same framework of SCoRe1, instead of CS-PCA, leads to huge errors due to: (i) the tendency of our protocol to systematically avoid transmissions when possible; (ii) the approximation of the error estimate; (iii) the variability of the signal and (iv) the fact that Spline does not exploit any previous knowledge on the statistics of the signal to recover.

The above results provide evidence that SCoRe1 is an effective solution for monitoring applications for WSNs in different scenarios. Equally important, the achieved performance shows that CS recovery can be effectively used for networking and this is achieved thanks to our approach which, differently from the literature, interprets CS as an interpolation technique, besides as a method to jointly perform data acquisition and compression.

## 2.6 Conclusions and Discussions

In this chapter we presented our research activity focused on the study of joint sampling, recovery and protocol adaptation for distributed signals monitored by a WSN. In particular, we studied the potential benefits and the applicability on networking of a novel signal processing technique called Compressive Sensing (CS).

First, we studied the behavior of CS when jointly used with a routing scheme for recovering two types of signals: synthetic ones and real sensor data. We showed that for the synthetic signals the reconstruction at the sink node is enhanced when applying CS, whereas the application of CS for real sensor data is not straightforward. Thus, as a next step we investigated the effectiveness of data recovery through joint CS and Principal Component Analysis (PCA) in Wireless Sensor Networks. At first, we framed our recovery scheme into the context of Bayesian theory proving that the principal components of different real world WSN signals are well modeled by a Laplacian distribution. This legitimates the use of CS in WSN environments and must be regarded as the first original contribution of this work.

Then, based on the above results, we proposed a novel technique for signal monitoring applications in WSN. This technique, called SCoRe1, is based both on PCA, to learn the data statistics, and CS, to recover the signal through convex optimization and a feedback controller to bound the error. Using data measured in different testbeds, we have shown that our technique is robust to unpredictable changes in the signal statistics and achieves good performance in terms of reconstruction accuracy *vs* network cost (i.e., number of transmissions required). Thanks to our approach, we showed that CS recovery can be adopted for networking when exploited as an interpolation technique besides as a method to jointly perform data acquisition and compression. This must be regarded as the second original contribution of our work.

Moreover, as further outcome of our research, the good simulation results achieved by SCoRe1 gave us the possibility of being actively involved in a new project, currently ongoing. The aim of this project is that of implementing the proposed SCoRe1 technique on real sensor nodes within a Client/Server architecture called WSN-Control , see Figure 2.26. Here, the WSN (possibly composed of multiple sensor islands) can be accessed through a number of WSN gateways. Sensor nodes adopt a protocol stack based on 6LoWPAN [78] and run a suitable routing protocol to send the gathered data to the gateways. For a more detailed description of the protocols running in the WSN the reader is referred to [79].

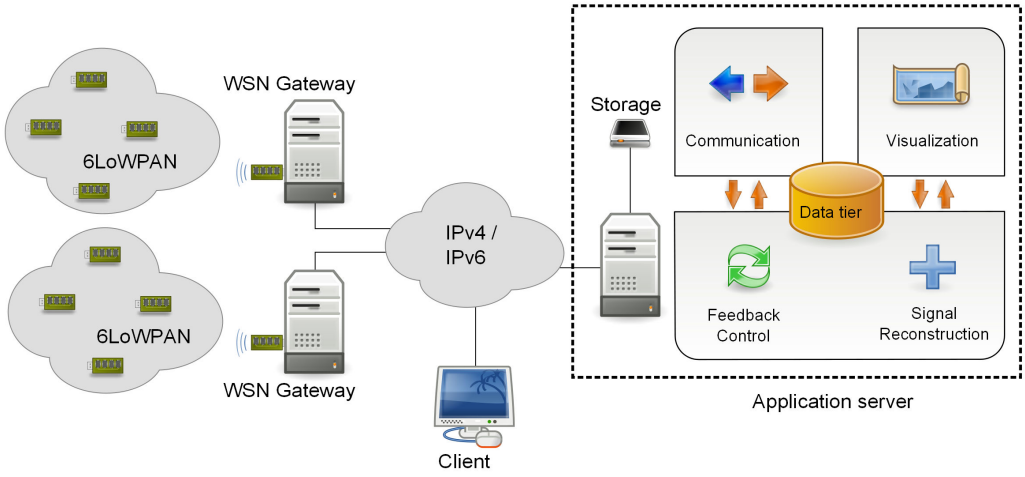



**Figure 2.26.** *WSN-Control architecture.*

Concerning SCoRe1, the core of the WSN-Control system is the Application Server (see Figure 2.26). In detail, this server is a Web application composed of the following blocks: 1) Visualization, which creates a 3D representation of the gathered data, and is also responsible for the user interface and for the related Applet and Java Server Page (JSP) technology [80]; 2) Communication, which is the block is responsible for the reception of data from the WSN and for the transmission of data gathering requests to the sensor nodes and 3) Signal Reconstruction and Feedback Control, where SCoRe1 reconstructs the entire WSN signal from the received measurements. In particular, our solution will allow us to minimize the number of nodes that send their measurements at each data collection round, while keeping the reconstruction error below a certain threshold. A more detailed discussion on the application of SCoRe1 in WSN-Control can be found in [26].

To conclude, we stress that even though our framework has been designed for WSN monitoring, it can be readily applied also to a wide range of applications and network infrastructures (e.g., cellular networks) that require the approximation of a large and distributed dataset, with a certain spatial or temporal correlation.

## 2.A Compressive Sensing in 2D

In this appendix, we review a known method from image processing to generalize the CS theory in Section 2.1 to 2D signals, as those considered in Section 2.3. Accordingly, the input signal is a $K \times K$ square matrix $\mathbf{X}$ with $N = K^2$ elements. Element $(i, j)$ of this matrix, $x(i, j)$, is the value sampled by the sensor placed in cell $(i, j)$ of the sensor grid. We assume that the 2D signal $\mathbf{X}$ is sparse under a given transformation. Thus, $\mathbf{X}$ can be written as

$$\mathbf{X} = \mathbf{LSR} \, , \tag{2.71}$$

where $\mathbf{L}$ and $\mathbf{R}$ are two non singular matrices and $\mathbf{S}$ is a $K \times K$ matrix representing the sparse signal in the transformation domain. As an example, the DCT of $\mathbf{X}$ in two dimensions is calculated as

$$\mathbf{S} = \mathbf{\Psi}^T \mathbf{X} \mathbf{\Psi} \, , \tag{2.72}$$

where $\mathbf{\Psi}^T$ indicates the transpose of $\mathbf{\Psi}$, and $\mathbf{\Psi}$ is the transformation matrix. The generic element $(i, j)$ of matrix $\mathbf{\Psi}$ is given as

$$\psi(i, j) = \omega_j \cos\left(\frac{\pi(2i-1)(j-1)}{2K}\right) , \tag{2.73}$$

and $\omega_j$ is defined as

$$w_j = \begin{cases} \frac{1}{\sqrt{K}} & j = 1 \\ \sqrt{\frac{2}{K}} & 2 < j \leq K \end{cases} . \tag{2.74}$$

In what follows, we use tools from linear algebra to reformulate the 2D problem as an equivalent 1D problem. It is worth noting that this transformation does not lose any information and preserves the correlation among sensed values in the 2D space.

Now we define a $\texttt{vec}(\cdot)$ function, transforming a $K \times K$ matrix into a vector of length $N$ (through a reordering of the matrix elements) as shown in (2.30).

As explained in Section 2.3, the values that we collect at the sink can be represented through a vector $\mathbf{y}$ of $M < N$ elements. They are linear combinations of the sensor readings represented by the matrix $\mathbf{X}$ of size $K \times K$, and thus $\mathbf{y} = \mathbf{\Phi}\texttt{vec}(\mathbf{X})$. The $M \times N$ matrix $\mathbf{\Phi}$ contains the combination coefficients that are picked at random according to a given distribution. From linear algebra we know that the vector form of a given product among three matrices $\mathbf{L}$, $\mathbf{R}$ and $\mathbf{C}$ can be rewritten as [81]

$$\texttt{vec}(\mathbf{LCR}) = (\mathbf{L}^T \otimes \mathbf{R})\texttt{vec}(\mathbf{C}) \, , \tag{2.75}$$

where $\otimes$ is the Kronecker product. Hence, using (2.71) and (2.75) we can write $\texttt{vec}(\mathbf{X}) = (\mathbf{L}^T \otimes \mathbf{R})\texttt{vec}(\mathbf{S})$. Using $\mathbf{y} = \mathbf{\Phi}\texttt{vec}(\mathbf{X})$ we obtain $\mathbf{y} = \mathbf{\Phi}(\mathbf{L}^T \otimes \mathbf{R})\texttt{vec}(\mathbf{S})$ that, defining $\mathbf{A} = \mathbf{\Phi}(\mathbf{L}^T \otimes \mathbf{R})$, can be rewritten as

$$\mathbf{y} = \mathbf{A}\texttt{vec}(\mathbf{S}) \,, \tag{2.76}$$

where $\mathbf{y}$ is the vector containing the received (combined) values and $\texttt{vec}(\mathbf{S})$ is a column vector of length $N$ containing the input signal in the transformation domain. Given (2.76) we can recover the sparse signal $\texttt{vec}(\mathbf{S})$ using the solvers developed for standard CS theory in 1D.

## 2.B CS Recovery Capability with Respect to Sparseness

In this appendix we report more analysis results obtained when Compressive Sensing is applied to synthetic signals that mimic realistic WSN signals. In detail, here we investigate the recovery capabilities of CS as a function of the sparseness of the 2D signals that are collected from a sensor grid. To isolate the impact of the measurement matrix $\mathbf{\Phi}$ and the structure of the input signals, we first study some extreme cases for $\mathbf{\Phi}$ together with band-pass input signals. The impact of routing and topology, which largely characterize $\mathbf{\Phi}$, is investigated in the Appendix 2.C.

**Signal of interest X:** Here, the matrix $\mathbf{X}$ is built starting from a sparse matrix $\mathbf{S}$, obtained through the following steps:

1. We build a preliminary signal $\mathbf{S}_1$ of size $K \times K$ having all frequencies (i.e., all entries in the matrix) with the same amplitude, i.e., $s_1(p,q) = 1, \forall\, p,q = 1,2,\ldots,K$.

2. We define a frequency mask as a 2D rectangular function that is one for entries in position $(p,q)$ with $p_{\text{low}} < p \leq p_{\text{high}}$ and $q_{\text{low}} < q \leq q_{\text{high}}$ and zero otherwise. This rectangular function is defined as

$$\texttt{rect}(p,q) \stackrel{def}{=} \begin{cases} 1 & \text{if } p_{\text{low}} < p \leq p_{\text{high}} \text{ and } q_{\text{low}} < q \leq q_{\text{high}} \\ 0 & \text{otherwise .} \end{cases} \tag{2.77}$$

3. We obtain a second signal $\mathbf{S}_2$ of size $K \times K$, whose entries $s_2(p,q)$ are calculated as

$$s_2(p,q) = s_1(p,q)\texttt{rect}(p,q) \ . \tag{2.78}$$

4. We finally obtain $\mathbf{S}$ as follows: if $s_2(p,q) = 0$ then $s(p,q) = 0$. If instead $s_2(p,q) = 1$, $s(p,q) = 0$ with probability $p_d$ and $s(p,q) = 1$ otherwise. The parameter $p_d$ represents the fraction of entries that are on average deleted from $\mathbf{S}_2$.

Therefore, the signal $\mathbf{S}$ is obtained by first applying a frequency mask, which helps to assess the reconstruction performance for low-frequency, mid-frequency, and high-frequency signals. In addition, we delete some randomly picked frequencies according to a given probability $p_d$. This is a simple but accurate method to control the characteristics of the signal in the DCT domain (i.e., the sparsity of the signal and its dominant frequency components and

allows us to understand the effects of the signal structure on the performance of CS).

**Matrix $\mathbf{\Phi}$:** first we observe that $\mathbf{\Phi}$ is an $M \times N$ matrix, where $M$ is the number of collected packets at the sink and $N$ the number of nodes in the sensor grid. We consider the following classes of matrices $\mathbf{\Phi}$

M1. The first type of matrix has elements picked at random and independently of each other. The generic element $(s,t)$ of this matrix $\varphi(s,t)$, with $1 \leq s \leq M$, $1 \leq t \leq N$, is set to $\varphi(s,t) = \texttt{rand}()$, where $\texttt{rand}()$ returns a random number uniformly picked in $(0,1]$. As proved in [16], this type of matrix used in conjunction with the DCT transformation gives very good recovery performance for Compressive Sensing.

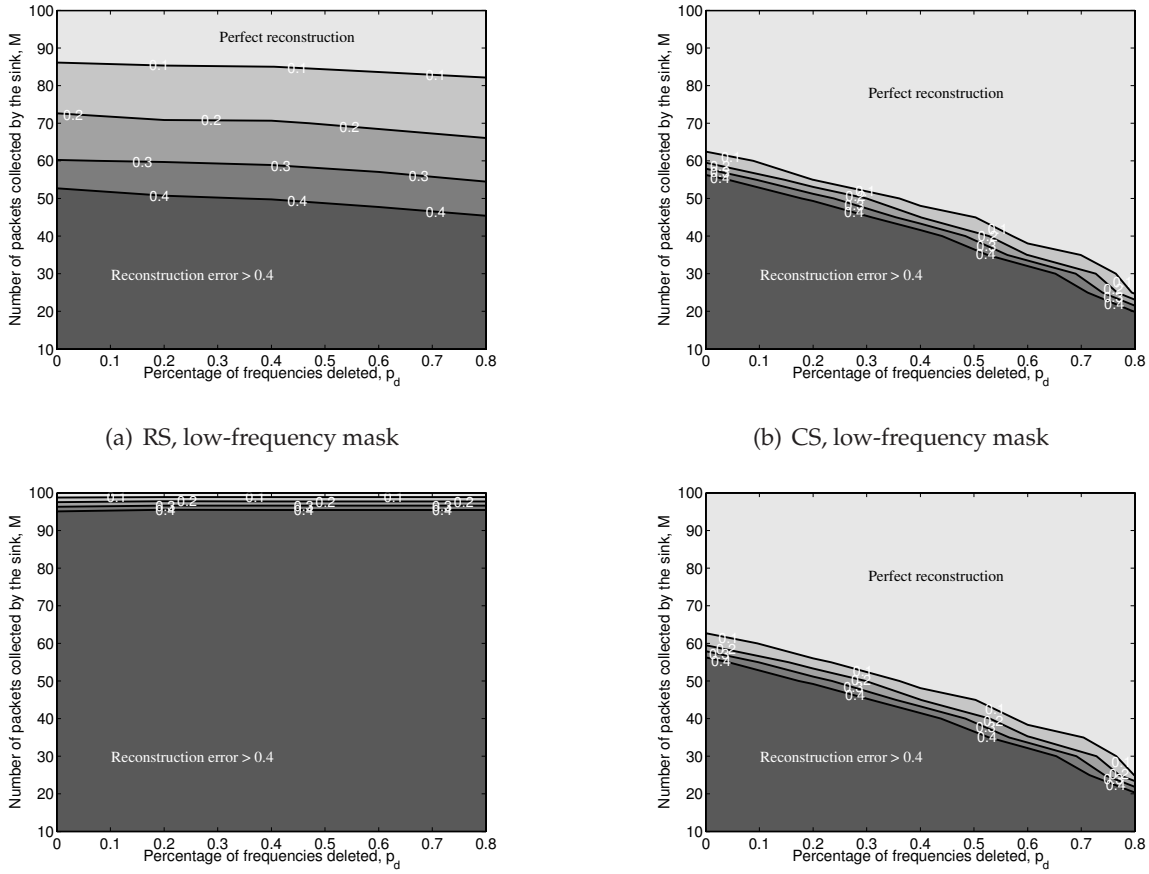
M2. We consider block matrices, where non-zero elements are grouped in blocks (sub-matrices) along the diagonal of the matrix $\mathbf{\Phi}$. This reflects networks where combinations of sensor readings occur within clusters of nodes rather than over all nodes in the network. In realistic scenarios, the information carried by a given packet depends on the specific path that this packet traverses from its first transmission to its delivery at the sink which, in turn, depends on the structure of the data gathering tree.

M3. As an extension we use a more random setting for the coding matrix $\mathbf{\Phi}$ than in M2. We set a certain fraction of matrix entries to 0 as in M2, but do this for randomly picked coefficients of the matrix. This reflects mixing opportunities for a sparse network with random node encounters, where mobile nodes exchange information whenever they meet, or for a sensor network with random unsynchronized sleep cycles, where data can be exchanged whenever two nodes happen to be awake at the same time.

**Reconstruction quality:** here we consider the quantity $\varepsilon$, computed as explained in Section 2.3.4, Equation (2.33).

**Performance evaluation:** in what follows we show the performance of CS for various input signals $\mathbf{X}$ (low-, mid-,and high-frequency) and matrices $\mathbf{\Phi}$. We further vary $M$, the number of packets gathered at the sink, and $p_d$. For comparison, we also show results for random sampling (RS) which works as follows: for a given $M$, we select uniformly at random $M$ out of the $N$ nodes in the network. These nodes send their own measurements to the sink. The

signal is reconstructed by interpolation of the $M$ collected values according to the method in [57]. For the following simulations, we use $N = 100$ and $K = \sqrt{N} = 10$.




(a) RS, low-frequency mask

(b) CS, low-frequency mask

(c) RS, high-frequency mask

(d) CS, high-frequency mask

**Figure 2.27.** *Reconstruction error for varying $p_d$ and number of received packets $M$.*

In Figures 2.27(a) and 2.27(b), we show the performance of RS and CS, respectively, considering low-frequency signals that were shaped according to the frequency mask $p_{low} = q_{low} = 0$ and $p_{high} = q_{high} = K/2$. In Figures 2.27(c) and 2.27(d), we show the same performance metrics for high-frequency signals with $p_{low} = q_{low} = K/2$ and $p_{high} = q_{high} = K$. In all graphs, the matrix $\mathbf{\Phi}$ used for CS is as specified in point M1 above. Contour levels are shown in all plots to represent the average reconstruction error $\varepsilon$. Exact reconstruction of the signal occurs for the region of the graph above $\varepsilon = 0.1$. While the impact of the reconstruction error depends on the specific application, from visual inspection we observed that the reconstructed signal already follows the original signal very well for $\varepsilon \leq 0.3$.

The performance of RS in the low-frequency case in Figure 2.27(a) is only marginally affected by $p_d$, i.e., the sparseness of the input signal does not influence the behavior of the RS

scheme. Also, for high-frequency signals (Figure 2.27(c)) the performance of RS is unacceptable as full reconstruction is possible only when $M \approx N$, as expected. We now analyze the behavior of CS. As can be seen by direct comparison of Figures 2.27(b) and 2.27(d), the performance of CS does not depend on the type of frequency mask. This behavior is expected because, according to the CS theory, the reconstruction algorithm only depends on the number of zero elements of the sparse matrix $\mathbf{S}$ and not on their position. This is important from a practical standpoint as it makes CS algorithms suitable for any type of spectral shape of the signal, given that it is sufficiently sparse. In fact, CS can perfectly recover the signal through the collection of a number of packets $M$ that is much smaller than the number of nodes $N$ in the network (in most cases, recovering $M = 50$ packets from $N = 100$ nodes suffices).

We also observe that the reconstruction error has a very sharp drop from a completely random reconstructed signal to the correct signal. This drop occurs over the course of the reception of a relatively small fraction of the total number of packets (on the order of 10%). Hence, by tracking how the reconstructed signal varies at the sink, it is easy to determine when a sufficient amount of data has been received and the data dissemination process for the current set of sensor readings can be terminated. These observations differ from the findings in [47, 48], where the authors state that the recovery results in a gradually varying reconstruction error as more and more packets are received at the sink.

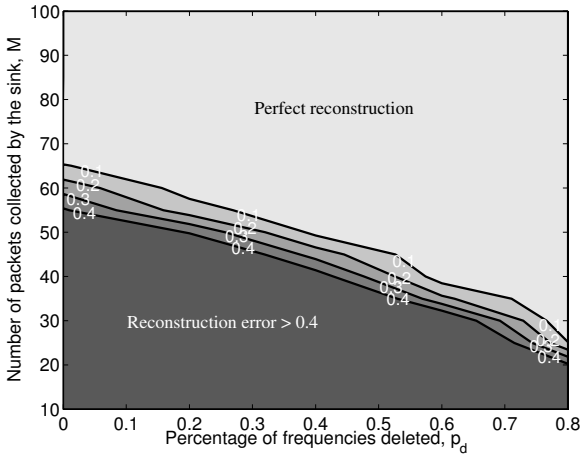



**Figure 2.28.** *CS, high-frequency mask, block-matrix: reconstruction error for varying $p_d$ and $M$.*

We now discuss the impact of different non-idealities of $\mathbf{\Phi}$, such as a block structure for non-zero entries and setting randomly picked elements of the matrix to 0 (M2 and M3). We performed simulations for all of these cases. In Figure 2.28, we only show the worst case

where the matrix presents a combination of the discussed non-idealities. $\mathbf{\Phi}$ is composed of 3 sub-matrices of equal size along the diagonal of the original matrix, and all other entries are set to 0. In addition, 30% of the remaining entries in the sub-matrices are set to 0. As can be seen from the graph, the results of this last set of simulations closely match those we obtain for the ideal case M1. The same holds for simulations where the M2 and M3 non-idealities are applied separately. Further tests, where we restrict the range of non-zero entries to only two different values, also confirm the results. This robustness with respect to the matrix $\mathbf{\Phi}$ is very important, since it allows the exploitation of CS in real networks with topology constraints, as we show in the Appendix 2.C.

## 2.C Performance Comparison of CS with Selected Protocols

In the Appendix 2.B we investigated different measurement matrices $\mathbf{\Phi}$, that correspond to extreme cases for dissemination processes, and compare the performance of Compressive Sensing and Random Sampling for different types of signals $\mathbf{X}$. In this appendix, we report further analysis results obtained when Compressive Sensing is applied to synthetic signals that mimic realistic WSN signals.

In detail, we now consider true multi-hop topologies and quantify how selected data gathering protocols perform in terms of energy cost (total number of transmissions) as well as reconstruction quality, as defined in Equation (2.33)).

**Network scenario and topology:** as explained in Section 2.3.2.

**Selected protocols:** the aim of our performance evaluation is to asses the benefits that CS brings about in multi-hop networks. In what follows, we introduce a few idealized schemes so as to represent different data gathering protocol classes. Note that we assume a unit cost for each packet transmission, and that we ignore processing overhead at the nodes, since it is expected to be cheap compared to the cost of packet transmission.

P1. *Random sampling (RS):* as P1 in Section 2.3.3.

