

# Distributed detection, reconstruction and interpolation for sensor network data

THÈSE N° 5748 (2013)

PRÉSENTÉE LE 21 JUIN 2013

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR  
LABORATOIRE DE TRAITEMENT DES SIGNAUX 4  
PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Tamara TOŠIĆ

acceptée sur proposition du jury:

Prof. J.-Ph. Thiran, président du jury  
Prof. P. Frossard, directeur de thèse  
Prof. A. P. Burg, rapporteur  
Prof. M. Kieffer, rapporteur  
Prof. F. Labeau, rapporteur



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE

Suisse  
2013



*To my family*



## Acknowledgements

This thesis is realized thanks to many people and it is a pleasure to acknowledge them. First, I want to thank my thesis advisor Prof. Pascal Frossard for his support and patience. Pascal, thank you for all the freedom, inspiration and support you provided during my PhD, as well as for persistence and enthusiasm in correcting my thesis manuscript. Also, I want to thank you for providing very relaxed, supportive and cheerful work environment in the LTS4 lab.

I want to thank to the thesis jury members, Prof. Michel Kieffer, Prof. Andreas Burg, Prof. Fabrice Labeau and the president of the jury, Prof. Jean-Philippe Thiran, for their valuable advices and comments about my thesis that helped me to improve the final manuscript.

I had many inspiring teachers and it was a great pleasure to learn from you. Thank you for your support and patience. Special thanks goes to Prof. Željko Trpovski and Prof. Hynek Hermansky, for their enthusiasm and wise advices. I want to thank Ivana and Nikos for working with me on the projects; to VJ and Elif for stimulating discussions; Laura, Thomas and David for their time and very precious comments when I needed them and to my students for their help in work. All the previous and current LTS4s, you're great and I had a lot of fun with you! Thank you Rosie for your help with administrative issues and for your patience and thank you Xiaowen, Nikos, Ivan, Thomas, Leila and Dorina for help in proof reading of the parts of this thesis.

Dear friends from Serbia and friends from Martigny and Lausanne - thank you for sharing ups and downs of everyday life, as well as work excitements and disappointments. You were there for me, no matter how many kilometers (or miles) and time zones we are apart! I really enjoy sharing things with you. :) You all helped me to "reset" the bad days with your warm hearts and smiles. Thank you with all my heart. Extra hugs go to Nada, Egi, Maja, Tanja, Ana Lj., Jelena, Vesna, Nikola, Dejan G., Kate, Sriram, Guillermo, Deepu, Ivana T., Ana, Anna, Xiaowen, Marija, Mirjana, Ana J. and the current and ex members of STI-IEL from the corridor for being so special!

This thesis would of not be possible without the unconditional love and support of my family and Nada. You were always there for me and you gave me all the love, courage and space that I needed - you are my source of strength and happiness.



# Abstract

In the past decade, technology developments have triggered the emergence of cheap sensing devices that open novel perspective for large-scale data sensing and analysis in different application domains, such environmental monitoring, healthcare or security, or distributed control systems. Moreover, networks of such sensors can advantageously replace complex and expensive systems for data acquisition in large areas. For example, a network of temperature sensors can efficiently replace expensive high resolution infrared satellite imaging. However, a key condition for the development of sensor networks resides in the design of efficient distributed solutions for data gathering and analysis. Such solutions need to be adaptive to the network characteristics, to the data to be measured, and to the computational load and communication cost that should ideally be shared among the different sensors in the system. These constraints pose important challenges in the design of effective in-network processing methods for distributed signal sensing and analysis.

In this thesis, we focus on the design of algorithms for effective data gathering and analysis in sensor networks. We study classical signal processing problems from the new perspective of modern distributed signal processing challenges. In particular, we address important issues, such as, (i) the distributed detection of defective sensors, (ii) the distributed data gathering for effective signal reconstruction from a small number of messages and (iii) the distributed interpolation of signals in sensor networks deployed on smooth geometric manifolds.

We first study the problem of distributed detection of defective sensors in networks. Erroneous data measurements collected by defective sensors can have unpredictable consequences and possibly lead to severe impairments on analysis or estimation tasks. The detection of erroneous data is therefore of crucial importance. We propose a novel distributed framework for sensor failure detection and we design probabilistic tests to detect the presence of defective sensors in a network. The test outcome messages from different sensors are combined together and propagated in the network. Failure detection can then be performed in any sensor in the network using a simple distance decoder. We analyse the performance of the proposed framework in different settings, and we show via extensive simulations that our flexible solution leads to the detection of defective sensors with high probability in realistic settings.

Next, we investigate the problem of distributed gathering of sensor data with the objective of effective signal reconstruction with small communication costs. In particular, we focus on the problem of data reconstruction where only a small number of collected messages along with simple signal priors are available at the decoder. We design a new data gathering framework, where sensors distributively transmit quantized measurements towards the receiver along directed communication links. The measurements are combined by the different sensors that participate to the adhoc communication process, such that each coded message gathers information from multiple sensor measurements.

More precisely, the sensors combine their measurements with messages received from neighbour nodes using modulo operations and transmit the resulting messages towards downstream nodes. Such network coding operations generally lead to reduced communication costs in architectures with path diversity such as adhoc sensor networks. We propose a detailed study of the performance of our novel data gathering algorithm and we show by simulations that the receiver is able to efficiently reconstruct the data function with a small number of coded messages, especially when proper signal priors can be used for decoding.

Finally, we develop a novel framework for data interpolation at arbitrary locations from sensor data that have been acquired at different positions on a smooth geometric manifold. We propose a distributed interpolation algorithm, which relies on the kernel-based estimation of local data functions at any sensor location. Such local functions are estimated by solving a regression problem that considers the measurements obtained by neighbour sensors and the geodesic distances between sensors on the manifold in order to properly take into consideration the geometry of the data. We show that the proposed method provides better interpolation performance compared to the classical approaches for signals that are irregularly sampled on spherical manifolds. We further study the data dissemination process in adhoc sensor networks and analyse the evolution of the performance of our distributed interpolation method when messages are progressively collected in the neighbourhood of the interpolating sensor. In the particular case of sensors that are organised in a small world configuration, we show that our distributed interpolation algorithm very rapidly converges to accurate signal estimation.

Overall, this thesis addresses several important issues related to the design of effective distributed data gathering and analysis methods in sensor networks. It studies important challenges in failure detection, data interpolation and signal reconstruction and it provides research contributions in distributed signal processing. In particular, it leverages on group testing ideas to propose a novel distributed algorithm based on probabilistic tests and messages combinations for effective failure detection in any sensor in the network. Furthermore, the unique combination of network coding and prior-based signal reconstruction at receiver permits to reach state-of-the-art performance in distributed data gathering applications. Finally, it introduces a distributed kernel-based data interpolation algorithm for effective signal estimation on smooth manifold, whose performance gracefully improves as the number of available measurements increases. This thesis certainly offers very important insights in the design of distributed signal processing methods that will pave the way to the large-scale deployment of cheap and low power sensors with exciting perspectives in numerous application domains.

**Keywords:** Distributed algorithms, Sensor Networks, Network Coding, Reconstruction with Signal Priors, Data Gathering, Distributed Data Interpolation on Manifolds, Failure Detection



---

# Résumé

Pendant la dernière décennie, les développements technologiques ont favorisé l'émergence de dispositifs de mesure bon marché qui ouvrent de nouvelles perspectives pour les mesures et l'analyse de données à grande échelle, dans différents domaines d'application comme l'observation de l'environnement, la santé et la sécurité, ou les systèmes de contrôle distribués. Des réseaux de tels capteurs peuvent avantageusement remplacer des systèmes complexes et chers pour l'acquisition de données dans de grands espaces. Par exemple, un réseau de capteurs de température peut remplacer des techniques coûteuses d'imagerie satellite à haute résolution. Toutefois, une condition primordiale pour le développement de réseaux de capteurs repose sur la conception de solutions distribuées efficaces pour la collecte et l'analyse de données. De telles solutions doivent être adaptées aux propriétés du réseau, aux données à mesurer et aux coûts de calcul et de communication qui devraient idéalement être distribués entre les différents capteurs du système. Ces contraintes pose des défis importants et stimulants dans la conception de méthodes efficaces de traitement de données dans les réseaux, dans le but de mesurer et d'analyser distribuée de signaux

Dans cette thèse, nous nous focalisons précisément sur la conception d'algorithmes efficaces pour la collecte et l'analyse de données dans des réseaux de capteurs. Nous étudions des problèmes classiques de traitement du signal, mais sous de nouvelles perspectives liées aux défis modernes qui existent en traitement de signal distribué. En particulier, nous étudions des problèmes importants pour le traitement de données dans des réseaux de capteurs, et notamment (i) la détection distribuée de capteurs défectueux, (ii) la collection distribuée de données pour la reconstruction efficace du signal complet à partir d'un petit nombre de messages et (iii) l'interpolation distribuée de signaux dans des réseaux de capteurs positionnés sur des variétés géométriques différentiables.

Nous étudions premièrement le problème de la détection distribuée de capteurs défectueux dans des réseaux. Des mesures erronées collectées par des capteurs défectueux peuvent en effet avoir des conséquences imprévisibles et parfois conduire à des erreurs considérables dans des tâches d'analyse ou d'estimation. La détection des données erronées est donc d'une importance cruciale. Nous proposons un nouveau cadre de travail pour la détection décentralisée de capteurs défectueux et nous concevons des tests probabilistes pour détecter leur présence dans le réseau. Les résultats de ces tests effectués par différents capteurs sont combinés et propagés dans le réseau. La détection de capteurs défectueux peut alors se faire dans n'importe quel senseur du réseau à l'aide d'un décodeur basé sur un simple calcul de distance. Nous analysons les performances de notre méthode dans différents environnements et nous montrons par de multiples simulations que notre solution simple et flexible conduit à la détection fiable des capteurs défectueux dans des conditions de fonctionnement réalistes.