P2. *Random sampling with CS (RS-CS):* as P2 in Section 2.3.3.

P3. *Data aggregation (DA):* for this scheme we account for a preliminary setup phase where nodes are grouped into clusters of depth $D$ as follows. The process starts by collecting into a first set $Q$ all nodes that are directly connected to the sink. For any given node $m \in Q$, we consider the tree $T_D$ of depth $D$ that has node $m$ as the root and includes $m$'s children up to a distance of $D$. We proceed as follows: A1) group the nodes within the tree $T_D$ into a cluster $C_m$ having node $m$ has the cluster head, A2) remove node $m$ from $Q$, and A3) consider all leaf nodes of $T_D$ and insert their children in $Q$. We continue the process in the same manner by picking the next element in $Q$, and so forth, until the set is empty. The order with which we pick elements from $Q$ is irrelevant. After the clusters are formed, we proceed with the data collection phase as follows: B1) all the nodes in the cluster send their packet to the cluster head using multi-hop geographic routing. This packet includes the value sensed at the originat-

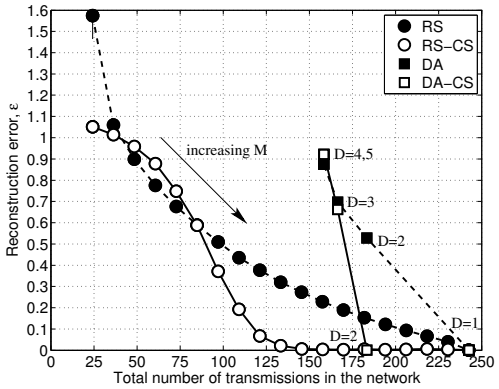
ing node. B2) the cluster head collects all incoming packets and averages the values therein with its own measurement. B3) the cluster head sends this averaged value to the sink using the shortest path exactly as in RS. The total cost of this operation is given by the number of transmissions occurring within the cluster (nodes $\rightarrow$ cluster head) plus the number of transmissions needed to reach the sink from the cluster head (cluster head $\rightarrow$ sink), which is equal to the number of hops in the path. This simple representation of data aggregation captures the main characteristics of more complex data aggregation schemes sufficiently well. The parameter $D$ can be varied in the simulations to obtain suitable trade-offs between representation accuracy (small $D$) and overall transmission costs (that decrease for higher $D$ values).

P4. *Data aggregation with CS (DA-CS):* nodes are aggregated into clusters as in DA. Thus, for any given cluster $C_m$ with cluster head $m$, all nodes transmit their sensed value to the cluster head. This incurs the same cost as in DA. However, the cluster head, instead of performing a simple average of these values, computes a weighted average as $y_m = \sum_{u \in C_m} \alpha_u x_u$, where $\alpha_u = \texttt{rand}()$ and $x_u$ is the value sensed by node $u$. As above, $y_m$ is stored in a packet along with all combination coefficients $\alpha$. The packet is then routed to the sink using the same strategy as in RS-CS. Traversed nodes combine the aggregated value with their *own* reading, multiplied by a random coefficient, and include the used coefficient in the packet. The cost of this second transmission phase again equals the number of hops separating node $m$ from the sink.

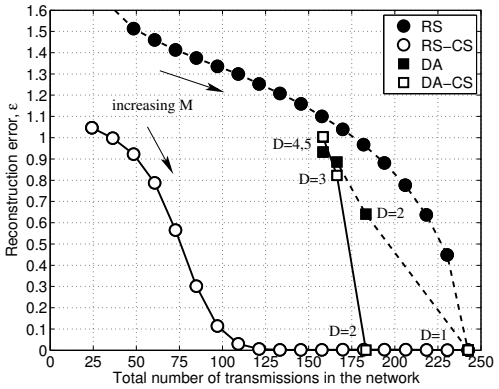
**Reconstruction quality:** here we consider the quantity $\varepsilon$, computed as explained in Section 2.3.4, Equation (2.33).

**Performance evaluation:** In the following, we discuss the performance of the above protocols in terms of reconstruction error and total energy cost.

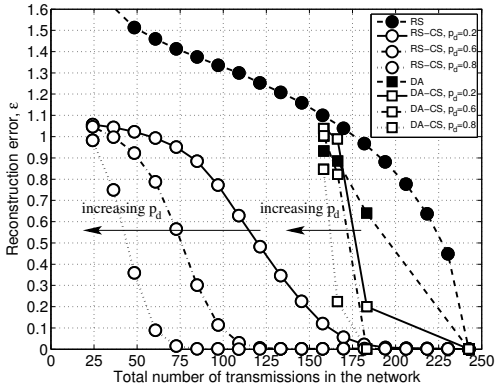
In Figures 2.29(a) and 2.29(b), we show the reconstruction error $\varepsilon$ as a function of the total transmission cost for RS, RS-CS, DA, and DA-CS, considering low- and high-frequency signals with $p_d = 0.6$. Again, we use $N = 100$ for the total number of sensor nodes. Results for RS and RS-CS were obtained by varying the parameter $M$ from $5$ to $N$ in steps of $5$. The obtained tradeoff curves are traversed from left to right for increasing $M$. Results for DA and DA-CS were instead obtained by varying the cluster depth $D$ from one to five. Tradeoff



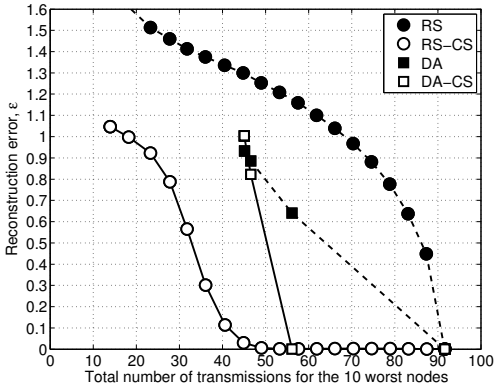

(a) Low-frequency, $p_d = 0.6$.




(b) High-frequency, $p_d = 0.6$




(c) High-frequency, $p_d \in \{0.2, 0.6, 0.8\}$.




(d) 10% worst nodes, $p_d = 0.6$

**Figure 2.29.** *Reconstruction error vs transmission cost for the selected data gathering schemes.*

curves in this case are traversed from left to right for decreasing $D$. Each data point in the graphs is obtained by averaging over $10000$ simulations. To improve the readability of the graphs, we omit confidence intervals but confirm that they are smaller than 1% of the plotted value in all cases.

We observe that the use of Compressive Sensing in both high- and low-frequency cases drastically improves the performance of the considered baseline schemes. In particular, in the high-frequency case, CS allows the perfect reconstruction of the signal in all of the cases we studied, whereas standard solutions achieve the same goal only when all sensor readings are collected at the sink, which has a much higher cost. For example, for a reconstruction error threshold of $\varepsilon = 0.1$, we see from Figure 2.29(a) that CS requires $125$ transmissions for recovery, while RS needs almost twice as many transmissions to reach the same quality. For schemes exploiting data aggregation, DA-CS allows an excellent recovery with $D = 2$,

while standard DA techniques do so for $D = 1$, i.e., upon receiving $100\%$ of the packets.[22] This is true for both high- and low-frequency signals. Also, in Figure 2.29(a) there are data points for which RS slightly outperforms RS-CS. These, however, are not meaningful as they reside in a region that should be avoided as the reconstruction error $\varepsilon$ is too high for a useful reconstruction of the original signal ($\varepsilon \geq 0.55$).

As a further observation, we see that DA-CS is more energy consuming than RS-CS for the same reconstruction quality. This can be explained as follows. For large $D$, all nodes within each cluster will be transmitting through multiple hops to reach their cluster head. From here on, the data will be aggregated and sent via unicast to the sink using the same strategy as in RS-CS. However, the first transmission phase (within the clusters) dominates the total cost. Instead, when $D$ is small, say, $D = 2$, we aggregate packets only among nodes that are a single hop away. In this case, the transmission cost is still high as all nodes transmit and only a few packets are aggregated. Overall, for the type of networks that we consider here, RS-CS is the most efficient technique for data collection and recovery. Note that this is not obvious *a priori* as one might expect DA-CS to outperform RS-CS in terms of reconstruction quality, since the packets received at the sink using DA-CS combine more information. From this first set of results we can conclude that mixing information from the different sources that a given packet encounters along its shortest path towards the sink suffices to achieve a good reconstruction of the original signal if 1) the $M$ source nodes are picked uniformly at random within the WSN field and 2) $M$ is a sufficiently large fraction of the total number of readings. In addition, this strategy is superior in terms of transmission cost to mixing information within clusters of nodes and then sending the result to the sink. This is an important result as RS-CS is inherently simpler to implement in distributed fashion, whereas DA-CS requires the organization of nodes into clusters.

Finally, note that schemes exploiting CS are mainly affected by the number of significant frequencies of the signal and not by their location within the DCT domain, as seen in Appendix 2.B. This is confirmed by Figures 2.29(a) and 2.29(b) where the performance of CS techniques does not significantly change, whereas the performance of RS and DA degrades significantly for high-frequency signals.

Further results are given in Figure 2.29(c), where we consider high-frequency signals and vary $p_d \in \{0.2, 0.6, 0.8\}$. A single curve is plotted for RS and DA as we found that the behavior of these schemes only marginally depends on $p_d$. For increasing $p_d$, CS curves shift

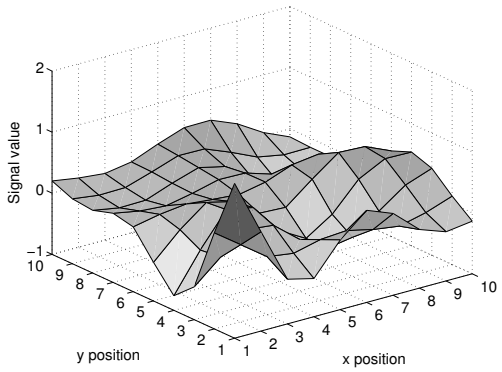
[22] For $D = 1$, each node forms its own cluster of size one.

to the left, which means that a given reconstruction quality requires a smaller number of transmissions. As expected, RS-CS performs better for increasing $p_d$. The same is true for DA-CS: for this scheme we obtain a good reconstruction quality for $D = 2$ when $p_d = 0.2$, while for $p_d = 0.6$, with this same cluster depth, the reconstruction error is nearly zero. For $p_d = 0.8$, we can approximate the signal with good accuracy already for $D = 3$.

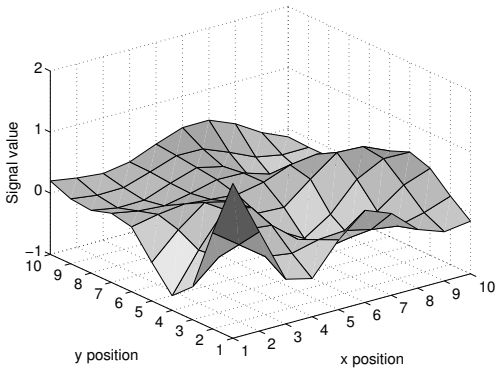
Also for other band-pass signals with frequency components in an intermediate range, the performance results are exactly the same for the case of CS, while RS performance is in between the cases for high and low frequencies.

We now consider a different performance metric. Specifically, we pick the $10\%$ of the nodes that performed the highest number of transmissions in each simulation and average their costs. Note that this metric is tightly related to the network lifetime. For a given topology, the nodes with the highest energy consumption will be the first to drain their batteries, thus impacting the network connectivity. Due to the Funneling effect [82], these are usually the nodes close to the sink. Figure 2.29(d) shows the tradeoff performance of the selected schemes using the latter metric. We can observe that DA-CS and RS-CS perform closely for $D = 2$ (DA-CS) and $M = 60$ (RS-CS). These parameters are the smallest that result in a zero reconstruction error for both protocols. Finally, for this cost metric, the difference between the two schemes is substantially reduced, which makes DA attractive when the objective is to prolong the network lifetime.

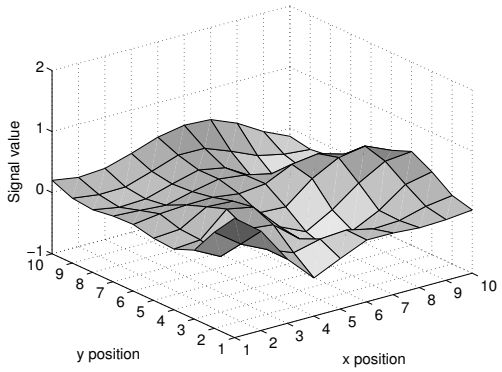
To give a better intuition for the shape of the signals under consideration, as well as the nature of the reconstruction error, we provide a graphical representation of example signals in Figure 2.30. The original 2D signal for a low frequency case with $p_d = 0.6$ is shown in Figure 2.30(a). From Figure 2.30(b) we can see that with $M = 60$ packets received at the sink node, RS-CS allows to perfectly reconstruct the signal ($\varepsilon = 0$). The total number of transmissions required is 146. The RS scheme incurs the same transmission cost, but only achieves a reconstruction error of $\varepsilon \simeq 0.4$. As is typical for the RS signal reconstruction shown in Figure 2.30(c), it captures the overall signal relatively well, but the spike in the signal at position (1,1), as well as the minimum value at position (1,4) are not sampled. Such extreme values are captured reliably only when $M \approx N$. Aggregation with the DA scheme for $D = 2$ is shown in Figure 2.30(d). It is better able to represent those extreme values (although not perfectly). However, it requires a total of 182 transmissions and has an even higher overall reconstruction error of $\varepsilon \simeq 0.5$. DA-CS for $D = 2$ (not shown in the graphs) by construction has the same transmissions cost of 182 as DA. It achieves the same perfect



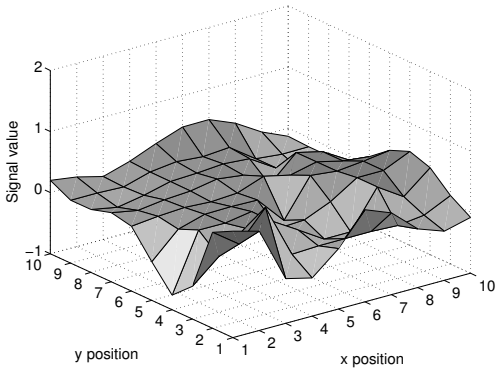

(a) Original signal.




(b) RS-CS reconstruction with $M = 60$.




(c) RS reconstruction with $M = 60$.




(d) DA reconstruction with $D = 2$.

**Figure 2.30.** *Original and reconstructed signal for a low frequency signal with $p_d = 0.6$ for the selected data gathering schemes.*

signal reconstruction as the RS-CS scheme.

## 2.D Preliminary Performance Evaluation of joint CS and PCA

### 2.D.1 Analysis of signals with a fixed support

In this section we study the effectiveness of joint CS and PCA recovery, presented in Sections 2.4 and 2.5, when applied to synthetic signals that are measured through the grid network model used in Section 2.3.

**Network:** as explained in Section 2.3.2.

**Signals:** as explained in Section 2.3.1 for Synthetic signals.

**Data gathering:** similarly to what done in Section 2.3, the *data collection* at the generic time $k$ adopts a simple *random sampling* scheme as follows. Each node becomes a source with probability $p = L/N$, which was varied in the simulations to obtain tradeoff curves for increasing transmission overhead. Hence, on average $L$ nodes transmit a packet containing their own sensor reading. Each packet is routed to the sink via geographic routing. The sink collects incoming data from all transmitting nodes according to $\mathbf{y} = \mathbf{\Phi x}$, where $\mathbf{x}$ is the original signal and $\mathbf{\Phi}$ represents the routing matrix. $\mathbf{\Phi}$ has a single one in each row and at most a single one in each column. In detail, row $i$ with $1 \leq i \leq M$ has a one in column $j$ with $1 \leq j \leq N$ if the $i$-th packet received by the sink was transmitted by node $j$. The cost of delivering a single packet to the sink is given by the number of hops that connect the source node to the sink.[23]

**Recovery:** we consider the following recovery techniques:

R1. *Random sampling with Spline interpolation (RS-Spline):* the signal is reconstructed by spline interpolation [57] of the values collected through RS.

R2. *Compressive Sensing (RS-DCT-CS):* we use the CS recovery technique described in Section 2.1, where $\mathbf{\Phi}$ is the RS routing matrix defined above and $\mathbf{\Psi}$ implements the DCT transformation.

R3. *Compressive Sensing with PCA (RS-PCA-CS):* the original signal is recovered through joint CS and PCA, as described in Section 2.4. The sample mean $\overline{\mathbf{x}}$ and the covari-

[23] Other cost metrics, e.g., energy, could also be used.

ance matrix $\widehat{\mathbf{\Sigma}}$ are calculated from a large enough number of instances of the synthetic signal so as to obtain accurate estimates of these statistics.

**Results:** to simplify the investigation and to pinpoint the fundamental performance trade-offs, we assume a unit cost for each packet transmission. The metrics of interest are the total number of transmissions in the network for any given time $k$ and the reconstruction quality at the sink, defined as $\varepsilon = \|\mathbf{x} - \widehat{\mathbf{x}}\|_2 / \|\mathbf{x}\|_2$, where $\mathbf{x}$ is the original signal and $\widehat{\mathbf{x}}$ is the signal reconstructed at the sink from the received samples $\mathbf{y}$.

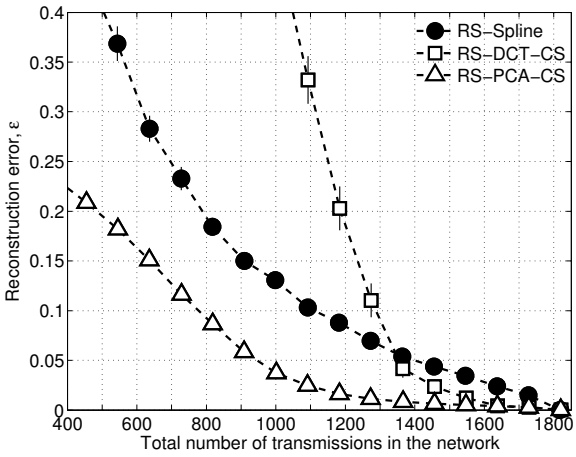



**Figure 2.31.** *Performance of three different recovery techniques for a synthetic low-pass signal: number of transmissions per data collection vs $\varepsilon$.*

In Figure 2.31 we compare the performance of the above recovery techniques in terms of $\varepsilon$ *vs* total number of transmissions per data collection for a low-pass signal. RS-DCT-CS outperforms RS-Spline only when $L$ approaches $N$, i.e., when the sink receives nearly all $N$ packets and the total number of transmissions is close to the maximum (about $1800$ for the considered network). In addition, the gain that RS-DCT-CS can provide is very small. Instead, RS-PCA-CS recovery significantly outperforms both RS-Spline and RS-DCT-CS for all values of $L$ and allows the recovery of $\widehat{\mathbf{x}}$ with small reconstruction errors. For example, an error requirement of $\varepsilon = 0.05$ is achieved in RS-PCA-CS with about $1000$ transmissions, whereas RS-DCT-CS would need $50\%$ more transmissions for the same error performance. We note that the performance of RS-Spline for high-frequency signals would be significantly worse, whereas the performance of RS-DCT-CS and RS-PCA-CS remains almost the same.

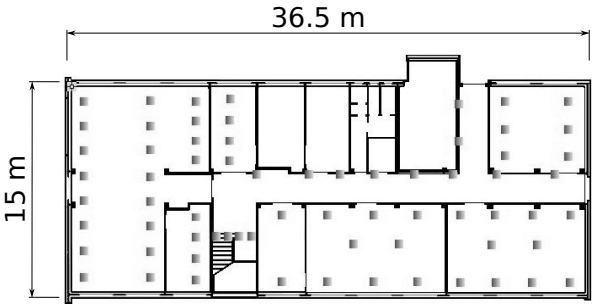



**Figure 2.32.** *Layout of the WSN testbed.*

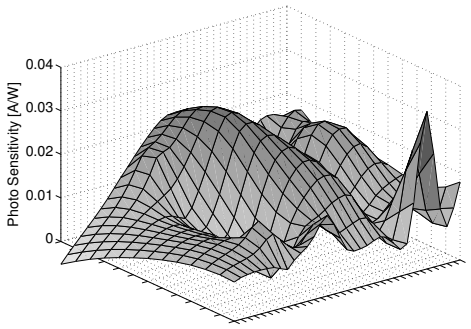



**Figure 2.33.** *Signal sample: luminosity in the range* $320 - 730$ *nm.*

### 2.D.2 Analysis of real signals from a WSN testbed

To test whether the proposed scheme works in realistic scenarios, in this section we apply the joint CS and PCA recovery described above to the signals that we gathered from an actual WSN deployment.

**Network:** we consider the WSN testbed of Figure 2.32.[24] This experimental network is deployed on the ground floor of the Department of Information Engineering at the University of Padova. The WSN consists of $N = 68$ TmoteSky wireless nodes equipped with IEEE 802.15.4 compliant radio transceivers.

**Signals:** From the above WSN, we gathered five different types of signals $\mathbf{x}$: 1) temperature, 2) humidity, 3) voltage, 4-5) luminosity in two different ranges ($320-730$ and $320-1100$ nm, respectively), collecting measurements from all nodes every $5$ minutes for $3$ days. We repeated the data collection for three different measurement campaigns, choosing different days of the week, see Table 2.2 for deployment W1. Figure 2.33 shows an example signal of type 4, i.e., luminosity in the range $320 - 730$ nm.

**Data gathering and Results:** to test the effectiveness of the proposed technique we considered the real data collected through the testbed in Figure 2.32 and a data gathering scheme based on geographic routing. We placed the sink in the center of the network, where the signal is reconstructed at each time $k$ based on our joint CS and PCA technique. Note that

[24] Our framework is flexible and does not depend on a specific topology; the only requirement is that the sensor nodes can be ordered, e.g., based on their IDs.

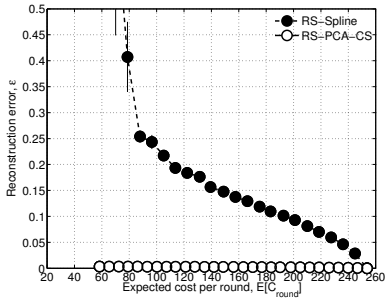



**Figure 2.34.** $\varepsilon$ *vs* $E[C_{\mathrm{round}}]$: *humidity.*

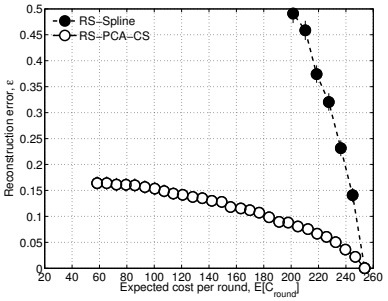



**Figure 2.35.** $\varepsilon$ *vs* $E[C_{\mathrm{round}}]$: *luminosity.*

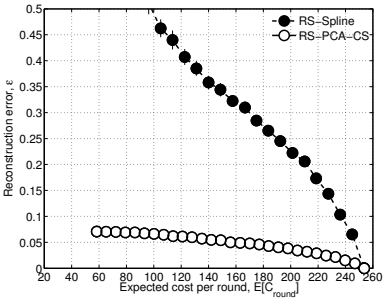



**Figure 2.36.** *Average* $\varepsilon$ *(signals 1–5) vs* $E[C_{\mathrm{round}}]$.

the signals in the testbed differ from those we generated in the previous section as they do not necessarily have the well-defined low-frequency representation that was assumed in Section 2.D.1 and are characterized by spatial and temporal correlations that are in general non-stationary. This means that the statistics that we use in our solution (i.e., sample mean and covariance matrix) must be learned at runtime and might not be valid through the entire data collection phase. Hence, in order to implement PCA in conjunction with CS for real signals, we alternate the following two phases, as proposed also in Section 2.4:

**1.** a *training phase* of $K$ data collection rounds, during which the sink collects the readings from all $N$ sensors and uses this information to compute $\overline{\mathbf{x}}$ and $\widehat{\mathbf{\Sigma}}$ as in Equation (2.34);

**2.** a subsequent *monitoring phase* of $\zeta K$ rounds during which, on average, only $L \leq N$ nodes become sources according to the random sampling scheme of Section 2.D.1 (each with probability $p = L/N$). The input signal is thus reconstructed from this data subset, using the statistics $\overline{\mathbf{x}}$ and $\widehat{\mathbf{\Sigma}}$ computed in the previous phase.

The ratio $\zeta$ between the duration of monitoring and training phases should be chosen according to the temporal correlation of the observed phenomena.

In Figures 2.34–2.36 we show the performance in terms of reconstruction error ($\varepsilon$) as a function of the average cost per round, which is given by the number of transmissions for the collection of a single instance of the signal $\mathbf{x}$. In these plots each training phase lasts $K = 2$ rounds and $\zeta = 4$ (the impact of these parameters is addressed at the end of this section). A training phase entails a cost $KC_N$, where $C_N$ is the total number of transmissions needed to gather the readings from all nodes. The average number of packets sent during the following $2\zeta = 8$ monitoring phases depends on $p$, which is varied from $1/N$ to 1, and

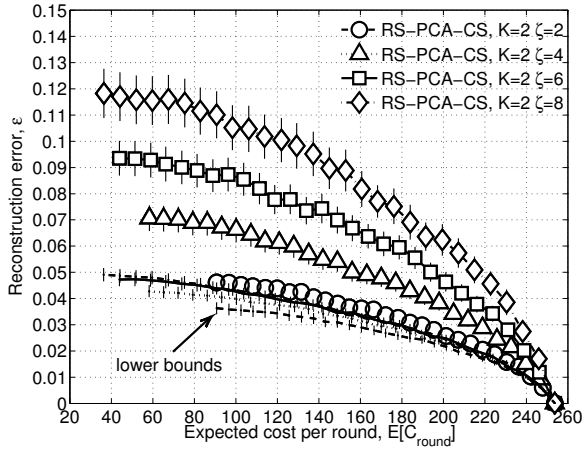



**Figure 2.37.** *Average $\varepsilon$ (signals 1-5) vs* $E[C_{\text{round}}]$, $K = 2$, $\zeta \in \{2, 4, 6, 8\}$.

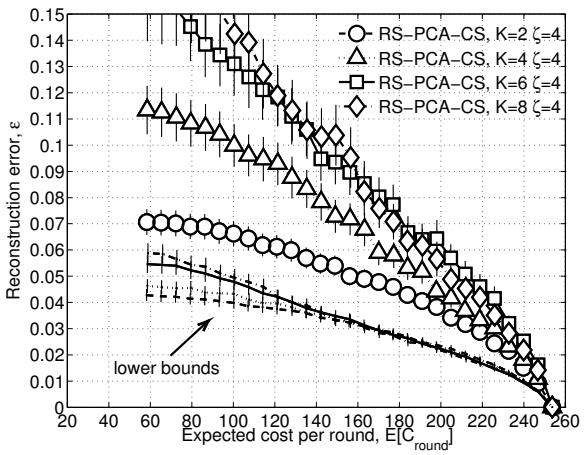



**Figure 2.38.** *Average $\varepsilon$ (signals 1-5) vs* $E[C_{\text{round}}]$, $K \in \{2, 4, 6, 8\}$, $\zeta = 4$.

$\varepsilon$ is computed for each case. For a given $p = L/N$ each monitoring phase has an average total cost of $\zeta K E[C_L]$, where $E[C_L]$ is the total number of transmissions needed to collect the readings from the source nodes during a data collection round. Thus, the average cost per round is calculated as:

$$E[C_{\text{round}}] = \frac{C_N + \zeta E[C_L]}{1 + \zeta} \ . \tag{2.79}$$

For comparison, in the plots we also show the recovery performance of RS-Spline, see Section 2.D.1. The cost per round for RS-Spline is $E[C_L]$.[25] In Figures 2.34–2.36 we demonstrate the effectiveness of our recovery technique ("RS-PCA-CS" in the figures). These results show that PCA is a suitable transformation to be used in conjunction with CS and that, despite the cost incurred in the training phases, the approach still provides substantial benefits with respect to standard data gathering schemes. In Figure 2.34 $\varepsilon$ is close to zero as this specific signal varies slowly in time, i.e., its correlation structure is quasi-stationary during a monitoring phase. Also, we note that for those signals showing higher variations over space and time, such as luminosity, RS-Spline has unsatisfactory performance.

In the last two graphs, Figures 2.37 and 2.38, we show the impact of $K$ and $\zeta$ on the performance. From Figure 2.37 (fixed $K$) we see that decreasing $\zeta$ leads to: 1) a higher minimum admissible cost to bear per round due to an increase of the overhead 2) despite the increase of overhead, a decreased cost per round for a given quality goal since the signal's

[25] We do not analyze the performance of RS-DCT-CS. In contrast to the the synthetic signals of Section 2.D.1, the real signals considered here are not sparse in the DCT domain, and thus RS-DCT-CS performs much worse than RS-Spline.

reconstruction algorithm uses fresher information and 3) a smaller variance for $\varepsilon$. From Figure 2.38 (fixed $\zeta$) we see that decreasing $K$ is beneficial. This means that, for the considered signals, a smaller reconstruction error is achievable through more frequent updates of $\overline{\mathbf{x}}$ and $\widehat{\mathbf{\Sigma}}$. In Figures 2.37 and 2.38 solid and dotted lines without marks represent lower bounds on the error recovery performance, which are obtained as follows. For each $(K, \zeta)$ pair, the actual recovery performance evaluates the reconstruction accuracy of the signal when training and monitoring phases alternate. In this case, during a monitoring period each input signal $\mathbf{x}_k$ is reconstructed using RS-PCA-CS with $\overline{\mathbf{x}}$ and $\widehat{\mathbf{\Sigma}}$ calculated exploiting the signals gathered during the last training phase. Differently, the lower bound on the reconstruction error of RS-PCA-CS for each $(K, \zeta)$ pair and for each time $k$ is obtained using RS-PCA-CS with $\overline{\mathbf{x}}$ and $\widehat{\mathbf{\Sigma}}$ calculated assuming perfect knowledge of the previous $K$ instances of the signal $\mathbf{x}_{k-1}, \ldots, \mathbf{x}_{k-K}$. The cost associated with the new $\varepsilon$ is set equal to that of the real RS-PCA-CS scheme for the given $(K, \zeta)$ pair. These curves reveal the impact of the obsolescence of $\overline{\mathbf{x}}$ and $\widehat{\mathbf{\Sigma}}$ during the monitoring phase for the considered signals. In particular, the recovery performance degrades for either increasing $\zeta$ (Figure 2.37) or $K$ (Figure 2.38).

## 2.E SCoRe1 Framework: Justification of Choices

In order to illustrate the choices done for the design of SCoRe1, see Section 2.5.1, in this appendix we consider two simple strategies for iteratively sensing and recovery a given signal. In particular, this section aims to explain the reasons for: 1) the adoption of an iterative and approximate training set $\widehat{\mathcal{T}}_K^{(k)}$ and 2) the definition of both an Error Estimation and a Feedback Control block in our monitoring framework.

To this end, let us consider two simple strategies that can be executed at the Data Collection Point. According to the first one, which corresponds to the approach adopted in the Appendix 2.D and here is referred to as *2 Phases*, the network monitors the entire signal $\mathbf{x}^{(k)}$ for a certain period of time (referred to as *training phase*) and sends it to the DCP, which is responsible for inferring the signal statistics during this period. For the subsequent period (*monitoring phase*), the DCP only requires a small fraction of the nodes to transmit, being able to accurately reconstruct the signal from its under-sampled version. Due to the fact that the monitored signal is non-stationary, its statistics may vary with time and should therefore be periodically updated at the DCP. In detail, this protocol alternates the following two phases of fixed length:

1. a *training phase* of $K_1$ data collection rounds, during which the DCP collects the readings from all $N$ sensors and uses them to compute the statistics needed by the recovery algorithm, see Section 2.4.1. During this phase, the probability of transmission at each sensor is set to $p_{tx}^{(k)} = 1$, so the DCP collects a training set $\mathcal{T}_{K_1} = \{\mathbf{x}^{(k-K_1)}, \ldots, \mathbf{x}^{(k-1)}\}$ that will be used to infer the relevant statistics;

2. a subsequent *monitoring phase* of $K_2$ rounds, with $K_2 \geq K_1$, during which (on average) only $L \leq N$ nodes transmit, according to the adopted random sampling scheme with $p_{tx}^{(k)} = L/N$. The signal of interest is thus reconstructed from this data set by the recovery algorithm, using the statistics computed in the training phase.

Note that for the *2 Phases* technique the training set $\mathcal{T}_{K_1}$ does not contains approximations (reconstructions) of the past signals, but those actually collected during the training phase. The major drawback of this technique is that it is very sensitive to the choice of the parameters that govern the compression and the recovery phases. These parameters are: (1) the average number of sensors $L$ that transmit during the monitoring phase, which determines $p_{tx}^{(k)} = L/N$. (2) The length of the training phase ($K_1$) and of the monitoring phase ($K_2$),

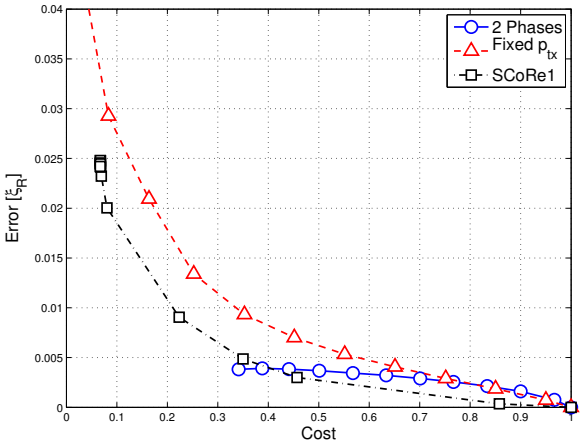



**Figure 2.39.** *Performance comparison of three iterative monitoring schemes, with online estimation of the past, for signals in class C1, temperature and humidity.*

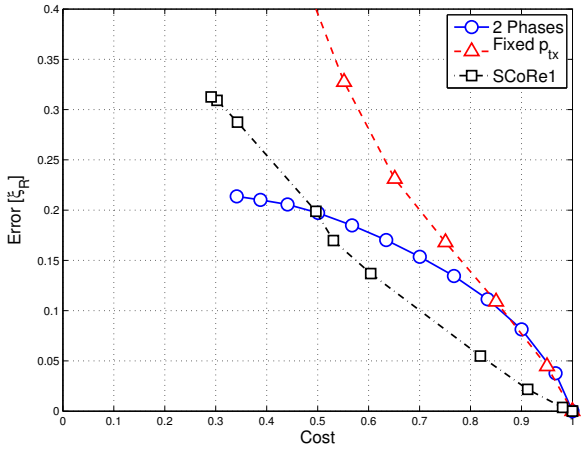



**Figure 2.40.** *Performance comparison of three iterative monitoring schemes, with online estimation of the past, for signals in class C2, photo sensitivity in the range* $320 - 730$ *nm and in the range* $320 - 1100$ *nm.*

that should be chosen according to the temporal correlation of the observed phenomenon. These parameters must be chosen at the beginning of the transmission and they can only be tuned manually. Hence, even if the initial choice is optimal, this technique is not able to adapt to sudden changes in the signal statistics. Moreover, the training phase accounts for the biggest part of the total cost in terms of number of transmissions, as shown by the study presented in the Appendix 2.D.

A solution to the latter problem is to eliminate the training phase, so the nodes at each time $k$ transmit with a fixed probability $p_{tx}^{(k)} = p_{tx}$: this is the *Fixed* $p_{tx}$ technique. Here, the training set needed by the recovery block to infer the statistics that allows the reconstruction of $\mathbf{x}^{(k)}$ from $\mathbf{y}^{(k)}$ is formed by the previously reconstructed signals $\widehat{\mathbf{x}}^{(j)}$ for $j < k$, so it can be written as $\widehat{\mathcal{T}}_K^{(k)} = \{\widehat{\mathbf{x}}^{(k-K)}, \ldots, \widehat{\mathbf{x}}^{(k-1)}\}$. The main drawbacks of this scheme is that without any control loop for $p_{tx}$, the energy consumption cannot be easily adapted to the observed signal and the reconstruction error can grow unbounded as shown shortly.

To compare SCoRe1 with the above schemes, *2 Phases* and *Fixed* $p_{tx}$ we use signals belonging to two out of the three different classes discussed in Section 2.5.3. In particular, we consider: C1) signals with high temporal and spatial correlation (e.g., the ambient temperature or the ambient humidity) and C2) signals with lower correlation (e.g., the photo sensitivity or the wind direction).

The results, reported in Figures 2.39 and 2.40, have been obtained from 100 independent

simulation runs and by averaging the performance over all signals in each class. The x-axis of these figures represents the normalized cost expressed as the average fraction of packet transmissions in the network per time sample, computed according to Equation (2.70); the y-axis still shows the signal reconstruction error at the end of the recovery process, calculated accordingly to (2.54). In order to vary the cost (x-axis) for the three techniques we modify the following parameters:

1) for *2 Phases* and *Fixed* $p_{tx}$ we vary the probability of transmission $p_{tx}$ that is set at the beginning of the data gathering in the range $]0, 1[$ ;

2) for SCoRe1, we vary the error threshold $\tau$ used in Equation (2.57) in the range $]0, 1[$ setting the feedback control parameters as $C_1 = 1.3$, $C_2 = 3$ and $p_{\min} = 0.05$ .

The training set length has been fixed to $K = K_1 = 2$, whilst the monitoring phase length set to $K_2 = 4$, as suggested in the Appendix 2.D. From Figure 2.39 we note that the three techniques all perform very well in case of a slowly varying signal (C1). However, when the signal varies in an unpredictable way (C2), see Figure 2.40, SCoRe1 clearly outperforms the other two techniques. In particular, SCoRe1 outperforms the *2 Phases* technique because (i) it avoids the training set phase, thus reducing the overall energy transmission required and (ii) updates iteratively its training set thus computing a more adaptive transformation basis for CS and improving its recovery performance. Concerning *Fixed* $p_{tx}$, this technique does not adapt its transmission probability as a function of the reconstruction error that, in turn, gets very large for small values of $p_{tx}$. The feedback loop mechanism implemented in SCoRe1, instead, makes this solution able to adapt to signal variations by decreasing or increasing $p_{tx}$, thus reducing the energy consumption or the estimation error, respectively.

These simulation results allow us to support the design choices, explained in Section 2.5.1, that are at the basis of SCoRe1.

Chapter 3

# Distributed Sub-gradient Methods for Delay Tolerant Networks

**Contents**



In this chapter the principal steps and outcomes of my research activity on Distributed Subgradient Methods for Delay Tolerant Networks (DTNs) are presented and discussed.

The research activity reported in this chapter has been carried out during my PhD visiting period at the *Institut national de recherche en informatique et automatique* (INRIA), Sophia Antipolis, France, under the supervision of Dr. Giovanni Neglia, MAESTRO team.

As previously mentioned, we investigate here a distributed optimization method, presented in Section 3.1, whose objective is that of optimizing network wide (global) performance metrics. In the considered framework nodes collaborate to minimize the sum of local objective functions, which in general depend on global variables such as the network proto-

col parameters or actions taken by all the nodes in the network. In the case where the local objective functions are convex, we can adopt a recently proposed framework that relies on local sub-gradient methods and consensus algorithms to average each node information, while granting converge towards global optimal solutions. In particular, we advocate the use of this framework for DTNs, as explained in Section 3.2. However, existing convergence results for this framework can only be applied to DTNs in the case of synchronous operations of the nodes and mobility models without memory. In our study we address and solve both these issues.

As a first contribution, in fact, we relax the above assumptions. First, in Section 3.3 we prove that the distributed sub-gradient method also converges under a more general Markovian mobility model with memory in the meeting process. In addition, in Section 3.4 we show that a direct application of the framework when nodes operate asynchronously may introduce a bias, leading to convergence to a sub-optimal solution. Hence, we propose some adjustments, and show by simulations that they are able to correct the bias.

Furthermore, inspired by the work in [83], in Section 3.5 we propose a possible application of the framework to a DTN scenario where a Service Provider (SP) disseminates a dynamic content over a mobile social network, with the help of the users that opportunistically share among themselves content updates. In this context the SP should decide how to allocate its bandwidth optimally, and to this purpose it needs to collect information about node utility functions and node meeting rates. We show that distributed sub-gradients can be effectively used to let the nodes perform such optimization.

## 3.1 Distributed Sub-gradient Method's Overview

In this section we review the main results in [20, 21] on convergence and optimality of the distributed sub-gradient method when a random network scenario is considered.

Let us consider a set of $M$ nodes (agents), that want to cooperatively solve the following optimization problem:

**Problem 8** (Global Optimization Problem)**.** *Given $M$ convex functions $f_i(\mathbf{x}) : \mathbb{R}^N \to \mathbb{R}$, determine:*

$$\mathbf{x}^* \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{argmin}} f(\mathbf{x}) = \sum_{i=1}^{M} f_i(\mathbf{x}) \ .$$

Clearly, for the above problem we assume that a feasible solution exists. The difficulty of the task arises from the fact that agent $i$, for $i = 1, 2, \cdots M$, only knows the corresponding function $f_i(\mathbf{x})$, namely its *local objective function*. For example $f_i$ could be a performance metric relative to node $i$, and $f$ could indicate global network performance.

If the functions $f_i$ are differentiable, each node could apply a gradient method to its function $f_i$ to generate a sequence of local estimates, but this would lead to $M$ biased estimates of the solution of Problem 8. In [20] and [21], it is shown that if nodes perform a gradient method but are also able to *average* their local estimates, under opportune conditions, these estimates all converge to a point of minimum of $f$, i.e., $\mathbf{x}^*$.

In particular, a time slotted system is assumed, where, at the end of a slot, each node $i$ communicates its local estimate to a subset of all the other nodes, and then updates the estimate according to the following equation[1]:

$$\mathbf{x}^i(k+1) = \sum_{j=1}^{M} a_{ij}(k)\mathbf{x}^j(k) - \gamma(k)\mathbf{d}^i(k) \ , \tag{3.1}$$

where the vector $\mathbf{d}^i(k) \in \mathbb{R}^M$ is a sub-gradient[2] of agent $i$'s objective function $f_i(\mathbf{x})$ computed at $\mathbf{x} = \mathbf{x}^i(k)$, the scalar $\gamma(k) > 0$ is the step-size of the sub-gradient algorithm at iteration $k$, and $a_{ij}(k)$ are non-negative weights, such that $a_{ij}(k) > 0$ if and only if node $i$ has received node j's estimate at the step $k$ and $\sum_{j=1}^{M} a_{ij}(k) = 1$. We denote by $\mathbf{A}(k)$ the matrix whose elements are the weights, i.e. $[\mathbf{A}(k)]_{ij} = a_{ij}(k)$.

We observe that the first addend in the right hand side of 3.1 corresponds to average according to a consensus algorithm [22].

[1] Also in this chapter, all the real valued vectors are assumed to be column vectors.

[2] $\mathbf{d}^i \in \mathbf{R}^N$ is a sub-gradient of the function $f_i$ at $\mathbf{x}^i \in \text{dom}(f_i)$ if and only if $f_i(\mathbf{x}^i) + (\mathbf{d}^i)^T(\mathbf{x} - \mathbf{x}^i) \leq f_i(\mathbf{x})$ for all $\mathbf{x} \in \text{dom}(f_i)$.

[20] proves that the iterations (3.1) generate sequences converging to a minimum of $f$ under the following set of conditions:

1. the step-size $\gamma(k)$ is such that $\sum_{k=1}^{\infty}\gamma(k)=\infty$ and $\sum_{k=1}^{\infty}\gamma(k)^2<\infty$;
2. the gradient of each function $f_i$ is bounded;
3. each matrix $\mathbf{A}(k)$ is symmetric (then doubly stochastic);
4. it exists $\eta>0$, such that $a_{ii}(k)>\eta$ and, if $a_{ij}(k)>0$, then $a_{ij}(k)\geq\eta$;
5. the information of each agent $i$ reaches every other agent $j$ (directly or indirectly) infinitely often;

6′) there is a deterministic bound for the intercommunication interval between two nodes.

We better formalize conditions 5 and 6′ (resp. 6″). Consider the graph $(\mathcal{V},E_\infty)$, where $\mathcal{V}$ is the set of nodes and the edge $(i,j)$ belongs to $E_\infty$ if nodes $i$ and $j$ communicate infinitely often (i.e., if $a_{ij}(k)$ is positive for infinite values $k$). Condition 5 imposes that the graph $(\mathcal{V},E_\infty)$ is (strongly) connected. Condition 6′ requires that there is a positive integer constant $B$, such that two nodes communicating infinitely often, communicate at least once every $B$ slots, i.e., if $(i,j)\in E_\infty$, then $\max\{a_{ij}(k),a_{ij}(k+1),\cdots,a_{ij}(k+B-1)\}>0$.

In [21], the inter-meeting times are not deterministically bounded, but matrices are required to be independently and identically distributed. In fact, condition 6′ is replaced by the following one:

6″) matrices $\mathbf{A}(k)$ are i.i.d. random matrices.