Ensuite, nous étudions le problème de la collecte distribuée des données des capteurs avec l'objectif d'une reconstruction efficace du signal avec de faibles coûts de communication. En particulier, nous nous concentrons sur le problème de la reconstruction de données lorsque un petit nombre de messages et un modèle simple du signal sont disponibles au décodeur. Nous concevons une nouvelle architecture de collecte de données, où les capteurs transmettent de manière distribuée des mesures quantifiées vers le récepteur en suivant des liens de communication dirigés. Les mesures sont combinées par les différents capteurs qui participent au processus de communication adhoc, de telle sorte que chaque message contiennent de l'information sur plusieurs mesures de différents capteurs. Plus précisément, les capteurs combinent leurs mesures avec des messages reçus de leurs voisins à l'aide d'opérations basées sur des modules, et trans-

mettent le résultat vers des capteurs plus proches du récepteur. De telles opérations de codage réseau conduisent généralement à des coûts de communication réduits dans des architectures qui présentent une diversité dans les chemins de communication, comme les réseaux de capteurs adhoc. Nous proposons une étude détaillée des performances de notre nouvel algorithme de collection de données, et nous montrons par des simulations que le récepteur est capable de reconstruire efficacement les données avec un petit nombre de messages codés, en particulier lorsqu'un modèle du signal peut être utilisé par le décodeur.

Finalement, nous développons une nouvelle solution pour l'interpolation de signaux à des endroits arbitraires, à partir de données de capteurs acquises à différentes positions sur une variété géométrique différentiable. Nous proposons un algorithme d'interpolation distribuée, qui repose sur une estimation à l'aide de noyaux de fonctions locales à n'importe quel senseur. De telles fonctions locales sont estimées à l'aide d'un problème de régression qui considère les mesures effectuées par les capteurs voisins, et les distances géodésiques entre les capteurs sur la variété, pour prendre en compte la géométrie des données. Nous démontrons que la méthode proposée conduit à de meilleures performances d'interpolation par rapport aux méthodes classiques, pour des signaux qui sont capturés variétés à des positions distribuées irrégulièrement sur des variétés sphériques. De plus, nous étudions le processus de distribution des données dans des réseaux de capteurs adhoc et nous analysons l'évolution des performances de notre méthode d'interpolation distribuée lorsque les messages des capteurs voisins sont progressivement reçus. Dans le cas particulier où les capteurs sont organisés dans une configuration 'small world', nous montrons que l'algorithme d'interpolation distribuée converge rapidement vers une estimation précise du signal.

En résumé, cette thèse aborde plusieurs problèmes importants liés à la conception de méthodes efficaces pour la collection et l'analyse de données dans des réseaux de capteurs. Elle étudie des défis importants par rapport à la détection de défauts, l'interpolation et la reconstruction de données et propose de nombreuses contributions en traitement distribué du signal. En particulier, elle étend les idées de tests de groupes pour proposer un nouvel algorithme distribué basé sur des tests probabilistes et des combinaisons de messages pour la détection de capteurs défectueux à n'importe quel point du réseau. Ensuite, le travail de thèse propose une combinaison unique de codage réseau et de méthodes de reconstruction basées sur des modèles de signaux, qui permet d'atteindre des performances qui correspondent à l'état de l'art dans les applications de collection distribuée de données. Finalement, elle introduit un nouvel algorithme distribué d'interpolation de données basée sur des noyaux pour l'estimation de signaux sur des variétés différentiables, dont les performances augmentent rapidement avec le nombre de mesures disponibles. Cette thèse offre certainement des enseignements importants pour la conception de méthodes de traitement de signal distribué qui ouvrent la voie au déploiement à grande échelle de capteurs à bas coûts, avec des perspectives intéressante dans de nombreux domaines d'applications.

**Mots Clés :** Algorithmes distribués, Réseaux de capteurs, Codage réseau, Reconstruction avec modèles de signal, Collecte de données, Interpolation distribuée de données sur des variétés, Détection de défauts.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Distributed sensor processing in sensor networks . . . . .	1
1.2	Thesis outline . . . . .	3
1.3	Summary of contributions . . . . .	4
<b>2</b>	<b>Related Work</b>	<b>7</b>
2.1	Network mechanisms . . . . .	7
2.2	Detection of defective sensors . . . . .	9
2.3	Inverse problems for signal recovery . . . . .	11
2.4	Interpolation on geometric manifolds . . . . .	15
<b>3</b>	<b>Distributed sensor failure detection in Sensor Networks</b>	<b>19</b>
3.1	Introduction . . . . .	19
3.2	Centralized detection with probabilistic Group Testing . . . . .	21
3.3	Proposed Distributed Detection . . . . .	22
3.3.1	Sensor network message design and dissemination . . . . .	22
3.3.2	Detection of one defective sensor in the network . . . . .	25
3.3.3	Detection of multiple defective sensors in the network . . . . .	29
3.3.4	Probability of at least two defective sensors in a cluster . . . . .	30
3.4	Experimental results and discussion . . . . .	32
3.4.1	Simulation setup . . . . .	32
3.4.2	Detection performance . . . . .	32
3.4.3	Influence of the master node selection process . . . . .	37
3.4.4	Communication overhead . . . . .	39
3.5	Conclusions . . . . .	41
<b>4</b>	<b>Distributed data gathering and reconstruction</b>	<b>43</b>
4.1	Introduction . . . . .	43
4.2	General decoder performance bounds . . . . .	46
4.3	Coding matrix analysis . . . . .	50
4.3.1	Matrix rank . . . . .	51
4.3.2	Disjunct and separable matrices . . . . .	53
4.3.3	Coding matrix design with a given signal pmf . . . . .	54

---

4.3.4	Probabilistic encoding matrix design . . . . .	56
4.4	Modified Belief Propagation reconstruction algorithm . . . . .	57
4.5	Simulation results . . . . .	61
4.5.1	Simulation setup . . . . .	61
4.5.2	General decoder performance bounds . . . . .	61
4.5.3	Layer BP performance . . . . .	63
4.6	Conclusions . . . . .	67
<b>5</b>	<b>Distributed interpolation of sensor network signals on manifolds</b>	<b>69</b>
5.1	Introduction . . . . .	69
5.2	Nonparametric weighted kernel regression on geometric manifolds . . . . .	71
5.3	Application to spherical signals . . . . .	76
5.3.1	Interpolation on the sphere . . . . .	76
5.3.2	Experimental interpolation results . . . . .	78
5.4	Iterative nonparametric weighted kernel regression on geometric manifolds . . . . .	86
5.4.1	Interpolation algorithm . . . . .	86
5.4.2	Analysis of data transmission phases . . . . .	88
5.4.3	Data dissemination analysis . . . . .	94
5.4.4	Experimental measurement transmission analysis . . . . .	96
5.5	Conclusions . . . . .	98
<b>6</b>	<b>Conclusions and Perspectives</b>	<b>101</b>
6.1	Limitations . . . . .	102
6.2	Future Work . . . . .	103
<b>A</b>	<b>Appendix</b>	<b>105</b>
A.1	Model for probability $P(q m)$ . . . . .	105
A.2	Ring properties . . . . .	105
A.3	The distribution of the product of two discrete random variables . . . . .	106
A.4	Disjunct and separable matrix definitions . . . . .	106
A.5	Taylor multivariate expansion . . . . .	107

# List of Figures

1.1	An ad-hoc sensor network acquires the values of the underlying physical phenomenon.	2
2.1	Illustration of data transmission centralized (left) and distributed (right) sensor networks. In centralized networks, a single node $s_0$ , denoted as a fusion centre, collects data from sensors and performs signal analysis. Sensors in distributed systems transmit their measurements in adhoc way and very often perform computations at sensors (so called “in-network” processing) to distribute the computational load. . . . .	8
2.2	Illustration of “push” (left) and “pull” (right) gossip transmission protocols. Left: each sensor randomly chooses one sensor to whom it transmits the message. Right: each sensor randomly chooses the sensor from whom it requests the message. . . .	9
2.3	Illustration of the Group Testing principle: the outcomes of carefully designed tests are sufficient to detect the defective element and the total number of tests is smaller compared to the scheme when each of the elements is tested separately. In the example, the goal is to detect the rotten bean bag marked with the red square by tasting three dishes. The rotten bean bag is uniquely detected given the tasting outcomes if the combination of bags used for cooking the dishes follows a particular design. . . . .	10
2.4	Illustration of sensor network distributed irregularly on a geometric manifold $\mathcal{M}$ . .	15
3.1	Noise effect on binary symbols in the test message: this model assumes that only the non-zero values in the test matrix are flipped with probability $(1 - p)$ to the value zero. . . . .	22
3.2	Illustration of the message design and dissemination through the sensor network. (a) Message formation based on local sensor measurements: Full and dashed arrows correspond to the steps of the message design, respectively. In the first step, the master sensor collects the sensor measurements from its neighbor sensors $\{s_1, \dots, s_4\}$ and forms the message $(g_l(t^-), \mathbf{W}_{l,:}(t^-))$ . In the second step, the message is propagated from the master sensor to its neighbor sensors. (b) Message dissemination based on a gossip algorithm with pull protocol, where the sensors request the messages from their neighbors chosen uniformly at random. (c) Rounds of communication in our iterative detection algorithm consist of the message design ( $t_I$ ) and the message dissemination ( $t_{II}$ ) phases. . . . .	23