In such case, $(i,j)\in E_\infty$ if and only if $\mathrm{E}[a_{ij}(k)>0]$. Note that, when the matrices $\mathbf{A}(k)$ are random (like in 6″), condition 5 requires the matrix $\mathrm{E}[\mathbf{A}(k)]$ to be irreducible and aperiodic.

Both papers address also the case when the gradient step-size does not vanish, but it is kept constant ($\gamma(k)=\gamma$). In this case, the sequence of estimates $\mathbf{x}^i$ does not converge in general to a point of minimum of $f$, but it may keep oscillating around one of such point. It is possible to bound the difference between the values that $f$ assumes at the points of a smoothed average of $\mathbf{x}^i$ and the minimum of $f$. As it is intuitive, the smaller $\gamma$, the smaller such difference.

We are going to provide an intuitive explanation of why the results on convergence and optimality hold, and an outline of the proofs in [20,21]. This will be useful for our following

extensions. We first formulate (3.1) in matrix form as follows:

$$\mathbf{X}(k+1) = \mathbf{A}(k)\mathbf{X}(k) - \gamma(k)\mathbf{D}(k) \;, \tag{3.2}$$

where

$$\mathbf{X}(k) \stackrel{\text{def}}{=} \left[\mathbf{x}^1(k+1), \cdots, \mathbf{x}^i(k+1), \cdots, \mathbf{x}^M(k+1)\right]^T$$

and

$$\mathbf{D}(k) \stackrel{\text{def}}{=} \left[\mathbf{d}^1(k+1), \cdots, \mathbf{d}^i(k+1), \cdots, \mathbf{d}^M(k+1)\right]^T \;.$$

This equation iteratively leads to (see Appendix 3.A)

$$\mathbf{X}(k+1) \;\; = \;\; \mathbf{A}_{(1)}^{(k)}\mathbf{X}(1) - \sum_{s=2}^{k} \mathbf{A}_{(s)}^{(k)}\gamma(s-1)\mathbf{D}(s-1) - \gamma(k)\mathbf{D}(k) \;, \tag{3.3}$$

where $\mathbf{A}_{(s)}^{(k)}$, with $s, k \geq 1$ and $s \leq k$, is the *backward matrix product*, i.e.,

$$\mathbf{A}_{(s)}^{(k)} \stackrel{\text{def}}{=} \mathbf{A}(k)\mathbf{A}(k-1)\cdots\mathbf{A}(s) \;.$$

We introduce also the average of all the nodes estimates, $\mathbf{y}(k) \in \mathbb{R}^M$, defined as:

$$\mathbf{y}(k)^T = \frac{1}{M}\mathbf{1}^T\mathbf{X}(k) \;.$$

By Equation (3.2), we obtain[3]

$$\begin{aligned} \mathbf{y}(k+1)^T &= \frac{1}{M}\mathbf{1}^T\mathbf{A}(k)\mathbf{X}(k) - \mathbf{1}^T\frac{\gamma(k)}{M}\mathbf{D}(k) = \\ &= \mathbf{y}(k)^T - \frac{\gamma(k)}{M}\mathbf{1}^T\mathbf{D}(k) \;. \end{aligned} \tag{3.4}$$

Assume for a moment that $\mathbf{x}^i(k) = \mathbf{x}^j(k) = \mathbf{y}(k)$, for each $i$ and $j$, then sub-gradients $\mathbf{d}^i(k)$ are all evaluated in $\mathbf{y}(k)$ and $\mathbf{1}^T\mathbf{D}(k)$ is a sub-gradient of the global function $f$ evaluated in $\mathbf{y}(k)$. Thus, the above Equation (3.4) corresponds to a basic iteration of a sub-gradient method for the global function $f$. The intuitive explanation of the result is that averaging keeps the estimates of $\mathbf{x}^i(k)$, for all $i$, close each other (and then close to $\mathbf{y}(k)$) and makes the local sub-gradients updates equivalent to a sub-gradient update for the global function $f$.

We illustrate this through a simple toy example that we are going to use different times across this chapter. Consider three nodes, labeled as 1, 2 and 3. Their local objective functions are $f_1(x) = f_2(x) = x(x-1)/2$ and $f_3(x) = 2x^2$, where $x \in \mathbb{R}$. Then the global function is $f(x) = \sum_i f_i(x) = 3x^2 - x$, it has minimum value equal to $-1/12$ and a unique point of

[3]Let us recall here that $\mathbf{A}(k)$ is a doubly stochastic matrix, for all $k$.

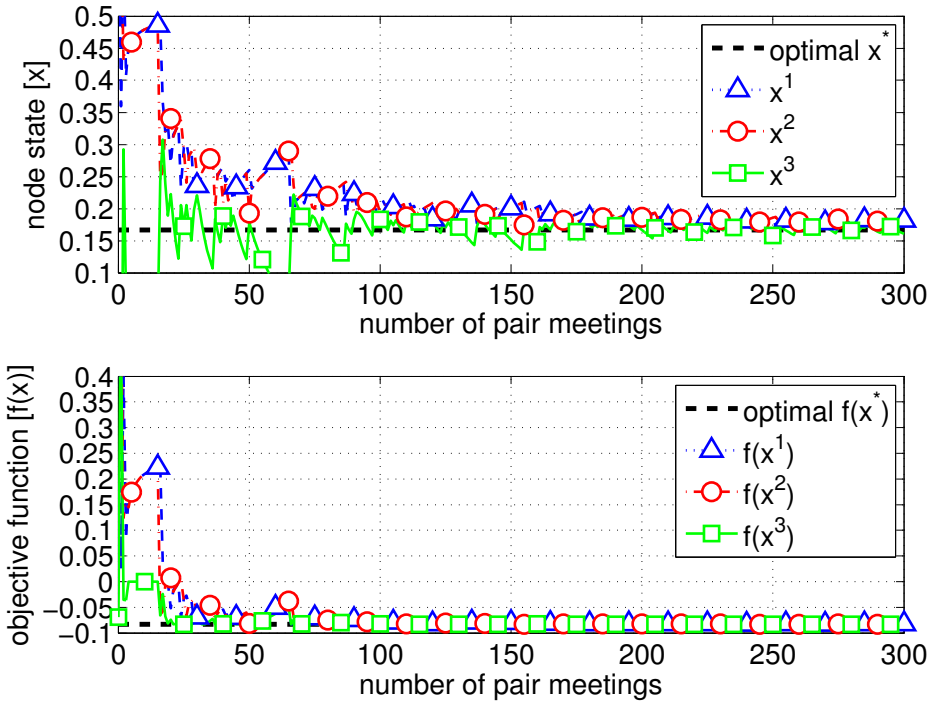



**Figure 3.1.** *Toy example, convergence of the three estimates. Top graph: state's estimates for each node. Bottom graph: objective function value computed in the state's estimates of each node.*

minimum in $x = 1/6$. The weight matrices $\mathbf{A}(k)$ are i.i.d. random matrices. At each step $\mathbf{A}(k)$ is equal to one of the following three matrices

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \tag{3.5}$$

with probability $2/3$, $1/6$ and $1/6$, respectively. Figure 3.1 shows the evolution of the estimates at the 3 nodes, when the algorithm is applied with $\gamma(k) = 1/k$. We can see that state's estimates tends to couple and then converge to the optimal value. In particular, estimates of node 1 and node 2 are kept closer each other since the first matrix of above is selected with higher probability. Similar results have been obtained with $\gamma(k) = \gamma \ll 1$.

The proofs of the convergence results in [20, 21] share mainly the same outline. A key element is proving that the averaging component (the consensus) of the algorithm converges exponentially fast. More formally, under 6′, [20] proves that $\mathbf{A}_{(s)}^{(k)}$ surely converges to the matrix $\mathbf{J} = 1/M\mathbf{1}\mathbf{1}^T$, and that there are two positive constants $C$ and $\beta$ such that $\left\|\mathbf{A}_{(s)}^{(k)} - \mathbf{J}\right\|_{\max} \leq C\beta^{k-s}$ for all $k \geq s$, where given a matrix $\mathbf{A}$ we define the *max norm* as $\|\mathbf{A}\|_{\max} \stackrel{\text{def}}{=} \max\{|a_{ij}|\}$. Under 6″, [21] proves almost surely convergence of $\mathbf{A}_{(s)}^{(k)}$ to $\mathbf{J}$, and an

exponential convergence rate in expectation, i.e. $E\left[\left\|\mathbf{A}_{(s)}^{(k)}-\mathbf{J}\right\|_{\max}\right] \leq C\beta^{k-s}$. Then, similar bounds are established for the distance between $\mathbf{y}$ and $\mathbf{x}^*$, and between $\mathbf{x}^i$ and $\mathbf{y}$, and convergence results (when $\gamma(k)$ satisfies condition 5) and asymptotic bounds (when $\gamma(k)$ is constant) follow from the exponential convergence rate of the averaging component. As a consequence of the different kind of convergence for $\mathbf{A}_{(s)}^{(k)}$ in the two cases, these results hold surely under $6'$ and almost surely under $6''$.

## 3.2 Application to Optimization in DTNs

The distributed sub-gradient method presented in Section 3.1 is particularly appealing in the context of Delay Tolerant Networks, see e.g., [24] and [25]. DTNs are sparse and/or highly mobile wireless ad hoc networks where no continuous connectivity guarantee can be assumed. This intrinsically leads to the impossibility of collecting, at low cost and at a single data processing point, the information needed to solve network optimization problems in a centralized fashion. Due to this, in our study we advocate the use of distributed approaches, which lend themselves well to distributed and communication efficient optimization. To be more concrete, in what follows we briefly discuss two possible DTN scenarios where a global network's function $f$ has to be optimized.

One central problem in DTNs is related to routing packets towards intended destinations. Common techniques, designed to overcome the absence of a complete route to the destination, rely on multi-copy dissemination of messages in the network [84]. In this context, it is natural to define global optimization functions that are able to take in account the trade-off between delivery time and the cost due to the use of resources such as buffer space, bandwidth and transmission power. Functions of this kind are convex and can be written as sum of locally measurable quantities, thus we can optimized them through the distributed sub-gradient framework [85].

A second DTN scenario concerns the dissemination of dynamic content, like news or traffic information. Referring to the application example of [83], we might think of a Service Provider (SP) with limited bandwidth, that has to decide the update rate to assign at each node. Nodes can share their content when they meet with the global objective of maintaining the average information in the network as "fresh" as possible. [83] shows that this problem can also be formalized as a classical convex optimization problem, and that the corresponding global objective function can be expressed in terms of the sum of local functions. The derivation of the latter local functions entails the collection of statistics which are computed at each node only considering its own meeting occurrences. As an application example for our techniques, in Section 3.5 we apply the distributed sub-gradient framework to this scenario.

From a general perspective, the distributed sub-gradient optimization in DTNs can be applied as follows. Nodes exchange their local estimates every time they meet and perform the update step in Equation (3.1) at a given sequence of time instants $\{t_k\}_{k\geq 1}$. This sequence

can either coincide with the meeting times, i.e., each time two nodes meets they exchange and subsequently update their estimates or be independent from them, i.e., in this case $\{t_k\}_{k\geq 1}$ is defined a priori and is known to every network node. In any case, the weight matrices $\mathbf{A}(k)$ originate from the node meeting process. In particular, we can consider the contact matrix $\mathbf{C}(k)$, where $c_{ij}(k) = 1$ if node $i$ has met node $j$ since last time instant $t_{k-1}$, and $c_{ij}(k) = 0$ otherwise. We denote as $\mathcal{C}$ the (finite) set of all possible $M \times M$ contact matrices describing the contacts among $M$ nodes. Each node $i$ can thus calculate its own weights $a_{ij}(k)$, for $j = 1, \ldots, M$, in one of the following two ways (which guarantee that the matrix $\mathbf{A}(k)$ is doubly stochastic):

**Rule 1** (Updates independent from meetings) For $j \neq i$, set $a_{ij}(k) = 1/M$ if $c_{ij}(k) = 1$, otherwise set $a_{ij}(k) = 0$. Set $a_{ii}(k) = 1 - \sum_{j\neq i} a_{ij}(k)$. This method requires each node to know $M$, i.e., the total number of nodes in the system.

**Rule 2** (Updates synchronized with meetings) Whenever node $i$ meets node $j$, it also updates its estimate. In this case, set $a_{ij}(k) = a$ with $0 < a < 1$, $a_{ii}(k) = 1 - a$ and $a_{ih}(k) = 0$ for $h \neq j, i$.

Next, we discuss two key issues that can negatively impact the convergence of the distributed optimization process in a DTN scenario. The first one is related to the validity of Assumptions $6'$ and $6''$. In fact, condition $6'$ is essentially equivalent to assume that there is a deterministic bound for the inter-meeting times of two nodes (that meet infinitely often), and this is for example not the case for all the random mobility models usually considered, see e.g., [86]. Condition $6''$ relaxes $6'$, but requires the independence of the meetings occurring in each time slot $[t_{k-1}, t_k]$, and meetings under realistic mobility are instead correlated (e.g., if in the recent past $i$ has met $j$ and $j$ has met $h$, then the three nodes are likely to be close in space and the probability that $i$ meets $h$ is higher that with uniform and independent mobility). We address this issue in Section 3.3, where we prove that convergence results hold under more general assumptions on the stochastic process of the matrices $\mathbf{A}(k)$.

The second issue is related to the synchronicity of the updates. In fact, the original framework [20,21] requires all the nodes to update their estimates at the same time instants. This is not always feasible in a disconnected and distributed scenario like a DTN. For example, under Rule 2 the reader may have noted that synchronous updates require each node to know when a meeting between any two nodes in the network occurs. This does not appear to be practical. Under Rule 1, which requires each node to know the total number of nodes

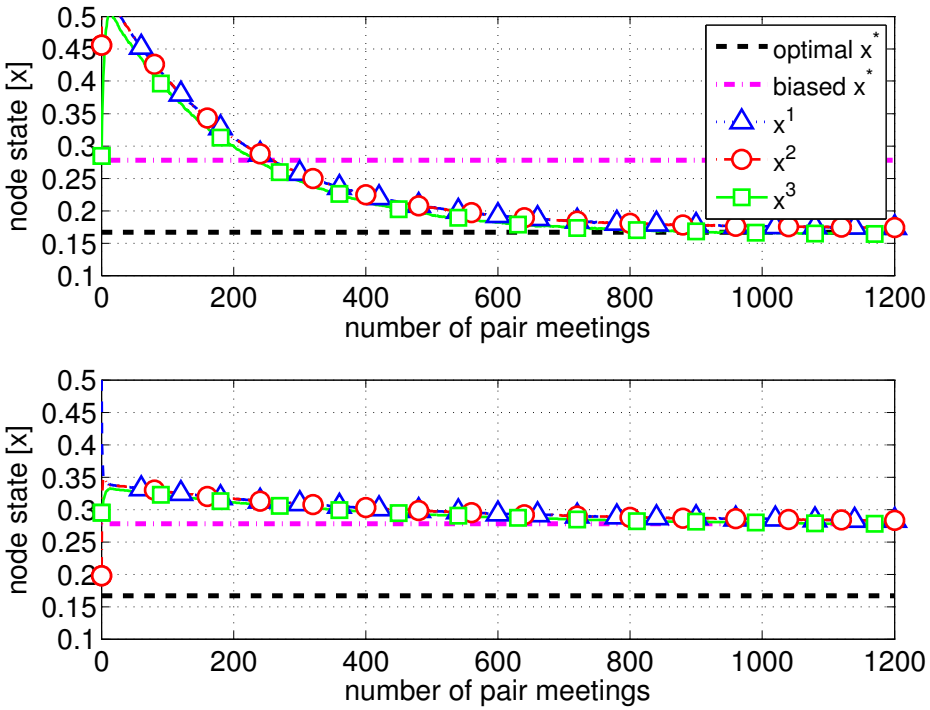



**Figure 3.2.** *Toy example, convergence of the three estimates in case of fixed step-size* $\gamma = 25 \cdot 10^{-4}$. *Results have been averaged over* 100 *simulation runs. Top graph: synchronous updates. Bottom graph: asynchronous updates.*

in the system, nodes should also try to keep their internal clocks synchronized in order to be able to perform their updates at "close enough" time instants and this presents some difficulties as well. We now show through an example that we cannot simply ignore the issue of synchronicity and that a direct application of the algorithm described in the previous section in general does not lead to correct results. Coming back to the toy example presented in Section 3.1, we observe that we can think our three matrices in (3.5) as generated according to Rule 2, when the meeting process has the following characteristics: at each time slot, node 1 and node 2 meet with probability $2/3$, node 1 and 3 meet with probability $1/6$ and node 2 and 3 meet with probability $1/6$. Figure 3.2 shows the evolution of the estimates when the step-size is constant and equal to $25 \cdot 10^{-4}$, both for the synchronous case, where all the nodes update their estimates when a meeting occurs (even the node that is not involved in the meeting), and for the asynchronous case, where only the nodes involved in the meeting perform the update. The curves represents the average estimates over 100 different simulations with different meeting sequences. We note that in the synchronous case (top graph) all nodes agree on the optimal value to set $x$, whereas, in the asynchronous case (bottom graph)

the estimates still converge, but not to the correct point of minimum for the global function $f$. We address this issue in Section 3.4, where we understand the roots of this convergence problem and propose some simple modifications to the basic framework to effectively cope with it.

In our opinion, these extensions to the basic framework proposed in [20,21], while motivated in this study by the DTN scenario, are of wide interest for other possible applications such as mobile wireless ad hoc and sensor networks.

## 3.3 Extension to more General Mobility Models

In our DTN scenario, we consider that the weights are determined from the contacts through a bijective function (as in the case of the two rules presented in Section 3.2). Then conditions $5$ and $6''$ of [21], can be expressed in terms of the sequence of contact matrices as follows: the contact matrices $\mathbf{C}(k)$ are i.i.d. and $\mathrm{E}[\mathbf{C}(k)]$ is an irreducible aperiodic matrix. In this section, we extend the convergence results to the following, more general, mobility model.

**Assumption 1** (Mobility model)**.** *It exists an irreducible, aperiodic and stationary Markov chain $\Phi$ with a finite or countable set of states $S$ and a function $g : S \to \mathcal{C}$, such that $\mathbf{C}(k) = g(\Phi_k)$, for each $\Phi_k \in S$. Moreover, $E[\mathbf{C}(k)]$ is an irreducible aperiodic matrix.*

Since there is a bijective correspondence among weight and contact matrices, we observe that under Assumption 1, it also exists a function $\hat{g} : S \to \mathcal{A}$, such that $\mathbf{A}(k) = \hat{g}(\Phi_k)$. The case when the contact matrices (and then the weight matrices) are i.i.d. is a particular case of our mobility model.

Our proof follows the same outline of [20, 21] presented in Section 3.1. The main issue is to prove the exponential rate of convergence of the backward product $\mathbf{A}_{(1)}^{(k)}$ to $\mathbf{J} = 1/M\mathbf{1}\mathbf{1}^T$.

Before proving the convergence of the backward product, we need to recall an ergodic property of the time shift operator $\theta$ for irreducible, aperiodic and stationary Markov chains. The definitions of measure-preserving and ergodic operators may be found in the Appendix 3.B (see also Chapter V of [87] for more details).

**Proposition 1.** *Given an irreducible aperiodic and stationary Markov chain $\Phi$ with finite or countable states, the shift operator $\theta$ is measure-preserving and ergodic together with all its powers $\theta^k$, where $k \in \mathbb{N}$.*

*Proof.* For stationary Markov chains the shift operator and its powers are measure-preserving by definition of stationarity. Moreover, irreducible, aperiodic and stationary Markov chains with finite or countable states are mixing (see Theorem 3.1 in [88]). From the definition of mixing, if $\theta$ is mixing, also the operator $\theta^k$ is mixing for any given $k \in \mathbb{N}$. But every mixing operator is also ergodic by Theorem 2 in [87], then $\theta^k$ is ergodic for any $k \in \mathbb{N}$. □

We observe that the stochastic process $\mathbf{A}(k) = \hat{g}(\Phi_k)$ is not in general a Markov chain, because different states of $S$ may be mapped to the same weight matrix, but nevertheless it is stationary and ergodic.

We will also need the following result:

**Lemma 1.** *(Windowing a Markov chain) Let $\Phi = \{\Phi_n, n \in \mathbb{N}\}$ be an irreducible, aperiodic and stationary Markov chain. Consider the stochastic process $\Psi = \{\Psi_n, n \in \mathbb{N}\}$, where $\Psi_n = (\Phi_n, \Phi_{n+1}, \cdots, \Phi_{n+h-1})$ with $h$ a positive integer. $\Psi$ is also an irreducible aperiodic stationary Markov chain.*

*Proof.* First of all it is evident that $\Psi$ is also a Markov chain, whose states are possible $h$-uples of states of $\Phi$, e.g $(s_1, s_2, \cdots, s_h)$. The transition probabilities could be calculated starting from those of $\Phi$. Stationarity of $\Psi$ easily follows from the stationarity of $\Psi$. $\Psi$ is also irreducible because $\Phi$ is irreducible. In fact, given two states $s' = (s_1, s_2, \cdots, s_h)$ and $t' = (t_1, t_2, \cdots, t_h)$, for the irreducibility of $\Phi$, it exists $n_0$, such that the chain $\Psi$ moves from $s'$ to a state $u' = (u_1, u_2, u_{h-1}, t_1)$ after $n_0$ steps, and then it is possible to move from $s_h$ to $t_1$. $t'$ is a state of $\Psi$ and therefore it is also a valid sequence of state transitions for $\Phi$, consequently in $h-1$ time steps, $\Phi$ can move from $t_1$ to $t_h$ going through $t_2$, $\cdots$, $t_{h-1}$ and $\Psi$ can move from $u'$ to $t'$. In conclusion in $n_0 + h - 1$ steps, $\Psi$ can move from $s'$ to $t'$.

Aperiodicity requires a more detailed discussion. In detail, given a possible state $s' = (s_1, s_2, \cdots, s_h)$, we want to prove that the greatest common divisor of the possible time steps after which the chain $\Psi$ can return in $s'$ is equal to 1. Note that even if $\Phi$ had the property that it is possible to directly move from each state to itself, for a state $s'$ with $s_i \neq s_1$ for some $i = 2, \cdots, h$, at least $h$ steps are required to return to that state. Consider the minimum number $k_0$ of time steps after which the chain $\Phi$ can move from $s_h$ to $s_1$ (again $k_0$ exists because $\Phi$ is irreducible). Consider then the increasing sequence of all the possible time steps $k_1, k_2, \ldots$ after which it is possible to return in $s_1$. Observe that also $2k_i$ belongs to this sequence. It is clear that $\Psi$ can return in $s'$ after $k_0 + k_1 + h - 1, k_0 + k_2 + h - 1, \ldots$ steps. Let us denote $g$ the greatest common divisor of this sequence of numbers, we have that for each $i > 0$ $(k_0 + k_i + h - 1) \mod g = 0$. In particular also $(k_0 + 2k_i + h - 1) \mod g = 0$, and it follows that $(k_1) \mod g = 0$. This implies that $g$ is also a divisor of the sequence $k_1, k_2, \ldots$. Since $\Phi$ is aperiodic, it follows that $g = 1$. This concludes the proof that $\Psi$ is also aperiodic. □

Now we have all the instruments to study the convergence of $\mathbf{A}_{(1)}^{(k)}$. First, we prove the convergence to $\mathbf{J}$. This is a corollary of results in [89].

**Proposition 2** (Convergence of the backward product)**.** *Let Assumption 1 hold, then*

$$\lim_{k\to+\infty} \mathbf{A}_{(1)}^{(k)} = \frac{1}{M}\mathbf{1}\mathbf{1}^T \triangleq \mathbf{J} \quad \textit{almost surely (a. s.)} .$$

*Proof.* We observe that $\mathbf{A}(k)$ is a stationary and ergodic sequence of stochastic matrices with strictly positive diagonal entries. Moreover, $\mathrm{E}[\mathbf{A}(k)]$ is an irreducible aperiodic matrix, then its eigenvalue with the second largest module has module strictly smaller than 1 ($|\lambda_2(\mathrm{E}[\mathbf{A}(k)])| < 1$). From Theorem 3 in [89], it follows that, with probability one, for each sequence $\mathbf{A}(k)$ it exists a vector $\mathbf{v} \in \mathbb{R}^M$, such that $\sum_i v_i = 1$ and

$$\lim_{k\to+\infty} \mathbf{A}_{(1)}^{(k)} = \mathbf{1}\mathbf{v}^T .$$

Note that in general $\mathbf{v}$ is a random variable, depending on the specific sequence $\{\mathbf{A}(k)\}_{k\geq 1}$. The matrices $\mathbf{A}(k)$ are doubly stochastic, then $\mathbf{w} = 1/M\mathbf{1}$ is a left eigenvector corresponding to the unit eigenvalue for all the matrices $\mathbf{A}(k)$. Theorem 4 in [89], guarantees that in this case the above vector $\mathbf{v}$ is a deterministic constant almost surely and in particular is equal to $\mathbf{w}$. This concludes the proof. □

Now we are ready to prove that the convergence rate is almost always exponential.

**Proposition 3.** *Under Assumption 1 on the mobility models, if the matrices are doubly stochastic, then for almost all the sequences there exist $C > 0$ and $0 < \beta < 1$ (with $C$ in general depending of the sequence) such that for $k \geq s$*

$$\left\| \mathbf{A}_{(s)}^{(k)} - \mathbf{J} \right\|_{\max} \leq C\beta^{k-s} .$$

*Proof.* Given a matrix $\mathbf{A}$, consider the coefficient of ergodicity, see e.g., [90]:

$$\tau_1(\mathbf{A}) = \frac{1}{2}\max_{i,j} \sum_{s=1}^{M} \left| [\mathbf{A}]_{is} - [\mathbf{A}]_{js} \right| .$$

In the proof of Theorem 3 in [89] is shown that it exists a positive natural $h$ and $\eta < 1$ such that

$$\mathbb{P}\left[ \tau_1\left( \mathbf{A}_{(s+(r-1)h)}^{(s+rh-1)} \right) < \eta \text{ for infinitely many r} \right] = 1 . \tag{3.6}$$

Then we decompose $\mathbf{A}_{(s)}^{(k)}$, in the product of $i_k$ blocks of size $h$ and one block of size $(k+1) \mod h$ as it follows:

$$\mathbf{A}_{(s)}^{(k)} = \mathbf{A}_{(s+i_k h)}^{(k)} \mathbf{A}_{(s+h(i_k-1))}^{(s+hi_k-1)} \cdots \mathbf{A}_{(s)}^{(s+h-1)} .$$

Because of the properties of a coefficient of ergodicity:

$$\tau_1\left(\mathbf{A}_{(s)}^{(k)}\right) \leq \tau_1\left(\mathbf{A}_{(s+i_k h)}^{(k)}\right) \prod_{j=1}^{i_k} \tau_1\left(\mathbf{A}_{(s+h(j-1))}^{(s+hj-1)}\right) \leq \prod_{j=1}^{i_k} \tau_1\left(\mathbf{A}_{(s+h(j-1))}^{(s+hj-1)}\right) .$$

Then, we can write:

$$\log\left(\tau_1\left(\mathbf{A}_{(s)}^{(k)}\right)\right) \leq \sum_{j=1}^{i_k} \log\left(\tau_1\left(\mathbf{A}_{(s+h(j-1))}^{(s+hj-1)}\right)\right) .$$

We now consider the Markov chain $\Phi$, that "generates" the sequence of matrices $\mathbf{A}(k)$ underlying the mobility process. Because of Lemma 1, $\Psi_t = (\Phi_t, \Phi_{t+1}, \cdots, \Phi_{t+h-1})$ is an irreducible aperiodic stationary Markov chain, and then all the powers of the shift operator $\theta$, and in particular $\theta^h$, are ergodic. We observe that $\log\left(\tau_1\left(\mathbf{A}_{(j)}^{(j+h-1)}\right)\right)$ is a function of $(\Phi_j, \Phi_{j+1}, \cdots, \Phi_{j+h-1})$ and then of $\Psi_j$. We call such function $f$, i.e.,

$$f(\Psi_j) \stackrel{\text{def}}{=} \log\left(\tau_1\left(\mathbf{A}_{(j)}^{(j+h-1)}\right)\right) .$$

Note that $f(\Psi_t) \leq 0$ and, from Equation (3.6), $f(\Psi_t) < \log(\eta) < 0$ infinitely often almost surely. To the random sequence $\{f(\Psi_s), f(\Psi_{s+h}), f(\Psi_{s+2h}), \cdots\}$, we can then apply the Birkhoff's Ergodic Theorem obtaining that:

$$\lim_{i\to\infty} \frac{1}{i} \sum_{j=1}^{i} \log\left(\tau_1\left(\mathbf{A}_{(s+h(j-1))}^{(s+hj-1)}\right)\right) = \lim_{i\to\infty} \frac{1}{i} \sum_{j=1}^{i} f(\Psi_{s+hj}) = \mathrm{E}[f(\Psi_t)] < 0 \quad \text{a. s. ,}$$

therefore,

$$\limsup_{h\to\infty} \frac{1}{h} \log\left(\tau_1\left(\mathbf{A}_{(s)}^{(s+h)}\right)\right) \leq \mathrm{E}[f(\Psi_t)] < 0 \quad \text{a. s. .}$$

Consider $\mathrm{E}[f(\Psi_t)] < \zeta < 0$, then for almost all the sequences it exists $h_0$, such that for all $h \geq h_0$, it holds:

$$\frac{1}{h} \log\left(\tau_1\left(\mathbf{A}_{(s)}^{(s+h)}\right)\right) \leq \zeta, \text{ i.e., } \tau_1\left(\mathbf{A}_{(s)}^{(s+h)}\right) \leq e^{\zeta h} .$$

If we define $\beta = \exp(\zeta) < 1$, $C = \beta^{-h_0}$ and recall that $\tau_1\left(\mathbf{A}_{(s)}^{(k)}\right) \leq 1$, we obtain:

$$\tau_1\left(\mathbf{A}_{(s)}^{(k)}\right) \leq C\beta^{k-s} \text{ for } k \geq s . \tag{3.7}$$

In the above equation, the value of the constant $h_0$ depends on the specific random sequence and also on $s$ (while the same value $\zeta$ can be selected for all the sequences and independently from $s$). We need then to use a corollary of the Ergodic Theorem about "nearly uniform" convergence that is stated as Proposition (1.5) in [91]: if $f(\cdot)$ is square integrable, then for almost all the sequences we can select $h_0$ independently from $s$. Clearly this is the

case for our function $f(\Psi_t)$, therefore we can conclude that $C$ in (3.7) only depends on the considered sequence.

So far we have established the existence of a geometric convergence result for the ergodic coefficient $\tau_1$. The last step of our proof requires us to prove a geometric bound for the distance between $\mathbf{A}_{(s)}^{(k)}$ and its almost sure limit $\mathbf{J}$. This is not particular difficult since we mainly have to follow the proof of Theorem 4.17 in [90].

First we observe that a geometric bound holds also for the difference between any two elements on the same column and different rows. In fact, from the definition of $\tau_1(\cdot)$:

$$\left| \left[\mathbf{A}_{(s)}^{(k)}\right]_{u,v} - \left[\mathbf{A}_{(s)}^{(k)}\right]_{w,v} \right| \leq 2\tau_1 \left(\mathbf{A}_{(s)}^{(k)}\right) .$$

We call $\epsilon$ the right side of the above expression that we can rewrite as:

$$\left[\mathbf{A}_{(s)}^{(k)}\right]_{u,v} - \epsilon \leq \left[\mathbf{A}_{(s)}^{(k)}\right]_{w,v} \leq \left[\mathbf{A}_{(s)}^{(k)}\right]_{u,v} + \epsilon ,$$

therefore, for the double stochasticity of $\mathbf{A}(k)$, for all $k$ we have that

$$\begin{aligned}
&\sum_{w=1}^{M} \left[\mathbf{A}(k+1)\right]_{z,w} \left( \left[\mathbf{A}_{(s)}^{(k)}\right]_{u,v} - \epsilon \right) \leq \\
&\leq \sum_{w=1}^{M} \left[\mathbf{A}(k+1)\right]_{z,w} \left[\mathbf{A}_{(s)}^{(k)}\right]_{w,v} \leq \\
&\leq \sum_{w=1}^{M} \left[\mathbf{A}(k+1)\right]_{z,w} \left( \left[\mathbf{A}_{(s)}^{(k)}\right]_{u,v} + \epsilon \right) ,
\end{aligned}$$

which is equal to

$$\left[\mathbf{A}_{(s)}^{(k)}\right]_{u,v} - \epsilon \leq \left[\mathbf{A}_{(s)}^{(k+1)}\right]_{z,v} \leq \left[\mathbf{A}_{(s)}^{(k+1)}\right]_{u,v} + \epsilon .$$

By induction:

$$\left[\mathbf{A}_{(s)}^{(k)}\right]_{u,v} - \epsilon \leq \left[\mathbf{A}_{(s)}^{(k+r)}\right]_{z,v} \leq \left[\mathbf{A}_{(s)}^{(k+1)}\right]_{u,v} + \epsilon ,$$

and letting $r$ go to infinity

$$\left[\mathbf{A}_{(s)}^{(k)}\right]_{u,v} - \epsilon \leq \frac{1}{M} \leq \left[\mathbf{A}_{(s)}^{(k+1)}\right]_{u,v} + \epsilon ,$$

i.e.,

$$\left| \left[\mathbf{A}_{(s)}^{(k)}\right]_{u,v} - \frac{1}{M} \right| \leq \epsilon ,$$

and being that this inequality is true for all $u$ and all $v$, a geometric bound can be derived also for $\left\| \mathbf{A}_{(s)}^{(k)} - \mathbf{J} \right\|_{\max}$. □

In [21] a different result is proven, i.e., that there exist $\hat{C}$ and $\hat{\beta}$ such that

$$\mathrm{E}\left[\left\|\mathbf{A}_{(s)}^{(k)} - \mathbf{J}\right\|_{\max}\right] \leq \hat{C}\hat{\beta}^{k-s} .$$

Then a series of inequalities for the expected values of $\|\mathbf{y}(k) - \mathbf{x}^i(k)\|_2$ are obtained for all $i$. Using Fatou's Lemma, along with the non-negativeness of distances, it is possible to derive inequalities that hold with probability 1. Using Proposition 3, instead, it is possible to obtain the same inequalities directly without the need to consider the expectation.

## 3.4 Asynchronous Updates

In this section, we study how the presented framework needs to be extended in order to support the case when nodes asynchronously update their status. We consider the sequence $\{t_k\}_{k\geq 1}$ of time instants at which one or more nodes perform an update of their estimates. Again, we denote that the estimate of node $i$ at time $t_k$ (immediately before the update) is $\mathbf{x}^i(k)$ and represent all the estimates through the matrix $\mathbf{X}(k)$. The evolution of the estimates can still be expressed in a matrix form similarly to (3.2):

$$\mathbf{X}(k+1) = \mathbf{A}(k)\mathbf{X}(k) - \mathbf{\Gamma}(k)\mathbf{D}(k) \,, \tag{3.8}$$

where $\mathbf{\Gamma}(k)$ is a diagonal matrix and the element $[\mathbf{\Gamma}(k)]_{ii}$ is simply the step-size used by node $i$ at the $k$-th update. We denote this step-size as $\gamma_i(k)$. If $j$ is not among the nodes which perform the update at time $t_k$, then it will simply be $a_{jj}(k) = 1$, $a_{jh}(k) = 0$ for $h \neq j$ and $\gamma_{jj}(k) = 0$.

In what follows we first consider the case of decreasing step-sizes, similarly to condition 1, described in Section 3.1. That is, we will consider that for each $i$, the sequence $\{\gamma_i(k)\}_{k\geq 1}$ satisfies: $\sum_{k=1}^{\infty}\gamma_i(k) = \infty$ and $\sum_{k=1}^{\infty}\gamma_i(k)^2 < \infty$.

We can go over the rationale in [21] and prove similar results for the new system description. In particular, our proof of the exponential convergence rate of $\mathbf{A}_{(s)}^{(k)}$ holds clearly also in this case. Bounds for the distance between $\mathbf{x}^i(k)$ and $\mathbf{y}(k) = 1/M\mathbf{1}^T\mathbf{X}(k)$ hold with minimal changes, so that we can prove the analogous of Proposition 2 in [21]:

**Proposition 4** (Convergence of Agent Estimates)**.** *Under Assumption 1, the estimate of each node converges almost surely to the vector* $\mathbf{y}(k)$*, i.e.,*

$$\lim_{k\to+\infty} \|\mathbf{y}(k) - \mathbf{x}^i(k)\|_{\ell_2} = 0 \quad \textit{a. s. , for all } i \,.$$

*Proof.* See Appendix 3.C. □

The following step is to use bounds for the distance between $\mathbf{y}(k)$ and $\mathbf{x}^*$ (a point of minimum for $f$) to show that $\lim_{k\to+\infty}\mathbf{y}(k) = \mathbf{x}^*$.

In particular the following inequality is derived in [21] (for the synchronous case they are considering):

$$\sum_{s=1}^{k}\gamma(s)\left[f(\mathbf{y}(s)) - f(\mathbf{x}^*)\right] \leq \frac{M}{2}\|\mathbf{y}(1) - \mathbf{x}^*\|_{\ell_2}^2 + \\ + 2L\sum_{j=1}^{M}\sum_{s=1}^{k}\gamma(s)\|\mathbf{y}(s) - \mathbf{x}^j(s)\|_{\ell_2} + \frac{L^2}{2}\sum_{s=1}^{k}\gamma^2(s) \,.$$

The first term at the right hand side of the above inequality is a constant, the last term is summable because of the assumption on the step-sizes. In [21] it is proven that, almost surely, $\sum_{s=1}^{\infty} \gamma(s)\|\mathbf{y}(s) - \mathbf{x}^j(s)\|_{\ell_2} < \infty$. Thus they show that

$$0 \leq \sum_{s=1}^{\infty} \gamma(s) \left[f(\mathbf{y}(s)) - f(\mathbf{x}^*)\right] < \infty \quad \text{a. s. ,} \tag{3.9}$$

from (3.9) and the fact that $\sum_{s=1}^{\infty} \gamma(s) = \infty$, it is possible to conclude that

$$\liminf_{k\to\infty} f(\mathbf{y}(k)) = f(\mathbf{x}^*) \quad \text{a. s. , and } \lim_{k\to\infty} \mathbf{x}^i(k) = \mathbf{x}^* \text{ a. s. .}$$

In the Appendix 3.D, a similar derivation is carried on, leading to the following generalization of (3.9):

$$\sum_{s=1}^{\infty} \sum_{i=1}^{M} \gamma_i(s) \left[f_i(\mathbf{y}(s)) - f_i(\mathbf{x}^*)\right] < \infty \quad \text{a. s. .} \tag{3.10}$$

Unfortunately, the different values of $\gamma_i(s)$ do not allow us to formulate the inequality above in terms of the global function $f$ as in (3.9).

We do not have currently a formal result stating under which conditions the asynchronous system converges to the optimal solution, but (3.10) suggests us that all the weights $\gamma_i(k)$ should have on average the same value. We then propose the following conjecture, that we support later with some examples:

**Conjecture 1.** *When updates are asynchronous, convergence results for sub-gradient methods hold if* $E[\gamma_i(k)] = E[\gamma_j(k)]$ *for each* $i$ *and* $j$.

Let us see how we can guarantee this condition in different cases. We consider that updates occur after every meeting following rule 2 in Section 3.2. Moreover consider that $\gamma_i(k) = 1/n_i(k)$, where $n_i(k)$ is the total number of updates node $i$ has performed until the time instant $t_k$. If the meeting process follows a Poisson process with total rate $\lambda$ and at each instant the probability that node $i$ meets another node is $p_i$, we expect that by time $k$, node $i$ has $p_i k$ meetings (and an equal number of updates). Then the expected value of its step-size is $\mathrm{E}[\gamma_i(k)] = \mathrm{E}[1/n_i(k)] = p_i/(p_i k) = 1/k$. In conclusion if step-sizes follow the rule $\gamma_i(k) = 1/n_i(k)$, we expect the asynchronous sub-gradient mechanism to converge to the optimal solution. Figure 3.3 (top graph) shows that this is true for our toy example. The simulations for the optimization problem considered in Section 3.5 confirm such convergence.

Let us now revisit the example in Section 3.2 showing that the estimates were not converging to a point of minimum (Figure 3.2, bottom graph). Here step-sizes were constant,

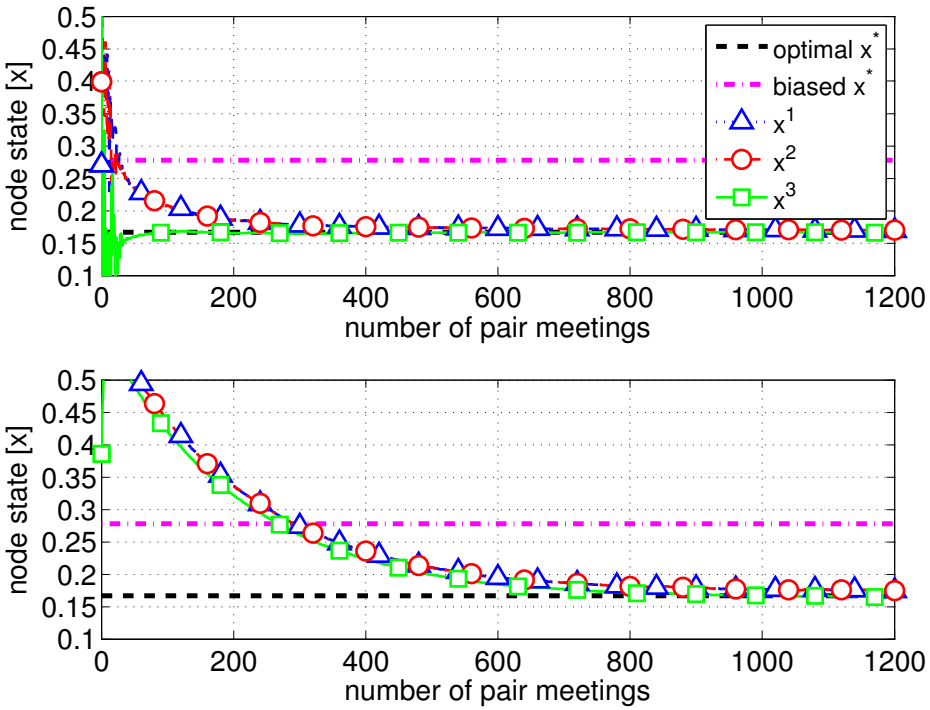



**Figure 3.3.** *Toy example, convergence of the three estimates in case of asynchronous updates. Results have been averaged over* 100 *simulation runs. Top graph: decreasing step-size* $\gamma_i(k) = 1/n_i(k)$*. Bottom graph: weighted fixed step-size* $\gamma_i(k) = p_i^{-1} \cdot 25 \cdot 10^{-4}$*.*

i.e. $\gamma_i(k) = \gamma$. Now, reasoning as above we can conclude that $\mathrm{E}[\gamma_i(k)] = p_i\gamma$. Hence the expected values are not equal as far as node meeting rates (and then update rates) are not equal: this was the case of our example, where[4] $p_1 = 5/6$, $p_2 = 5/6$ and $p_3 = 1/3$. Intuitively, we expect convergence to be biased towards values closer to the optimum of the local functions of those nodes that perform the updates more often. Equation (3.10) suggests us that what the distributed mechanism was really doing is to minimize the function $\sum_i p_i f_i = (3/2)x^2 - (5/6)x$ rather then $f = \sum_i f_i = 3x^2 - x$. This is the case, being that the estimates are converging to $5/18$ (dot-dashed line in all the previous figures). If now we want to correct the bias, it is sufficient to consider that each node selects its step-size inversely proportional to its meeting rate. Figure 3.3 (bottom graph) shows that also this correction leads the estimates to converge to the correct results.

[4]Note that meetings always involve two nodes, this is the reason why $p_1 + p_2 + p_3 = 2$.

## 3.5 Application in DTNs: a Case Study

In this section we apply the distributed sub-gradient method with our enhancements to a DTN scenario inspired by the work in [83]. As explained in Section 3.4 our enhancements consist of: 1) allowing nodes to update asynchronously their estimates, i.e., whenever any two of them meet and 2) applying the decreasing step size rule to avoid possible bias effects in the convergence towards the global optimum.