3.3	The message formation at sensor $s_2$ in round $t$ . We assume that $s_2$ pulls the sensor $s_1$ to send its message, created in the previous round (round $(t-1)$ ). We assume that the sensor $s_3$ is defective $\mathbf{f} = [0010\dots]$ . The outcome value and the test identifier vector are formed by bitwise OR operator. . . . .	24
3.4	Simulation results for fully connected (FG), $k = 16$ -regular connected (RG, $k = 16$ ) and $k = 6$ -connected graphs (RG, $k = 6$ ) and sensor participation constant $\alpha = qK$ with $S = 20$ sensors, $K = 1$ and a random selection (RM) of $L = 5$ master sensors: (a) Probability of defective sensor detection; (b) Average rank of messages received per sensor. . . . .	33
3.5	Probability of defective sensor detection; Simulation results for irregular graphs ( $k > 3$ ) and random selection (RM) of (Top $S = 20$ ) and (Bottom $S = 70$ ) sensors, $K = 1$ . (a) $L = 5$ master sensors; (b) sensor participation constant $\alpha = qK = 0.7$ . . . . .	34
3.6	Probability of defective sensor detection for fully connected (FG) and irregular graphs (IG), $d > 3$ with $S = 70$ sensors, $K = 2$ and deterministic selection (DM) of master sensors. (a) fixed number of clusters $L = 10$ ; (b) fixed sensor participation constant $\alpha = qK = 0.3$ . . . . .	35
3.7	Detection performance for networks with (top $S = 20$ ) and (bottom $S = 70$ ) sensors and $L = 5$ master sensors. Abbreviations: GP: Proposed method, RWGP: Random Walk gossip dissemination algorithm with pull protocol, RW: Random Walk in the network initiated at $L$ sensors. (a) fully connected sensor network; (b) irregular sensor network. . . . .	36
3.8	Average rank value for irregular sensor networks with $L = 5$ master sensors: (a) $S = 20$ sensors (b) $S = 70$ sensors. Abbreviations: GP: Proposed method, RWGP: Random Walk gossip dissemination algorithm with pull protocol, RW: Random Walk in the network initiated at $L$ sensors, SF: pull store-and-forward algorithm with a random message transmission, GSF: pull store-and-forward algorithm with a greedy message transmission. . . . .	37
3.9	Simulation results for fully connected graphs with $S = 20$ sensors, $K = 1$ , where RM and DM denote the random and deterministic selection mode of master sensors, respectively. Top row: Probability of defective sensor detection. Bottom row: Average rank of messages received per sensor. Column (a): fixed values of the master sensors ( $L = 5$ ). Column (b): fixed values of the sensor participation constant ( $\alpha = qK = 0.7$ ). . . . .	38
3.10	Simulation results for fully connected (FG) and irregular graphs (IG), $d > 3$ with $S = 20$ sensors, $K = 2$ and deterministic selection (DM) of $L = 5$ master nodes: (a) Probability of defective item detection; (b) Average rank value. . . . .	39
3.11	Simulation results for fully connected (FG) and irregular graphs (IG), $d > 3$ with $S = 20$ sensors, $K = 2$ and deterministic selection (DM) of master nodes, $\alpha = 0.3$ : (a) Probability of defective item detection; (b) Average rank value. . . . .	39
3.12	(a) Comparison of the communication overhead for several algorithms, for the following parameter values: $(S, L, L_n, \alpha, R_d, I_d, \tau) = (70, 5, 50, 0.7, 7, 7, 80)$ . Graph is fully connected. Abbreviations: GP: Proposed method, RWGP: Random Walk rounds with gossip algorithm and pull protocol dissemination, RW: Random Walk in the network initiated at $L$ sensors. (b) Comparison of detection vs. number of rounds of the distributed detection scheme. . . . .	41

4.1	Directed sensor network. Parents $\mathcal{P}_{s_i}$ of sensor $s_i$ are sensors positioned farther than $s_i$ from the receiver $r$ . . . . .	44
4.2	To simplify the error event probability calculation, we split the analysis into the cases depending on the values of the encoding matrix elements that multiply the signal elements at the positions where the vectors $X$ and $\hat{X}$ differ. Here, $A$ is the number of nonzero coding matrix coefficients that multiply the elements of vectors $X$ and $\hat{X}$ in positions where they differ. The value $B$ is the number of zero encoding coefficients that multiplies the nonzero elements of the vector obtained by subtracting the vectors $X$ and $\hat{X}$ . . . . .	48
4.3	The partition function values $P(a, b)$ for the pairs of values $a$ and $b$ , where $a, b \in \{1, \dots, 20\}$ define the number of representations of the number $a$ with $b$ summands. Equivalently, they define the number of representations of the number $a$ with an arbitrary number of summands, where the maximum summand is equal to $b$ . These values form the lower triangular matrix. . . . .	49
4.4	Dependence of the partition functions and their ratio $\frac{P((m-1)q, A, n)}{P((m-1)q, A)}$ from the set of parameters $(A, m)$ , plotted for the set of parameters $(K, q) = (5, 8)$ . Note that $A$ defines the number of nonzero summands in the partition function and $m$ defines the value of number to be partitioned (equal to $(m-1)q$ ). The value $m = 3$ ( $m-1 = 2$ ) defines the real field interval with the highest values of the partition function in (a) and the ratio in (c). . . . .	53
4.5	Dependence of the partition functions and their ratio $\frac{P((m-1)q, A, n)}{P((m-1)q, A)}$ from the set of parameters $(A, n)$ , plotted for the set of parameters $(K, q) = (5, 8)$ . The parameter $A$ defines the number of nonzero summands in the partition function and $n$ determines the maximum summand value used for partitioning the number $(m-1)q$ . The error occurs most probably for $n = q - 1$ , which is the highest value that $n$ can take in a ring $F_q$ of size $q$ . . . . .	54
4.6	The structure of Belief Propagation algorithm denoted used for signal reconstruction.	58
4.7	The topology correlation model is based on the assumption that the sensors which are locally positioned may have correlated values. The neighborhood set $\mathcal{N}_{s_j} = \{s_i, s_k, r\}$ of the sensor $s_j$ build sensors positioned within a radius $\rho$ from the sensor $s_j$ . In this illustration $dist(s_i, s_k) < \rho$ and $dist(s_i, r) > \rho$ . Thus the collection of clique sets in this neighborhood corresponds to a union of the set of single elements $\mathcal{C}_1 = \{s_i, s_j, s_k, r\}$ , pairs $\mathcal{C}_2 = \{(s_i, s_j), (s_j, s_k), (s_j, r)\}$ and triplets $\mathcal{C}_3 = \{(s_i, s_j, s_k), (s_k, s_j, r)\}$ . The pair $(s_k, r)$ is not correlated in the considered neighborhood, since it does not contain the sensor $s_j$ . . . . .	59
4.8	Error decoding probability (log scale) for line array sensor network with parameters $(S, K, q, \text{sparse signal distribution})$ . PM: proposed method bound; DR: Draper et al. bound (a) PM and DR are calculated using parameter values $(20, [2, 3, 5], 7, \text{Uniform})$ . (b) PM and DR are calculated using parameter values $(20, 2, [7, 17], \text{Uniform})$ . . . .	62
4.9	Error decoding probability (log scale) for line array and tree-based sensor network with parameters: $(S, K, q, \text{sparse signal model}) = (20, 5, 8, \text{DiscreteLaplacian})$ . The error is calculated for sparsity of $K$ and up to $K$ . . . . .	63

4.10	Performance analysis for LA networks with parameters $(S, K, N, q, p_w) = (20, 2, 10, 8, 0.2)$ . (a) Average recovery probability; (b) Average Mean Square Error; (c) Bit consumption; (d) Average energy consumption. . . . .	64
4.11	Performance analysis for TR networks with parameters $(S, K, N, q, p_w) = (20, 2, 10, 8, 0.2)$ . (a) Average recovery probability; (b) Average Mean Square Error; . . . . .	65
4.12	Bit consumption for TR network with parameters $(S, K, N, q, p_w) = (20, 2, 10, 8, 0.2)$ .	66
4.13	Theoretical and experimental values of recovery probability. The considered setup is defined by the set of parameters $(S, K, N, q, p_w, \text{signal prior}) = (20, 2, 10, 8, 0.2, \text{Discrete Laplacian})$ . . . . .	66
5.1	The graph representation of the irregular sensor network on a geometric manifold $\mathcal{M}$ is defined by a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ , where sensors $s_i$ , $1 \leq i \leq S$ build the set of vertices, their connections $e_{i,j} = (s_i, s_j)$ , define the edges $\mathcal{E}$ whose weights are $w(s_i, s_j)$ . The set $\mathcal{N}_{s_i}^{(g)}$ defines a set of sensors that are connected to $s_i$ ( $w(s_i, s_j) \leq \rho$ ), while the local sensor neighborhood $\mathcal{N}_{s_i}$ determines the set of nodes whose geodesic distance to $s_i$ is smaller than $\sigma$ . . . . .	70
5.2	Geometry of a 2-d sphere: spherical polar coordinates $(\theta, \phi)$ . . . . .	76
5.3	The inverse stereographic projection [1]: a point with the Cartesian coordinates $(x, y)$ in the plane tangent to the North pole is mapped onto the sphere and its coordinates become $(\theta, \phi)$ . . . . .	77
5.4	Omnidirectional test image. . . . .	79
5.5	PSNR results for sets of $16 \cdot 10^3$ and $33 \cdot 10^3$ interpolation points in presence of additive Gaussian noise, $\sigma = 0.07$ . . . . .	81
5.6	Original spherical images (unwrapped) are down-sampled to dimension $256 \times 256$ and their intensity values are rescaled to the range $[0,1]$ . . . . .	81
5.7	Results for the Chessboard smooth image data. Top row: The proposed IM method, parameter $\sigma = 0.03$ . Bottom row: The <i>Nearest Neighbor</i> method: the parameter used in simulation is in the range $\{1, 2, 3\}$ , optimized for each interpolation point. (a) SNR=10 dB, (b) SNR=15 dB, (c) SNR=20 dB, (d) SNR=25 dB. . . . .	82
5.8	Results for the Earth Topo image data with 5% of missing data. Top row: Proposed method (IM, $n = 2$ ). The value $\sigma$ used in (a) and (b) is $\sigma = 0.03$ and for (c) and (d) $\sigma = 0.02$ . Bottom row: The <i>Nearest Neighbor</i> method uses 10 neighbor values for interpolation. (a) SNR=10 dB, (b) SNR=15 dB, (c) SNR=20 dB, (d) SNR=25 dB. . . . .	83
5.9	Results for the irregular Room image data set. Top row: <i>Proposed method (IM, <math>n = 2</math>)</i> , $\sigma = 0.07$ . Middle row: <i>Proposed method (IM, <math>n = 0</math>)</i> , $\sigma = 0.01$ . Bottom row: The <i>Nearest Neighbor</i> method uses optimized parameter value. (a) SNR=10 dB, (b) SNR=15 dB, (c) SNR=20 dB, (d) SNR=25 dB. . . . .	84
5.10	Results for the regular Room image data set. Top row: <i>Proposed method (IM, <math>n = 2</math>)</i> . Middle row: <i>Proposed method (IM, <math>n = 0</math>)</i> . Bottom: The <i>Nearest Neighbor</i> method uses optimized parameters. (a) SNR=10 dB, (b) SNR=15 dB, (c) SNR=20 dB, (d) SNR=25 dB. . . . .	85