All the nodes in the network are interested in the same dynamic information content and can share it whenever they meet. The information update is performed by a Service Provider (SP) that injects fresh information in the network according to a Poisson process of parameter $\mu$ update/sec. At a given instant $\bar{t}$ we call $t_i(\bar{t})$ the time at which the SP generated the most recent content version available at node $i$, then $Y_i(\bar{t}) = \bar{t} - t_i(\bar{t})$ is the age of such version. An information content has a non-increasing value in time. For example, we give value $u_i(Y_i(\bar{t}))$ to the information stored in node $i$, where $u_i(\cdot)$ is a non-increasing function. The goal of the SP is to optimize

$$f(\mathbf{x}) = \sum_{i=1}^{M} f_i(\mathbf{x}) = \sum_{i=1}^{M} E_{\mathbf{x}}[u_i(Y_i)] \,, \tag{3.11}$$

where $\mathbf{x} \in \mathbb{R}^M$ is the rate allocation vector, such that $\sum_{i=1}^{M} x_i \leq \mu$ and $x_i \geq 0$ for all $i$. Note that the age $Y_i(\bar{t})$ is modeled as a random variable $Y_i$ depending on $\mathbf{x}$. In [83], Equation (3.11) is proved to be concave and therefore the optimal $\mathbf{x}$ can be obtained by the SP using standard optimization techniques (see e.g., [92]) such as the projected gradient descent algorithm[5]: namely, iteratively computing $\mathbf{x}(k+1) = \Pi\left(\mathbf{x}(k) + \gamma_k \nabla f(\mathbf{x}(k))\right)$, where $\{\gamma_k\}_{k\geq 0}$ is a positive sequence of parameters such that $\sum_k \gamma_k = \infty$, $\lim_{k\to\infty} \gamma_k = 0$ and $\Pi$ is the projection onto the feasible set for $\mathbf{x}$. In general, a closed formula for $f(\mathbf{x})$ is not known; thus, the gradient needs to be estimated as explained in [83]. Here, our purpose is to focus on the distributed sub-gradient method so we consider the specific case where updates can travel at most two hops, thus avoiding to address the gradient estimation's issue. In detail, at a given instant $\bar{t}$, let us call $t_j^{SP}(\bar{t})$ the time at which the SP directly injected fresh content to node $j$, then we define the following protocol's rule:

**Definition 1** (Content Sharing)**.** When a node $j$ meets a node $i$ at time $\bar{t}$, $j$ will copy to user $i$ the last content downloaded directly from the SP if this content is more recent then the content stored in $i$, i.e., $t_i(\bar{t}) < t_j^{SP}(\bar{t})$ .

[5]For a discussion about the projected gradient method implemented in a distribute fashion see [93].

In this case, assuming also that (a) $u(Y_i) = \chi\{Y_i \leq \tau\}$ for all nodes $i$, where $\tau$ is a given threshold after which the information is worthless[6] (e.g., the information consists of news about events that expire after some time) and (b) the meeting process among node pairs is Poisson distributed, we can compute the local utility function for each node as

$$f_i(\mathbf{x}) = 1 - \left[\prod_{j \in \mathcal{N}_i} \frac{x_j e^{-\lambda_{ij}\tau} - \lambda_{ij} e^{-x_j \tau}}{x_j - \lambda_{ij}}\right] e^{-x_i \tau} , \tag{3.12}$$

where $\lambda_{ij}$ is the meeting rate between $i$ and $j$ and $\mathcal{N}_i \stackrel{\text{def}}{=} \{j : \lambda_{ij} > 0\}$. The global utility function in (3.11) is then simply obtained summing (3.12) over $i = 1, 2, \ldots, M$. The computations to derive (3.12) can be found in the Appendix 3.E.

The local gradient function needed in (3.1) can be computed directly from (3.12), where nodes only need to estimate simple statistics on their own meeting rates. Clearly, (3.11) can be optimized also in a centralized fashion collecting, for example at the SP itself, information about the statistic of the overall network meeting process [83]. However, in a DTN scenario the SP may be able to communicate with a group of connected nodes only for short periods of time, that we would like to exploit transmitting the actual content users care about. Moreover, issues related to privacy easily apply to this scenario: for example, a node may prefer not to disclose information about its meetings to the SP and, in some cases, it would be equally desirable to maintain information about the utility function $u_i(\cdot)$ reserved (e.g., in military applications). A distributed approach is therefore of actual interest not only for DTNs, but also for scenarios that go beyond them.

To optimize $f(\mathbf{x})$ in a distributed fashion we can use the framework presented in Section 3.1. Each node $i$ can compute a local estimate of the optimal allocation $\mathbf{x}$, i.e., $\mathbf{x}^i$, through iterative updates. In detail, when two nodes $i$ and $j$ meet they: i) update $\mathbf{x}^i$ and $\mathbf{x}^j$ as in (3.1) and, ii) project the result so obtained onto the feasible set $\sum_{l=1}^{M} x_l \leq \mu$ and $x_l \geq 0$ for all $l \in \{1, \ldots, M\}$.

In our implementation of sub-gradient optimization all the $\mathbf{x}^i$ eventually converge to the optimum $\mathbf{x}^*$ of (3.11). Henceforth, the SP can retrieve the optimal transmission rates using the following "push-policy". During the execution of the algorithm each node $i$ maintains its own estimate $\mathbf{x}^i$. The SP collects $\mathbf{x}^i$ from every node and obtains the rate allocation vector as $\mathbf{x} = (\sum_{i=1}^{M} \mathbf{x}^i)/M$.

To test the performance achievable by the distributed sub-gradient method under traces with memory, we simulated meeting events among $M = 10$ nodes as follows. Calling $R_1$

[6] $\chi\{y \leq \tau\}$ is equal to 1 if $y \leq \tau$ and 0 otherwise.

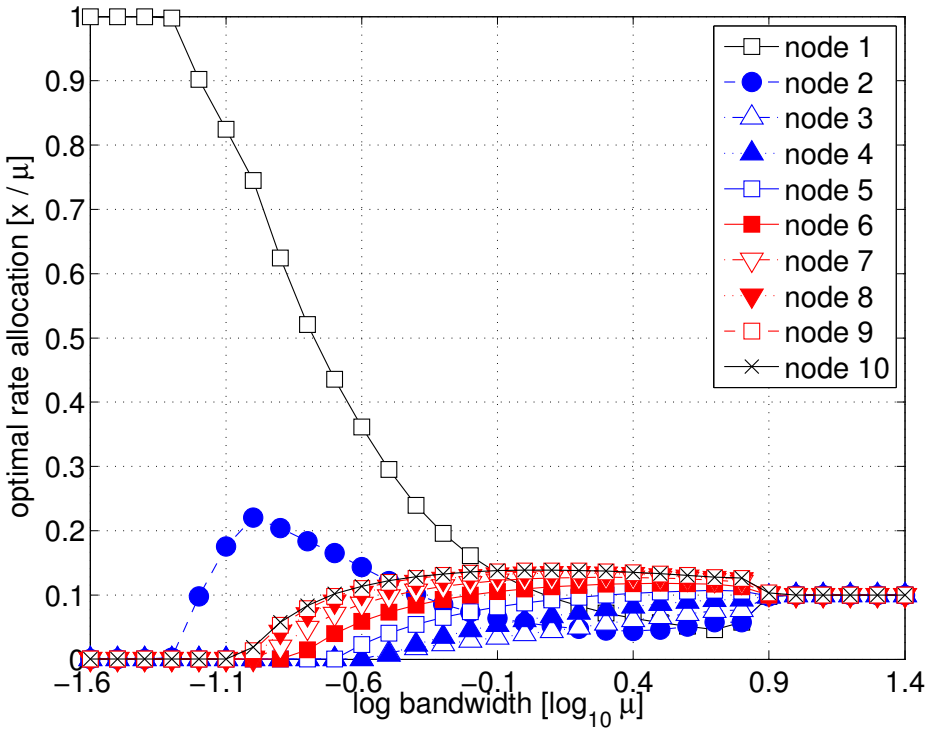



**Figure 3.4.** *Optimal bandwidth allocation for a network of* 10 *nodes.*

and $R_2$ two distinct regions of the space, nodes can be placed either in $R_1$ or in $R_2$. Only nodes that are within the same region can communicate with each other. We let nodes free to change region of placement according to a Poisson process of overall rate $\lambda_d = 0.1$, thus network's full connectivity is guaranteed. In addition, according to a Poisson process of parameter $\lambda_m = 1$ (note that $\lambda_m > \lambda_d$), we generate meeting events among pair of nodes belonging to the same region. Each node is selected for a meeting according to a weight which is proportionally inverse to its index, i.e., node $i$ is selected with weight $w_i = i^{-3}$. Note that we generate a meeting process that is both stationary and ergodic, and along this process nodes have diverse contact rates. In particular node 1 has the highest contact rate, whilst node 10 the lowest. To sum up, letting nodes $i$ and $j$ update their states at their meeting times according to the above process, we obtain a corresponding sequence $\{\mathbf{A}(k)\}_{k\geq 1}$ that can be viewed as generated from an ergodic and stationary Markov chain. Also, given that nodes have different contact rates and asynchronous updates are performed, we know that a direct application of the distributed sub-gradient algorithm may lead to sub-optimal results, see the toy example of Section 3.2.

Figures 3.4–3.5 show simulation results for the above setting of parameters and $\tau = 20$.

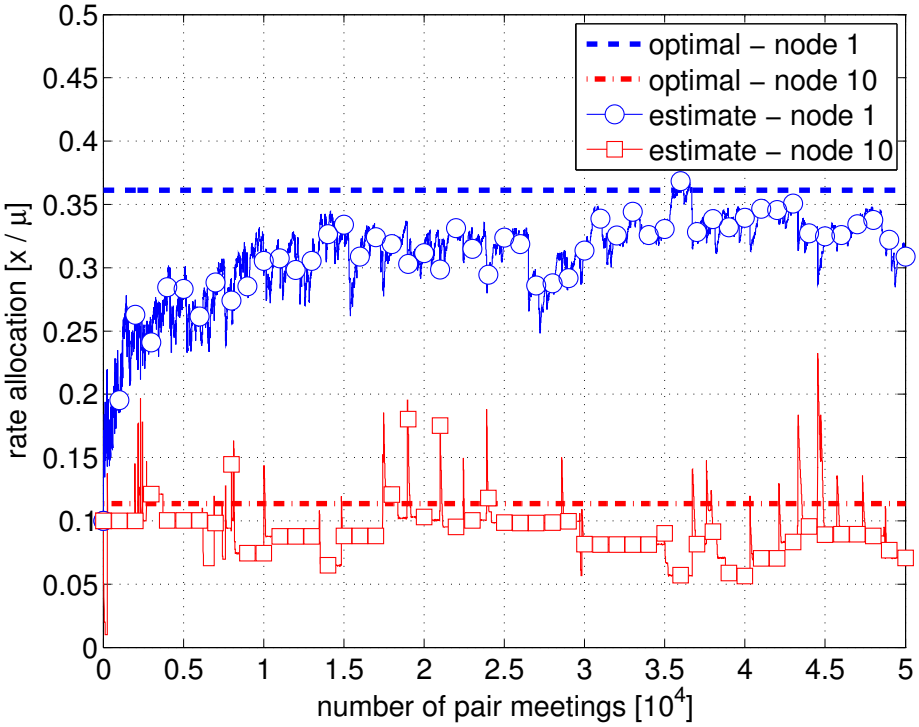



**Figure 3.5.** *Example of convergence of estimate for two nodes when the sub-gradient method is used. Asynchronous updates and decreasing step-size.*

Figure 3.4 shows the optimal allocation rate for each user. When the bandwidth that the SP can use to send updates is very low (i.e., small $\mu$), the best solution is that the SP uniquely sends updates to the node that has the higher contact rate, i.e., node 1; for large values of $\mu$, instead, the SP can evenly send updates to all the nodes in the network. Interestingly, as already observed in [83], for some values of $\mu$ (in our case $\mu$ around $10^{0.7}$ update/sec) the optimal choice is for the SP to allocate more bandwidth (i.e., a larger fraction of $\mu$) to the node with the lowest contact rate, namely, node 10. For these values of $\mu$, in fact, those nodes with a large contact rate such as node 1 are able to maintain high values for their utility functions just by collecting information from the large number of nodes they meet.

In Figure 3.5 we show the mean trajectory towards the optimal for two elements in $\mathbf{x} = (\sum_{i=1}^{M} \mathbf{x}^i)/M$, where the vectors $\mathbf{x}^i$ have been obtained along a sequence of $5 \cdot 10^4$ meetings considering $\mu = 10^{-1.1}$ update/sec. We note that the estimates provided through the distributed sub-gradient method converge to the theoretical optimal allocation values in Figure 3.4. Concerns about the convergence rate of such estimates are out of the scope of the present report and will be addressed in the future research[7].

[7]Note, however, that a wide literature addressing this issue already exists. E.g., for the problem of designing

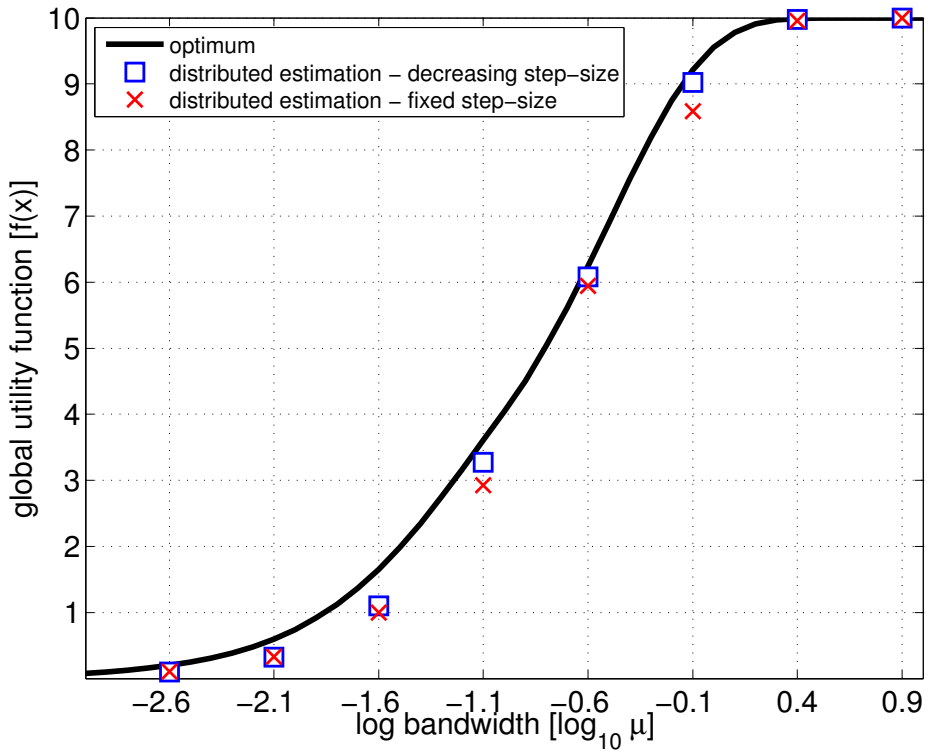



**Figure 3.6.** *Performance comparison: centralized solver vs distributed method.*

Finally, in Figure 3.6 we draw with a solid-line the maximum of $f(\mathbf{x})$ corresponding to the optimal rate allocations in Figure 3.4, which was obtained using a centralized solver [92]. Note that with unlimited bandwidth, the maximum of $f(\mathbf{x})$ is equal to 10, i.e., when each node $i$ has utility $u_i = 1$. For eight different values of the available bandwidth $\mu$, we also plot with squared points the utility function values corresponding to the optimal rate allocation achieved again with $\mathbf{x} = (\sum_{i=1}^{M} \mathbf{x}^i)/M$ using sub-gradient optimization together with our enhancements. With crosses we show the performance of the sub-gradient optimization with a fixed step size [20], which neglects the asynchronous update issue. As expected, the results of the latter algorithm are sub-optimal. Most importantly, the solutions achieved with our approach are very close to the actual optimum for all values of $\mu$. This confirms the validity of the distributed framework that we presented.

suitable sequences $\{\mathbf{A}(k)\}$ to speed up the convergence of consensus see [94].

## 3.6 Conclusions and Discussions

In this chapter we considered the recent framework of the distributed sub-gradient optimization proposed in [20], and later extended in [21] for application on random scenarios. We pointed out that existing convergence results for this framework can be applied to DTNs only in the case of synchronous node operation and in the presence of simple mobility models without memory. Therefore, we addressed both these issues.

First, we proved convergence to optimality of the sub-gradient optimization technique under a more general class of mobility processes formally defined using a Markovian mobility model with memory in the meeting process. This result, in particular, must be regarded as the an original contribution of this work.

Second, we proposed some modifications to the original sub-gradient algorithm so as to avoid bias problems (i.e., consisting of the convergence towards sub-optimal solutions) when nodes operate asynchronously. Also this analysis has to be regarded an important contribution, especially because it can be used for the distributed optimization of practical network protocols. In fact, as a case study, we applied the presented framework to the optimization of the dissemination of dynamic content in a DTN.

All the provided results confirmed that the distributed sub-gradient method is an effective and very promising tool for optimization in distributed contexts.

## 3.A Derivation of Equation (3.2)

In this appendix we show the mathematical steps that lead from Equation (3.2) to Equation (3.3).

From (3.2) we have that

$$\begin{aligned}\mathbf{X}(k+2) &= \mathbf{A}(k+1)\mathbf{X}(k+1) - \mathbf{\Gamma}(k+1)\mathbf{D}(k+1) = \\ &= \mathbf{A}(k+1)[\mathbf{A}(k)\mathbf{X}(k) - \mathbf{\Gamma}(k)\mathbf{D}(k)] - \mathbf{\Gamma}(k+1)\mathbf{D}(k+1) = \\ &= \mathbf{A}_{(k)}^{(k+1)}\mathbf{X}(k) - \mathbf{A}_{(k+1)}^{(k+1)}\mathbf{\Gamma}(k)\mathbf{D}(k) - \mathbf{\Gamma}(k+1)\mathbf{D}(k+1)\ ,\end{aligned}$$

where $\mathbf{\Gamma}(k)$ is a diagonal matrix and the element $[\mathbf{\Gamma}(k)]_{ii} = \gamma(k)$, for all $i$. Iteratively, replacing $k+2$ with $k+s+1$, $k+1$ with $k+s$ and $k$ with $k+s-1$, respectively, leads to

$$\begin{aligned}&\mathbf{X}(k+s+1) = \\ &= \mathbf{A}_{(k+s-1)}^{(k+s)}\mathbf{X}(k+s-1) - \mathbf{A}_{(k+s)}^{(k+s)}\mathbf{\Gamma}(k+s-1)\mathbf{D}(k+s-1) - \mathbf{\Gamma}(k+s)\mathbf{D}(k+s) = \\ &= \mathbf{A}_{(k+s-1)}^{(k+s)}\left[\mathbf{A}(k+s-2)\mathbf{X}(k+s-2) - \mathbf{\Gamma}(k+s-2)\mathbf{D}(k+s-2)\right] + \\ &\quad - \mathbf{A}_{(k+s)}^{(k+s)}\mathbf{\Gamma}(k+s-1)\mathbf{D}(k+s-1) - \mathbf{\Gamma}(k+s)\mathbf{D}(k+s) = \\ &= \mathbf{A}_{(k+s-2)}^{(k+s)}\mathbf{X}(k+s-2) - \mathbf{A}_{(k+s-1)}^{(k+s)}\mathbf{\Gamma}(k+s-2)\mathbf{D}(k+s-2) + \\ &\quad - \mathbf{A}_{(k+s)}^{(k+s)}\mathbf{\Gamma}(k+s-1)\mathbf{D}(k+s-1) - \mathbf{\Gamma}(k+s)\mathbf{D}(k+s) = \\ &= \mathbf{A}_{(k+s-2)}^{(k+s)}\mathbf{X}(k+s-2) - \sum_{l=0}^{1}\mathbf{A}_{(k+s-l)}^{(k+s)}\mathbf{\Gamma}(k+s-1-l)\mathbf{D}(k+s-1-l) + \\ &\quad - \mathbf{\Gamma}(k+s)\mathbf{D}(k+s) = \\ &\qquad\vdots \\ &= \mathbf{A}_{(1)}^{(k+s)}\mathbf{X}(1) - \sum_{l=0}^{k+s-2}\mathbf{A}_{(k+s-l)}^{(k+s)}\mathbf{\Gamma}(k+s-1-l)\mathbf{D}(k+s-1-l) - \mathbf{\Gamma}(k+s)\mathbf{D}(k+s)\ ,\end{aligned}$$

and finally, replacing $k+s$ with $k$, we have that

$$\mathbf{X}(k+1) = \mathbf{A}_{(1)}^{(k)}\mathbf{X}(1) - \sum_{l=0}^{k-2}\mathbf{A}_{(k-l)}^{(k)}\mathbf{\Gamma}(k-1-l)\mathbf{D}(k-1-l) - \mathbf{\Gamma}(k)\mathbf{D}(k) \tag{3.13}$$

or, replacing in Equation (3.13) $k-l$ with $s$

$$\mathbf{X}(k+1) = \mathbf{A}_{(1)}^{(k)}\mathbf{X}(1) - \sum_{s=2}^{k}\mathbf{A}_{(s)}^{(k)}\mathbf{\Gamma}(s-1)\mathbf{D}(s-1) - \mathbf{\Gamma}(k)\mathbf{D}(k)\ ,$$

that is exactly Equation (3.3), as we wanted.

## 3.B Stationarity and Ergodicity: Concepts

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability spaces. Let also the function $T : \Omega \to \Omega$ be a measurable transformation of $\Omega$ into itself.

**Definition 2** (Measure-Preserving)**.** A measurable transformation $T : \Omega \to \Omega$ into $\Omega$ is measure-preserving if, for every $B$ in $\mathcal{F}$, $\mathbb{P}(TB) = \mathbb{P}(B)$.

**Definition 3** (Stationarity)**.** A random sequence $\omega \in \Omega$ is stationary if the shift operator is measure preserving.

**Definition 4** (Invariant set)**.** If $T$ is a measure-preserving transformation, $B \in \mathcal{F}$ is an invariant set if $TB = B$, or equivalently $\mathbb{P}\left[(B \setminus TB) \cup (TB \setminus B)\right] = 0$.

**Definition 5** (Ergodicity)**.** A measure-preserving transformation $T$ is ergodic, if, given any invariant set $B \in \mathcal{F}$, it holds $\mathbb{P}(B) = 0$ or $\mathbb{P}(B) = 1$.

**Definition 6** (Mixing)**.** A measure-preserving transformation $T$ is mixing if, for all $B$ and $C$ in $\mathcal{F}$,

$$\lim_{n\to\infty} \mathbb{P}(B \cap T^n C) = \mathbb{P}(B)\mathbb{P}(C).$$

Note that when a random sequence is said to be ergodic tout court, it means that the shift operator is ergodic. Similarly when a random sequence is said to be mixing (or mixing in the ergodic-theoretic sense), it means that the shift operator is mixing.

## 3.C Proof of Proposition 4

In this appendix we prove Proposition 4, reported in the following for reader convenience. Since we proved Proposition 3, the result here presented (along with Proposition 5 in the Appendix 3.D) can be viewed as an easy extension of Theorem 1 in [21] to the case of mobility process obtained according to Assumption 1.