5.11	Schematic illustration of the measurement dissemination process that is based on a gossip algorithm with pull protocol. Every sensor requests one measurement from one of its neighbor, chosen uniformly at random. . . . .	86
5.12	Schematic illustration of the two main dissemination cases. The standard deviation of a kernel defines the interpolation data set as the set of sensors $\mathcal{N}_{s_0}$ that are positioned at a distance smaller than $\sigma$ from the interpolation sensor $s_0$ . (a) Sensors within a distance $\sigma$ can directly communicate with the interpolation sensor. (b) The range $\sigma$ is larger than the maximal sensor communication range, so that adhoc transmission of measurements is adopted. To simplify the measurement dissemination analysis, the sensors are divided in sets $\mathcal{H}_j$ of cardinality $h_j$ , $j \in \{1, \dots, J\}$ , based on the number of the transmission rounds $j$ required for the sensor measurement to reach the interpolation sensor, using the shortest transmission path. . . . .	89
5.13	Schematic illustration of a general discrete phase type distribution, defined by the absorbing Markov chain with one absorbing state $\mathcal{A}$ . Each of the states of the Markov chain $\{1, 2, \dots, s^*\}$ represents one of the phases. We use it to model the random variable $T_r$ that describes the number of rounds that pass until the interpolation sensor that collected $r$ sensor measurements receives a novel measurement that is different to the previously collected values. . . . .	92
5.14	Schematic illustration of probabilities of the state $s_1$ in Markov chain with one absorbing state. . . . .	93
5.15	Mean square error of interpolation decreases over the transmission rounds, $( \mathcal{N}_{s_i} , \sigma) = (\{42, 54, 69\}, \{0.05, 0.06, 0.07\})$ : (a) most of the sensors are positioned within the communication range $\rho$ ; (b) a small subset of sensors is positioned within the communication range $\rho$ . . . . .	97
5.16	Average rank values of the sensors vs. the number of transmission rounds: (a) most of the sensors are positioned within the communication range $\rho$ ; (b) a small subset of sensors is positioned within the communication range $\rho$ . . . . .	97
5.17	Total bit consumption for different number of transmission rounds for cases: (a) most of the sensors are positioned within the communication range $\rho$ ; (b) a small subset of sensors is positioned within the communication range $\rho$ . . . . .	98
A.1	Properties of structures in algebra. . . . .	106

# List of Tables

3.1	Theoretical measurement rounds requirements for networks with $S$ sensors. . . . .	35
5.1	MSE for different values of the standard deviation parameter in the noiseless case. The number of interpolation points is $16k$ . The parameters of the NN method are optimized to provide minimal MSE (five neighbors). . . . .	80
5.2	MSE in the noisy cases with different SNRs. The number of sample points is $16 \cdot 10^3$ and the standard deviation is fixed to $\sigma = 0.0069$ . The parameters of the classical NN method are optimized to provide minimal MSE (five neighbors). . . . .	80
5.3	The structural similarity measure (SSIM) values for interpolation of noisy spherical signals. Kernel and NN parameters are optimized for each of the datasets. . . . .	82