Let us first to explicitly formalize the following assumption, according to both [20] and [21]:

**Assumption 2** (Bounded Sub-gradients)**.** *Given $\mathbf{x} \in \mathbb{R}^N$, consider the sub-gradients of all the nodes in the network computed in $\mathbf{x}$, i.e., $\mathbf{d}^1, \ldots, \mathbf{d}^i, \ldots, \mathbf{d}^M$. There exists a scalar $L$ such that $\|\mathbf{d}^i\|_{\ell_2} \leq L$ for any $\mathbf{x} \in \mathbb{R}^N$ and for all $i \in \{1, \ldots, M\}$. Namely, the sub-gradients of all nodes in the network are bounded.*

Then, we recall the following lemma from [93]:

**Lemma 2.** *Let $0 < \beta < 1$ and let $\{\alpha(k)\}_{k \geq 0}$ be a positive scalar sequence. Assume that $\lim_{k \to +\infty} \alpha(k) = 0$. Then*

$$\lim_{k \to +\infty} \sum_{l=0}^{k} \beta^{k-l} \alpha(l) = 0 \, .$$

*In addition, if $\sum_{k=1}^{\infty} \alpha(k) < \infty$, then*

$$\sum_{k=1}^{\infty} \sum_{l=0}^{k} \beta^{k-l} \alpha(l) < \infty \, .$$

Using Lemma 2 we can prove that

**Proposition** (Convergence of Agent Estimates)**.** *Under Assumption 1 and decreasing step-size rule (see Section 3.4), the estimate of each node converges almost surely to the vector $\mathbf{y}(k)$, i.e.*

$$\lim_{k \to +\infty} \|\mathbf{y}(k) - \mathbf{x}^i(k)\|_{\ell_2} = 0 \quad \textit{a. s. , for all } i \, .$$

*Proof.* Iterating Equation (3.8), and considering only the $i-$th row of $\mathbf{X}(k)$, we obtain

$$\mathbf{x}^i(k) = \sum_{j=1}^{M} \left[ \mathbf{A}_{(1)}^{(k-1)} \right]_{ij} \mathbf{x}^j(1) - \sum_{s=1}^{k-2} \sum_{j=1}^{M} \left[ \mathbf{A}_{(s+1)}^{(k-1)} \right]_{ij} \gamma_j(s) \mathbf{d}^j(s) - \gamma_i(k-1) \mathbf{d}^i(k-1) \, . \quad (3.14)$$

Recalling that $\mathbf{y}(k) \stackrel{\text{def}}{=} 1/M \mathbf{1}^T \mathbf{X}(k)$, we can write

$$\mathbf{y}(k+1) = \mathbf{y}(k) - \frac{1}{M} \sum_{j=1}^{M} \gamma_j(k) \mathbf{d}^j(k) \quad (3.15)$$

and iteratively

$$\mathbf{y}(k) = \frac{1}{M}\sum_{j=1}^{M}\mathbf{x}^j(1) - \frac{1}{M}\sum_{s=1}^{k-1}\sum_{j=1}^{M}\gamma_j(s)\mathbf{d}^j(s)\,. \tag{3.16}$$

From (3.14) and (3.16) we have

$$\begin{aligned}
&\|\mathbf{y}(k) - \mathbf{x}^i(k)\|_{\ell_2} = \\
&= \Bigg\| \frac{1}{M}\sum_{j=1}^{M}\mathbf{x}^j(1) - \sum_{j=1}^{M}\left[\mathbf{A}_{(1)}^{(k-1)}\right]_{ij}\mathbf{x}^j(1) - \frac{1}{M}\sum_{s=1}^{k-1}\sum_{j=1}^{M}\gamma_j(s)\mathbf{d}^j(s) + \\
&+ \sum_{s=1}^{k-2}\sum_{j=1}^{M}\left[\mathbf{A}_{(s+1)}^{(k-1)}\right]_{ij}\gamma_j(s)\mathbf{d}^j(s) + \gamma_i(k-1)\mathbf{d}^i(k-1)\Bigg\|_{\ell_2} \leq \\
&\leq \sum_{j=1}^{M}\|\mathbf{x}^j(1)\|_{\ell_2}\left\|\frac{1}{M} - \left[\mathbf{A}_{(1)}^{(k-1)}\right]_{ij}\right\|_{\ell_2} + \\
&+ \sum_{s=1}^{k-2}\sum_{j=1}^{M}\|\gamma_j(s)\|_{\ell_2}\left\|\frac{1}{M} - \left[\mathbf{A}_{(s+1)}^{(k-1)}\right]_{ij}\right\|_{\ell_2}\|\mathbf{d}^j(s)\|_{\ell_2} + \\
&+ \frac{1}{M}\sum_{j=1}^{M}\|\gamma_j(k-1)\|_{\ell_2}\|\mathbf{d}^j(k-1)\|_{\ell_2} + \|\gamma_i(k-1)\|_{\ell_2}\|\mathbf{d}^i(k-1)\|_{\ell_2} \leq \\
&\leq \sum_{j=1}^{M}\|\mathbf{x}^j(1)\|_{\ell_2}b(k-1,1) + L\sum_{j=1}^{M}\sum_{s=1}^{k-2}\gamma_j(s)b(k-1,s+1) + \\
&+ L\left(\frac{1}{M}\sum_{j=1}^{M}\gamma_j(k-1) + \gamma_i(k-1)\right)\,,
\end{aligned} \tag{3.17}$$

where the last inequality follows from the bounded sub-gradient Assumption 2 and defining the quantity

$$b(k,s) \stackrel{\text{def}}{=} \max_{i,j}\left|\left[\mathbf{A}_{(s)}^{(k)}\right]_{ij} - \frac{1}{M}\right| \quad \text{for all } k \geq s\,.$$

Immediately we note that the last term in the right hand side (rhs) of Equation (3.17) goes to zero as $k$ goes to infinity, since by assumption $\lim_{k\to\infty}\gamma_j(k)$, for all $j$. From the proof of Proposition 3 we have that for almost all the sequences $\{\mathbf{A}(k)\}_{k\geq 1}$

$$b(k,s) \leq C\beta^{k-s} \quad \text{for all } k \geq s\,, \tag{3.18}$$

where $C > 0$ and $0 < \beta < 1$ (with $C$ in general depending of the considered sequence). Therefore for almost all the sequences also the first term in the rhs of (3.17) goes to zero increasing $k$. Finally, we if (3.18) holds, Lemma 2 applies and $\lim_{k\to+\infty}\sum_{s=1}^{k-2}\gamma_j(s)b(k-1,s+1) = 0$ for all $j$. Thus, for almost all the sequences, for all $i$ we have that

$$0 \leq \lim_{k\to+\infty}\|\mathbf{y}(k) - \mathbf{x}^i(k)\|_{\ell_2} \leq 0\,,$$

proving the desired result. □

## 3.D Derivation of Equation (3.10)

In this appendix we derive Equation (3.10). To this end, let us consider the following lemma, that is a generalization of Lemma 5 in [20].

**Lemma 3** (Basic Iterate Relation). *Let $\mathbf{x}^i(k)$ be generated according to (3.14) for all $i \in \{1,\dots,M\}$, $k \geq 1$ and $\mathbf{y}(k)$ be generated according to (3.15) for all $k \geq 1$. Let also $\{\mathbf{g}^i(k)\}_{k\geq 1}$ be a sequence of sub-gradient of $f_i(\cdot)$ computed in $\mathbf{y}(k)$, for all $i \in \{1,\dots,M\}$, then for any $\mathbf{x} \in \mathbb{R}^N$ and $k \geq 1$ we have*

$$\begin{aligned}\|\mathbf{y}(k+1)-\mathbf{x}\|_{\ell_2}^2 \leq \|\mathbf{y}(k)-\mathbf{x}\|_{\ell_2}^2 + \\ + \frac{2}{M}\sum_{j=1}^{M}\gamma_j(k)\left[\left(\|\mathbf{d}^j(k)\|_{\ell_2}+\|\mathbf{g}^j(k)\|_{\ell_2}\right)\|\mathbf{y}(k)-\mathbf{x}^j(k)\|_{\ell_2}\right] + \\ - \frac{2}{M}\sum_{j=1}^{M}\gamma_j(k)\left[f_j(\mathbf{y}(k))-f_j(\mathbf{x})\right] + \frac{1}{M^2}\sum_{j=1}^{M}\gamma_j^2(k)\|\mathbf{d}^j(k)\|_{\ell_2}^2 .\end{aligned}$$

*Proof.* It follows straightforwardly from the same rationale of Lemma 5 in [20]. Considering Equation (3.15), we can write, for any $\mathbf{x} \in \mathbb{R}^N$ and all $k \geq 1$

$$\|\mathbf{y}(k+1)-\mathbf{x}\|_{\ell_2}^2 = \left\|\mathbf{y}(k)-\frac{1}{M}\sum_{j=1}^{M}\gamma_j(k)\mathbf{d}^j(k)-\mathbf{x}\right\|_{\ell_2}^2 ,$$

implying that

$$\begin{aligned}\|\mathbf{y}(k+1)-\mathbf{x}\|_{\ell_2}^2 \leq \|\mathbf{y}(k)-\mathbf{x}\| - \frac{2}{M}\sum_{j=1}^{M}\gamma_j(k)\left\{\mathbf{d}^j(k)\right\}^T(\mathbf{y}(k)-\mathbf{x}) + \\ + \frac{1}{M^2}\sum_{j=1}^{M}\gamma_j^2(k)\|\mathbf{d}^j(k)\|_{\ell_2}^2 . \qquad (3.19)\end{aligned}$$

Considering the term $\left\{\mathbf{d}^j(k)\right\}^T(\mathbf{y}(k)-\mathbf{x})$, for any $j$, we have

$$\begin{aligned}\left\{\mathbf{d}^j(k)\right\}^T(\mathbf{y}(k)-\mathbf{x}) &= \left\{\mathbf{d}^j(k)\right\}^T(\mathbf{y}(k)-\mathbf{x}^j(k)) + \left\{\mathbf{d}^j(k)\right\}^T(\mathbf{x}^j(k)-\mathbf{x}) \geq \\ &\geq -\|\mathbf{d}^j(k)\|_{\ell_2}\|\mathbf{y}(k)-\mathbf{x}^j(k)\|_{\ell_2} + \left\{\mathbf{d}^j(k)\right\}^T(\mathbf{x}^j(k)-\mathbf{x}) .\end{aligned}$$

Since $\mathbf{d}^j(k)$ is a sub-gradient of $f_j$ at $\mathbf{x}^j(k)$, we also have for any $j$ and any $\mathbf{x} \in \mathbb{R}^N$,

$$\left\{\mathbf{d}^j(k)\right\}^T(\mathbf{x}^j(k)-\mathbf{x}) \geq f_j(\mathbf{x}^j(k)) - f_j(\mathbf{x}) .$$

Moreover, by using a sub-gradient $\mathbf{g}_j(k)$ of $f_j$ at $\mathbf{y}(k)$, we obtain for any $j$ and any $\mathbf{x} \in \mathbb{R}^N$,

$$\begin{aligned}f_j(\mathbf{x}^j(k)) - f_j(\mathbf{x}) &= f_j(\mathbf{x}^j(k)) - f_j(\mathbf{y}(k)) + f_j(\mathbf{y}(k)) - f_j(\mathbf{x}) \geq \\ &\geq \{\mathbf{g}_j(k)\}^T(\mathbf{x}^j(k)-\mathbf{y}(k)) + f_j(\mathbf{y}(k)) - f_j(\mathbf{x}) \geq \\ &\geq -\|\mathbf{g}_j(k)\|_{\ell_2}\|\mathbf{x}^j(k)-\mathbf{y}(k)\|_{\ell_2} + f_j(\mathbf{y}(k)) - f_j(\mathbf{x}) .\end{aligned}$$

By combining the preceding three relations, it follows that for any $j$ and any $\mathbf{x} \in \mathbb{R}^N$,

$$\left\{\mathbf{d}^j(k)\right\}^T (\mathbf{y}(k) - \mathbf{x}) \geq - \left(\|\mathbf{d}^j(k)\|_{\ell_2} + \|\mathbf{g}^j(k)\|_{\ell_2}\right) \|\mathbf{y}(k) - \mathbf{x}^j(k)\|_{\ell_2} + \\ + f_j(\mathbf{y}(k)) - f_j(\mathbf{x}) \,,$$

and since $\gamma_j(k) \geq 0$, for all $j$, it also holds the following

$$\gamma_j(k) \left\{\mathbf{d}^j(k)\right\}^T (\mathbf{y}(k) - \mathbf{x}) \geq -\gamma_j(k) \left(\|\mathbf{d}^j(k)\|_{\ell_2} + \|\mathbf{g}^j(k)\|_{\ell_2}\right) \|\mathbf{y}(k) - \mathbf{x}^j(k)\|_{\ell_2} + \\ + \gamma_j(k) \left[f_j(\mathbf{y}(k)) - f_j(\mathbf{x})\right] \,.$$

Summing this relation over all $j$, we obtain

$$\sum_{j=1}^{M} \gamma_j(k) \left\{\mathbf{d}^j(k)\right\}^T (\mathbf{y}(k) - \mathbf{x}) \geq \\ \geq - \sum_{j=1}^{M} \gamma_j(k) \left(\|\mathbf{d}^j(k)\|_{\ell_2} + \|\mathbf{g}^j(k)\|_{\ell_2}\right) \|\mathbf{y}(k) - \mathbf{x}^j(k)\|_{\ell_2} + \sum_{j=1}^{M} \gamma_j(k) \left[f_j(\mathbf{y}(k)) - f_j(\mathbf{x})\right] \,.$$

By combining the preceding inequality with Equation (3.19) we finally obtain the desired result, i.e., for all $\mathbf{x} \in \mathbb{R}^N$ and all $k \geq 1$

$$\|\mathbf{y}(k+1) - \mathbf{x}\|_{\ell_2}^2 \leq \|\mathbf{y}(k) - \mathbf{x}\|_{\ell_2}^2 + \\ + \frac{2}{M} \sum_{j=1}^{M} \gamma_j(k) \left[\left(\|\mathbf{d}^j(k)\|_{\ell_2} + \|\mathbf{g}^j(k)\|_{\ell_2}\right) \|\mathbf{y}(k) - \mathbf{x}^j(k)\|_{\ell_2}\right] + \\ - \frac{2}{M} \sum_{j=1}^{M} \gamma_j(k) \left[f_j(\mathbf{y}(k)) - f_j(\mathbf{x})\right] + \frac{1}{M^2} \sum_{j=1}^{M} \gamma_j^2(k) \|\mathbf{d}^j(k)\|_{\ell_2}^2 \,.$$

□

To carry on our rationale, we also need the following result, that can be proved using Lemma 2 in the Appendix 3.C.

**Proposition 5.** *Under Assumption 1 and decreasing step-size rule (see Section 3.4), let $\mathbf{x}^i(k)$ be generated according to Equation (3.14) for all $i \in \{1, \ldots, M\}$, $k \geq 1$ and $\mathbf{y}(k)$ be generated according to Equation (3.15) for all $k \geq 1$. Let us also define a sequence $\{\gamma_{\max}(s)\}_{s \geq 1}$ such that $\gamma_{\max}(s) \stackrel{\text{def}}{=} \max_i \gamma_i(s)$ for each $s \geq 1$. Then*

$$\sum_{k=1}^{+\infty} \gamma_{\max}(k) \|\mathbf{y}(k) - \mathbf{x}^i(k)\|_{\ell_2} < \infty \quad \textit{a. s. , for all } i \,.$$

*Proof.* First of all we note that: 1) since $\{\gamma_i(k)\}_{k\geq 1}$ satisfies $\lim_{k\to\infty}\gamma_i(k)=0$ for all $i$, also $\lim_{k\to\infty}\gamma_{\max}(k)=0$, and 2) since $\{\gamma_i(k)\}_{k\geq 1}$ satisfies $\sum_{k=1}^{\infty}\gamma_i^2(k)<\infty$ for all $i$ in the limited set $\{1,\dots,M\}$, also $\sum_{k=1}^{\infty}\gamma_{\max}^2(k)<\infty$. In fact, we have that $\sum_{k=1}^{\infty}\gamma_{\max}^2(k)\leq\sum_{k=1}^{\infty}\sum_{i=1}^{M}\gamma_i^2(k)\leq\sum_{i=1}^{M}\sum_{k=1}^{\infty}\gamma_i^2(k)<\infty$.

Now, from Equation (3.17) in the Appendix 3.C and using $\gamma_i(s)\leq\gamma_{\max}(s)$ for all $i$, we have that

$$\|\mathbf{y}(k)-\mathbf{x}^i(k)\|_{\ell_2}\leq\sum_{j=1}^{M}\|\mathbf{x}^j(1)\|_{\ell_2}b(k-1,1)+LM\sum_{s=1}^{k-2}\gamma_{\max}(s)b(k-1,s+1)+2L\gamma_{\max}(k-1)\,,$$

that, with $C_1\stackrel{\text{def}}{=}\max\left\{\sum_{j=1}^{M}\|\mathbf{x}^j(1)\|_{\ell_2},ML\right\}$ and $\gamma_{\max}(0)\stackrel{\text{def}}{=}1$, can be rewritten as

$$\|\mathbf{y}(k)-\mathbf{x}^i(k)\|_{\ell_2}\leq C_1\sum_{s=0}^{k-2}\gamma_{\max}(s)b(k-1,s+1)+2L\gamma_{\max}(k-1)\,,$$

multiply at both sides for $\gamma_{\max}(k)$ we have

$$\gamma_{\max}(k)\|\mathbf{y}(k)-\mathbf{x}^i(k)\|_{\ell_2}\leq C_1\sum_{s=0}^{k-2}\gamma_{\max}(k)\gamma_{\max}(s)b(k-1,s+1)+2L\gamma_{\max}(k)\gamma_{\max}(k-1)\,.$$

Recalling Proposition 3 we have that for almost all the sequences $\{\mathbf{A}(k)\}_{k\geq 1}$, $b(k,s)\leq C_2\beta^{k-s}$ for all $k\geq s$, where $C_2>0$ and $0<\beta<1$ (with $C_2$ in general depending of the considered sequence). Calling $C$ the product $C_1C_2$, we have for almost all the sequences

$$\gamma_{\max}(k)\|\mathbf{y}(k)-\mathbf{x}^i(k)\|_{\ell_2}\leq C\sum_{s=0}^{k-2}\gamma_{\max}(k)\gamma_{\max}(s)\beta^{k-s-2}+2L\gamma_{\max}(k)\gamma_{\max}(k-1)\,.$$

Noting that $\gamma_{\max}(k)\gamma_{\max}(s)\leq\gamma_{\max}^2(k)+\gamma_{\max}(s)^2$ and $2\gamma_{\max}(k)\gamma_{\max}(k-1)\leq\gamma_{\max}^2(k)+\gamma_{\max}^2(k-1)$, we obtain for almost all the sequences

$$\begin{aligned}\gamma_{\max}(k)\|\mathbf{y}(k)-\mathbf{x}^i(k)\|_{\ell_2}\leq&C\gamma_{\max}^2(k)\sum_{s=0}^{k-2}\beta^{k-s-2}+C\sum_{s=0}^{k-2}\gamma_{\max}^2(s)\beta^{k-s-2}+L\gamma_{\max}^2(k)+\\&+L\gamma_{\max}^2(k-1)\leq\\&\leq\frac{C\gamma_{\max}(k)^2}{1-\beta}+C\sum_{s=0}^{k-2}\gamma_{\max}^2(s)\beta^{k-s-2}+L\gamma_{\max}^2(k)+L\gamma_{\max}^2(k-1)\,,\end{aligned}$$

and summing this last inequality over all the $k$

$$\begin{aligned}\sum_{k=1}^{+\infty}\gamma_{\max}(k)\|\mathbf{y}(k)-\mathbf{x}^i(k)\|_{\ell_2}\leq&\frac{C}{1-\beta}\sum_{k=1}^{+\infty}\gamma_{\max}^2(k)+C\sum_{k=1}^{+\infty}\sum_{s=0}^{k-2}\gamma_{\max}^2(s)\beta^{k-s-2}+L\sum_{k=1}^{+\infty}\gamma_{\max}^2(k)+\\&+L\sum_{k=1}^{+\infty}\gamma_{\max}^2(k-1)\quad\text{a. s. .}\end{aligned}\tag{3.20}$$

The desired result follows straightforwardly since: 1) the first, third and fourth term at the right hand side of Equation (3.20) are summable as stated at the beginning of the proof, and 2) the second term of Equation (3.20) is summable because Lemma 2 in the Appendix 3.C applies. □

Now we have all the tools to derive Equation (3.10). Applying iteratively Lemma 3 we have

$$\|\mathbf{y}(k+1)-\mathbf{x}\|_{\ell_2}^2 \leq \|\mathbf{y}(1)-\mathbf{x}\|_{\ell_2}^2 + \frac{2}{M}\sum_{s=1}^{k}\sum_{j=1}^{M}\gamma_j(s)\left[\left(\|\mathbf{d}^j(s)\|_{\ell_2}+\|\mathbf{g}^j(s)\|_{\ell_2}\right)\|\mathbf{y}(s)-\mathbf{x}^j(s)\|_{\ell_2}\right]+$$
$$-\frac{2}{M}\sum_{s=1}^{k}\sum_{j=1}^{M}\gamma_j(s)\left[f_j(\mathbf{y}(s))-f_j(\mathbf{x})\right]+\frac{1}{M^2}\sum_{s=1}^{k}\sum_{j=1}^{M}\gamma_j^2(s)\|\mathbf{d}^j(s)\|_{\ell_2}^2 .$$

and then we can write

$$\sum_{s=1}^{k}\sum_{j=1}^{M}\gamma_j(s)\left[f_j(\mathbf{y}(s))-f_j(\mathbf{x})\right] \leq \frac{M}{2}\|\mathbf{y}(1)-\mathbf{x}\|_{\ell_2}^2+$$
$$+\sum_{s=1}^{k}\sum_{j=1}^{M}\gamma_j(s)\left[\left(\|\mathbf{d}^j(s)\|_{\ell_2}+\|\mathbf{g}^j(s)\|_{\ell_2}\right)\|\mathbf{y}(s)-\mathbf{x}^j(s)\|_{\ell_2}\right]+\frac{1}{2M}\sum_{s=1}^{k}\sum_{j=1}^{M}\gamma_j^2(s)\|\mathbf{d}^j(s)\|_{\ell_2}^2 \leq$$
$$\leq \frac{M}{2}\|\mathbf{y}(1)-\mathbf{x}\|_{\ell_2}^2+2L\sum_{s=1}^{k}\sum_{j=1}^{M}\gamma_j(s)\|\mathbf{y}(s)-\mathbf{x}^j(s)\|_{\ell_2}+\frac{L^2}{2M}\sum_{s=1}^{k}\sum_{j=1}^{M}\gamma_j^2(s) ,$$

where the last inequality follows from the bounded sub-gradient Assumption 2. Using the sequence $\{\gamma_{\max}(s)\}_{s\geq 1}$ defined in the Proposition 5 of above and rearranging some terms, we obtain

$$\sum_{s=1}^{k}\sum_{j=1}^{M}\gamma_j(s)\left[f_j(\mathbf{y}(s))-f_j(\mathbf{x})\right] \leq \frac{M}{2}\|\mathbf{y}(1)-\mathbf{x}\|_{\ell_2}^2+2L\sum_{j=1}^{M}\sum_{s=1}^{k}\gamma_{\max}(s)\|\mathbf{y}(s)-\mathbf{x}^j(s)\|_{\ell_2}+$$
$$+\frac{L^2}{2M}\sum_{j=1}^{M}\sum_{s=1}^{k}\gamma_{\max}^2(s) .$$

For $k$ that goes to infinity, $\sum_{s=1}^{k}\gamma_{\max}^2(s)<\infty$ as note in the proof of Proposition 5, whilst $\sum_{s=1}^{k}\gamma_{\max}(s)\|\mathbf{y}(s)-\mathbf{x}^j(s)\|_{\ell_2}<\infty$ for the Proposition 5 itself. $\frac{M}{2}\|\mathbf{y}(1)-\mathbf{x}\|_{\ell_2}^2$ is a constant term, therefore Equation (3.10) follows.

## 3.E Derivation of the Utility Function in Equation (3.12)

In this appendix we briefly review the mathematical steps of the analysis carried on in [83], under the assumptions done in Section 3.5 and with the formalism introduced therein. The overall objective of this appendix is to formally derive Equation (3.12).

For a given time $\bar{t}$ and node $i \in \mathcal{N}$ we define the process $B_i(\bar{t}, t) \subseteq \mathcal{N}$ as the set containing all the nodes $j$ such that, if a message is given to them at time $\bar{t} - t$, it can reach user $i$ in two hops by time $\bar{t}$. For each node pair $(i, j)$, $j \neq i, j, i \in \mathcal{N}$, we define also $s_{ij}(\bar{t})$ as

$$s_{ij}(\bar{t}) \stackrel{\text{def}}{=} \inf_{t \geq 0} \{t : j \in B_i(\bar{t}, t)\} \, .$$

$\bar{t} - s_{ij}(\bar{t})$ indicates the minimum amount of time required to transfer information from node $j$ to node $i$ at time $\bar{t}$ through file sharing. If we assume that the inter-meeting process time between the node pair $(i, j)$ is exponentially distributed with parameter $\lambda_{ij}$ (i.e., Poisson meeting process) we have that in the case of two-hops protocol also $s_{ij}(\bar{t})$ is exponentially distributed with parameter $\lambda_{ij}$.

Let then $Y_i^{SP}(\bar{t})$ be, at time $\bar{t}$, the elapsed time since user $i$ downloaded content directly from the SP, i.e., $Y_i^{SP}(\bar{t}) = \bar{t} - t_i^{SP}(\bar{t})$. Note that the random variable $Y_i^{SP}(\bar{t})$ for all $i \in \mathcal{N}$ is exponentially distributed with parameter $x_i$, this because the SP transfers updates directly to node $i$ with rate $x_i$ and in stationary conditions the forward process and the backward one have the same statistic, see e.g., [95].

Lemma 1 of [83] states that

$$Y_i(\bar{t}) = \min_{j \in \mathcal{N}} \left\{ s_{ij}(\bar{t}) + Y_j^{SP}(\bar{t} - s_{ij}(\bar{t})) \right\} \, , \tag{3.21}$$

therefore, in our case, $Y_i(\bar{t})$ is the minimum over $M$ independent random variables. One of this random variables is exponentially distributed with parameter $x_i$ (i.e., $s_{ij}(\bar{t}) + Y_j^{SP}(\bar{t} - s_{ij}(\bar{t}))$ with $j = i$) and takes in account the directed updates of $i$ from the SP. The remaining $M - 1$ random variables are sum of two independent exponential random variables: one distributed with parameter $\lambda_{ij}$ and the second with parameter $x_j$. Each of these $M - 1$ variables models the update of $i$ in two hops from the PS through a given relay node $j \neq i$. For the independence of $s_{ij}$ and $Y_j^{SP}$, and assuming in general $\lambda_{ij} \neq x_j$, we have that

$$p_{s_{ij} + Y_j^{SP}}(y) = \lambda_{ij} x_j \left[ \frac{e^{-\lambda_{ij} y} - e^{-x_j y}}{x_j - \lambda_{ij}} \right] \quad y > 0 \, ,$$

whence

$$P\left[ s_{ij} + Y_j^{SP} > y \right] = \frac{x_j e^{-\lambda_{ij} y} - \lambda_{ij} e^{-x_j y}}{x_j - \lambda_{ij}} \, ,$$

therefore

$$P\left[Y_i > y\right] = \left[\prod_{j \in \mathcal{N}_i} \frac{x_j e^{-\lambda_{ij} y} - \lambda_{ij} e^{-x_j y}}{x_j - \lambda_{ij}}\right] e^{-x_i y} \,,$$

where $\mathcal{N}_i \stackrel{\text{def}}{=} \{j : j \neq i, \lambda_{ij} > 0\}$. In the case of utility $u(Y_i) = \chi\left\{Y_i \leq \tau\right\}$ we have that

$$f_i(\mathbf{x}) \;\; = \;\; E_{\mathbf{x}}\left[u_i(Y_i)\right] = P\left[Y_i \leq \tau\right] = 1 - \left[\prod_{j \in \mathcal{N}_i} \frac{x_j e^{-\lambda_{ij}\tau} - \lambda_{ij} e^{-x_j \tau}}{x_j - \lambda_{ij}}\right] e^{-x_i \tau} \,,$$

thus

$$f(\mathbf{x}) \;\; = \;\; \sum_{i=1}^{M} f_i(\mathbf{x}) = M - \sum_{i=1}^{M} \left[\prod_{j \in \mathcal{N}_i} \frac{x_j e^{-\lambda_{ij}\tau} - \lambda_{ij} e^{-x_j \tau}}{x_j - \lambda_{ij}}\right] e^{-x_i \tau} \,.$$

From Equation (3.12) is straightforward to compute the components of the gradient vector we need to implement the distributed sub-gradient method, for all $i$:

$$\frac{\partial f_i(\mathbf{x})}{\partial x_i} = \tau \left[\prod_{j \in \mathcal{N}_i} \frac{x_j e^{-\lambda_{ij}\tau} - \lambda_{ij} e^{-x_j \tau}}{x_j - \lambda_{ij}}\right] e^{-x_i \tau} \,,$$

and, for $z \neq i$

$$\frac{\partial f_i(\mathbf{x})}{\partial x_z} \;\; = \;\; \left[\prod_{\substack{j \in \mathcal{N}_i \\ j \neq z}} \frac{x_j e^{-\lambda_{ij}\tau} - \lambda_{ij} e^{-x_j \tau}}{x_j - \lambda_{ij}}\right] e^{-x_i \tau} \cdot \frac{\lambda_{iz}\left\{e^{-\lambda_{iz}\tau} + \left[\tau(\lambda_{iz} - x_z) - 1\right] e^{-x_z \tau}\right\}}{(x_z - \lambda_{iz})^2} \,.$$

# Bibliography


[1] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-Network Aggregation Techniques for Wireless Sensor Networks: A Survey," *IEEE Wireless Communication Magazine*, pp. 70–87, Apr. 2007.

[2] A. Scaglione and S. D. Servetto, "On the Interdependence of Routing and Data Compression in Multi-Hop Sensor Networks," in *ACM MOBICOM*, Atlanta, GA, USA, Sep. 2002.

[3] S. Servetto, "Distributed Signal Processing Algorithms for the Sensor Broadcast Problem," in *Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, USA, Mar. 2003.

[4] S. Servetto., "Sensing Lena - Massively Distributed Compression of Sensor Images," in *IEEE Conference on Image Processing (ICIP)*, Ithaca, NY, USA, Sep. 2003.

[5] S. Pradhan and K. Ramchandran, "Distributed Source Coding Using Syndromes (DISCUS): Design and Construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626–643, Mar. 2003.

[6] H. Luo and G. Pottie, "Routing Explicit Side Information for Data Compression in Wireless Sensor Networks," in *Distributed Computing in Sensor Systems (DCOSS)*, Marina del Rey, CA, USA, Jun. 2005.

[7] M. Gastpar, P. Dragotti, and M. Vetterli, "The Distributed Karhunen-Loeve Transform," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 5177–5196, Dec. 2006.

[8] D. Slepian and J. Wolf, "Noiseless Coding of Correlated Information Sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

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

[10] Z. Xiong, A. Liveris, and S. Cheng, "Distributed Source Coding for Sensor Networks," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–92, Sep. 2004.

[11] S. Pattem, B. Krishnamachari, and R. Govindan, "The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks," in *Int. Conf. on Information Processing in Sensor Networks (IPSN)*, Berkeley, CA, USA, Apr. 2004.

[12] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, "Energy-Efficient Data Representation and Routing for Wireless Sensor Networks Based on a Distributed Wavelet Compression Algorithm," in *IPSN'06: Proc. of the 5th international conference on Information Processing in Sensor Networks*, Nashville, TN, USA, Apr. 2006.

[13] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Data Gathering with Tunable Compression in Sensor Networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 19, no. 2, pp. 276–287, 2008.

[14] D. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 4036–4048, Apr. 2006.

[15] E. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. on Information Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[16] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[17] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak, "Compressive Sensing for Networked Data: a Different Approach to Decentralized Compression," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 92–101, Mar. 2008.

[18] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: design, analysis and applications," in *INFOCOM 2005. 24th Annual Joint Conf. of the IEEE Comp. and Comm. Soc. Proceedings IEEE*, Miami, Florida, USA, 2005.

[19] S.M. Hedetniemi and S.T. Hedetniemi and A.L. Liestman, "A Survey of Gossiping and Broadcasting in Communication Networks," *Networks*, vol. 18, no. 4, pp. 319–349, 1988.

[20] A. Nedic and A. Ozdaglar, "Distributed Subgradient Methods for Multi-Agent Optimization," *Automatic Control, IEEE Transactions on*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[21] I. Lobel and A. Ozdaglar, "Distributed Subgradient Methods for Convex Optimization over Random Networks," *Automatic Control, IEEE Transactions on - Accepted for publication*, May 2010.

[22] R. Wei, R. Beard, and E. Atkins, "A survey of Consensus Problems in Multi-agent Coordination," *American Control Conference, 2005. Proceedings of the 2005*, vol. 3, pp. 1859–1864, Jun. 2005.

[23] P. Denantes, F. Bénézit, P. Thiran, and M. Vetterli, "Which Distributed Averaging Algorithm Should I Choose for my Sensor Network?" in *Proc. 27th IEEE Conf. Computer Communications and Networks*, 2008.

[24] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant Networking: an Approach to Interplanetary Internet," *Communications Magazine, IEEE*, vol. 41, no. 6, pp. 128–136, Jun. 2003.

[25] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks," *Communications Magazine, IEEE*, vol. 44, no. 11, pp. 134–141, Nov. 2006.

[26] G. Quer, "Optimization of Cognitive Wireless Networks using Compressive Sensing and Probabilistic Graphical Models," PhD dissertation, Department of Information Engineering, University of Padova, Padova, IT, 2011.

[27] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Trans. on Information Theory*, vol. 52, no. 9, pp. 4036–4048, Sep. 2006.

[28] E. Candès, "Compressive Sampling," in *Int. Congress of Mathematics*, Madrid, Spain, 2006.

[29] R. Baraniuk, "Compressive Sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, Jul. 2007.

[30] E. Candès and M. Wakin, "An Introduction to Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[31] C. E. Shannon, "Communication in the presence of noise," *Proc. Institute of Radio Engineers*, vol. 37, no. 1, pp. 10–21, Jan. 1949.

[32] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed L0 norm," *IEEE Trans. on Signal Processing*, 2009, Accepted for publication.

[33] S. Bercker, J. Bobin, and E. J. Candés, "NESTA: a fast and accurate first order method for sparse recovery." *Submitted for publication*. [Online]. Available: http://www-stat.stanford.edu/~candes/papers/NESTA.pdf

[34] Y. E. Nesterov, "A method for unconstrained convex minimization problem with rate of convergence $\mathcal{O}(1/k^2)$," *Doklady AN USSR (Translated as Soviet Math. Docl.)*, vol. 269, no. 3, pp. 543–547, 1983.

[35] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic Decomposition by Basis Pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[36] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. R. Statist. Soc. B*, vol. 58, pp. 267–288, 1996.

[37] R. T. Rockafellar, *Convex Analysis*. Princeton Landmarks in Mathematics and Physics, Princeton University Press, 1970.

[38] Y. E. Nesterov, "Smooth minimization of non–smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, may 2005.

[39] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Compressive wireless sensing," in *Int. Conf. on Information Processing in Sensor Networks (IPSN)*, Nashville, TN, USA, Apr. 2006.

[40] ——, "Joint source-channel communication for distributed estimation in sensor networks," *IEEE Trans. on Information Theory*, vol. 53, no. 10, pp. 3629–3653, Oct. 2007.

[41] M. Rabbat, J. Haupt, A. Singh, and R. Novak, "Decentralized Compression and Predistribution via Randomized Gossiping," in *Information Processing in Sensor Networks (IPSN)*, Nashville, Tennessee, USA, Apr. 2006.

[42] G. Shen, S. Y. Lee, S. Lee, S. Pattem, A. Tu, B. Krishnamachari, A. Ortega, M. Cheng, S. Dolinar, A. Kiely, M. Klimesh, and H. Xie, "Novel distributed wavelet transforms and

routing algorithms for efficient data gathering in sensor webs," in *NASA Earth Science Technology Conference (ESTC2008)*, University of Maryland, MD, USA, Jun. 2008.

[43] M. Duarte, M. Wakin, and R. Baraniuk, "Wavelet-domain compressive signal reconstruction using a Hidden Markov Tree model," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, Apr. 2008, pp. 5137–5140.

[44] M. Coates, Y. Pointurier, and M. Rabbat, "Compressed Network Monitoring for IP and All-optical Networks," in *Internet Measurement Conference (IMC)*, San Diego, CA, USA, Oct. 2007.

[45] M. Duarte, M. Wakin, D. Baron, and R. Baraniuk, "Universal Distributed Sensing via Random Projections," in *Information Processing in Sensor Networks (IPSN)*, Nashville, TN, USA, Apr. 2006.

[46] M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk, "Distributed Compressed Sensing of Jointly Sparse Signals," in *Signals, Systems and Computers, 2005. Conference Record of the Thirty-Ninth Asilomar Conference on*, Oct. 2005.

[47] S. Shintre, B. Dey, S. Katti, S. Jaggi, D. Katabi, and M. Medard, ""Real" and "Complex" Network Codes: Promises and Challenges," in *Network Coding, Theory and Applications, 2008. NetCod 2008. Fourth Workshop on*, Jan. 2008, pp. 1–6.

[48] S. Katti, S. Shintre, S. Jaggi, D. Katabi, and M. Medard, "Real Network Codes," in *Forty-Fifth Annual Allerton Conference*, Allerton House, UIUC, IL, USA, Sep. 2007.

[49] Y. Weiss, H. S. Chang, and W. T. Freeman, "Learning Compressed Sensing," in *Forty-Fifth Annual Allerton Conference*, Allerton House, UIUC, IL, USA, Sep. 2007.

[50] "MIT Wireless Network Coverage," Last time accessed: January 2009. [Online]. Available: http://nie.chicagotribune.com/activities_120505.htm

[51] T. Kamakaris and J. V. Nickerson, "Connectivity maps: Measurements and applications," in *Proceedings of the 38th Hawaii International Conference on System Sciences*, Hilton Waikoloa Village Island of Hawaii, HI, USA , Jan. 2005.

[52] "EPFL LUCE SensorScope WSN," Last time accessed: January 2009. [Online]. Available: http://sensorscope.epfl.ch/

[53] "Tropical Rainfall Measuring Mission," Last time accessed: January 2009. [Online]. Available: http://trmm.gsfc.nasa.gov

[54] "Partnership for Interdisciplinary Studies of Coastal Oceans," Last time accessed: January 2009. [Online]. Available: www.piscoweb.org

[55] "ENEA: Ente Nuove Tecnologie e l'Ambiente," Last time accessed: January 2009. [Online]. Available: www.enea.it

[56] Napler Addison, *The Illustrated Wavelet Transform Handbook*. Taylor & Francis, 2002.

[57] D. T. Sandwell, "Geophysical Research Letters," *Biharmonic Spline Interpolation of GEOS-3 and SEASAT Altimeter Data*, vol. 14, no. 2, pp. 139–142, 1987.

[58] S. Gull, "The Use and Interpretation of Principal Component Analysis in Applied Research," *Maximum Entropy and Bayesian Methods in Science and Engineering*, vol. 1, pp. 53–74, 1988.

[59] J. Skilling, *Maximum Entropy and Bayesian Methods*. Kluwer academic, 1989.

[60] S. Ji, Y. Xue, and L. Carin, "Bayesian Compressive Sensing," *IEEE Trans. on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.

[61] C. R. Rao, "The Use and Interpretation of Principal Component Analysis in Applied Research," *Sankhya: The Indian Journal of Statistics*, vol. 26, pp. 329–358, 1964.

[62] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[63] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[64] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*. Springer, 2008.

[65] P. Casari, A. P. Castellani, A. Cenedese, C. Lora, M. Rossi, L. Schenato, and M. Zorzi, "The "Wireless Sensor Networks for City-Wide Ambient Intelligence (WISE-WAI)" Project," *Sensors*, vol. 9, no. 6, pp. 4056–4082, May 2009.

[66] R. Crepaldi, S. Friso, A. F. Harris III, M. Mastrogiovanni, C. Petrioli, M. Rossi, A. Zanella, and M. Zorzi, "The Design, Deployment, and Analysis of SignetLab: A Sen-

sor Network Testbed and Interactive Management Tool," in *IEEE Tridentcom*, Orlando, FL, US, May 2007.

[67] "EPFL Grand-St-Bernard SensorScope WSN," Last time accessed: September 2009. [Online]. Available: http://sensorscope.epfl.ch/index.php/Grand-St-Bernard_Deployment

[68] "CitySense," Last time accessed: October 2009. [Online]. Available: http://www.citysense.net

[69] T. Watteyne, D. Barthel, M. Dohler, and I. Auge-Blum, "Sense&Sensitivity: A Large-Scale Experimental Study of Reactive Gradient Routing," *Measurement Science and Technology, Special Issue on Wireless Sensor Networks: Designing for Real-world Deployment and Deployment Experiences*, vol. 21, no. 12, Oct. 2010.

[70] D. J. MacKay, "Bayesian Interpolation," *Neural Computation Journal*, vol. 4, no. 3, pp. 415–447, May 1992.

[71] G. Schwarz, "Estimating the Dimension of a Model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[72] K.M. Hanson, *Image Recovery: Theory and Applications*. ed. H. Stark.

[73] D. T. Sandwell, "Biharmonic Spline Interpolation of GEOS-3 and SEASAT Altimeter Data," *Geophysical Research Letters*, vol. 14, no. 2, pp. 139–142, 1987.

[74] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[75] K. Fan, "On a theorem of Weil concerning eigenvalues of linear transformation I," *Proc. of the National Academy of Sciences*, vol. 35, pp. 652–655, 1949.

[76] A. Barron, "Entropy and the Central Limit Theorem," *The Annals of Probability*, vol. 14, no. 1, pp. 336–342, Jan. 1986.

[77] P. Casari, A. P. Castellani, A. Cenedese, C. Lora, M. Rossi, L. Schenato, and M. Zorzi, "The Wireless Sensor Networks for City-Wide Ambient Intelligence (WISE-WAI) Project," *Sensors*, vol. 9, no. 6, pp. 4056–4082, May 2009.

[78] G. Mulligan, "The 6LoWPAN architecture," in *EmNets'07: Proceedings of the 4th workshop on Embedded networked sensors*, Cork, Ireland, Jun. 2007.

[79] A. P. Castellani, N. Bui, P. Casari, M. Rossi, Z. Shelby, and M. Zorzi, "Architecture and Protocols for the Internet of Things: A Case Study," in *International Workshop on the Web of Things (WoT)*, Mannheim, Germany, Mar. 2010.

[80] D. Heffelfinger, *Java EE 5 Development with NetBeans 6*. Packt Publishing, 2008.

[81] G. R. Belitskii and Yu. I. Lyubich, *Matrix norms and their applications*. Birkhäuser, 1988.

[82] G.-S. Ahn, E. Miluzzo, A. T. Campbell, S. G. Hong, and F. Cuomo, "Funneling-MAC: A Localized, Sink-Oriented MAC For Boosting Fidelity in Sensor Networks," in *ACM Sensys*, Boulder, CO, USA, Nov. 2006.

[83] S. Ioannidis, A. Chaintreau, and L. Massoulie, "Optimal and Scalable Distribution of Content Updates over a Mobile Social Network," in *IEEE INFOCOM 2009*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1422–1430.

[84] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 1, pp. 77–90, Feb. 2008.

[85] G. Neglia, G. Reina, and S. Alouf, "Distributed Gradient Optimization for Epidemic Routing: A Preliminary Evaluation," in *Wireless Days (WD), 2009 2nd IFIP*, Paris, FR, Dec. 2009.

[86] C. Bettstetter, "Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects," *ACM Mobile Comp. Comm. Rev.*, vol. 5, no. 3, pp. 55–67, 2001.

[87] A. N. Shiryaev, *Probability (2nd ed.)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1995, translator-Boas, R. P.

[88] R.C. Bradley, "Basic Properties of Strong Mixing Conditions. A Survey and Some Open Questions," *Probability Surveys*, 2005.

[89] A. Tahbaz-Salehi and A. Jadbabaie, "Consensus Over Ergodic Stationary Graph Processes," *Automatic Control, IEEE Transactions on*, vol. 55, no. 1, pp. 225–230, Jan. 2010.

[90] E. Seneta, *Non-negative Matrices and Markov Chains*. Springer, New York $2^{rd}$ edition, 1981.

[91] J.C. Oxtoby, "Ergodic sets," *Bulletin of the American Mathematical Society*, vol. 58, pp. 116–136, 1952.

[92] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[93] A. Nedic, A. Ozdaglar, and P. Parrilo, "Constrained Consensus and Optimization in Multi-Agent Networks," *Automatic Control, IEEE Transactions on*, Feb. 2010, accepted for publication.

[94] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized Gossip Algorithms," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[95] S.M. Ross, *Introduction to Probability Models*. Elsevier, Academic Press, 2007.