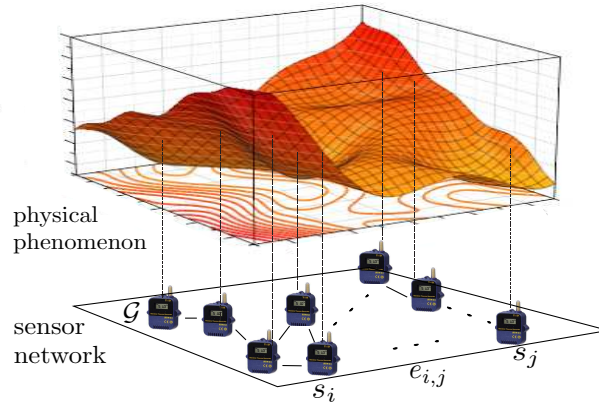
# Chapter 1

## Introduction

### 1.1 Distributed sensor processing in sensor networks

With the advent of new sensing technologies and architectures, we face now important novel challenges imposed by large-scale sensor networks. The decrease of costs for sensor devices with independent battery on one side, and the availability of untethered protocols for communication on the other side have fostered a wide deployment of sensor networks in different applications. Nowadays, the analysis of complex processes and phenomenas can be performed using data gathered by sensor networks (Fig. 1.1), for e.g., in medical, industrial or environmental monitoring, to name a few. The development of these new applications creates a demand for efficient and accurate data processing algorithms. This leads to important challenges due to numerous constraints in sensor network systems: limitations of sensor power or network capacity, topology and connectivity changes, sensor failure or sensing noise, to name a few. This has further led to intense research efforts towards the design of effective distributed data processing methods. Novel algorithms need to deal with problems specific to large-scale networks, such as accurate and efficient data collection and estimation. Even if the noise effects and delays caused by data collection may be tolerable, it is often expensive to transmit the full set of raw data to a control entity. Additionally, transmission noise may degrade sensor measurements, or more severe, they may be lost because of the sensor failures. This necessitates the development of robust distributed algorithms with low computational complexity in the sensors. Hence, the design of efficient and flexible methods for networks that jointly perform the processing tasks and preserve the system resources becomes of particular importance for modern sensor systems.

In this thesis, we investigate different signal processing problems, such as detection, reconstruction and interpolation, challenges for distributed processing in sensor network [2]. We present novel solutions to sensor network signal estimation from incomplete, noisy or irregular data that is gathered distributedly. To build such algorithms, we exploit the correlation of data collected by neighbor sensors and we use, when possible, network coding mechanisms and finite field modulo operations at sensors to represent data with a limited number of bits. In particular, we study new distributed detection methods for defective sensor identification, which enhance network robustness and further improve signal estimation performance. Furthermore, we propose a new distributed data gathering algorithm with reconstruction at receiver given a limited number of quantized net-



**Figure 1.1:** An ad-hoc sensor network acquires the values of the underlying physical phenomenon.

work readings along with signal statistics. Finally, we investigate the distributed data interpolation problem performed at sensors that are positioned on a smooth manifold, when interpolation is done at arbitrary locations.

In more details, we focus in this thesis on several instances of distributed problems for sensor networks. In networks, noise deteriorates the data and potentially masks the outcome. Hence, a very important research problem consists in the design of robust algorithms that can detect defective sensors in the network and eliminate their measurements from the processing data set. Therefore, we first consider the problem of distributed failure detection in sensor networks. Classical detection approaches mainly depend on the network topology and often perform hierarchical decision building. The decisions of groups of sensors are merged together into a final decision about the identification of defective sensors in the network. These approaches are efficient in small networks with fixed topologies; however, they become highly inefficient if the network is susceptible to topology changes. We rather propose a robust distributed detection algorithm based on probabilistic tests, which leads to accurate detection of failure sensors, located at any arbitrary point in the network.

Another challenging problem is the data reconstruction from incomplete set of network observations. The class of problems for signal recovery given incomplete set of data is well known as the family of inverse problems. A common strategy to solve this type of problems consists in solving the convex optimization problem, where additional signal information, such as smoothness priors, is available. However, sensor devices have a limited battery power, so that messages need to be quantized and coded, which completely changes the problem of data reconstruction. A natural way to limit the number of bits that are used in message representation is to perform modulo operations at sensors. Novel tendencies in sensor development boost *multiple valued logic* at sensors, motivated by the pioneering works [3], [4]. In these scenarios, sensors perform modulo computations in finite fields whose base is higher than two in order to achieve higher power efficiency. We propose here a novel framework where sensors perform modulo operations in a field of arbitrary size. Then, the data reconstruction problem at receiver can not be solved by solutions common to inverse problems, because operations at sensors are non-linear. Classical distributed inverse algorithms such as [5]

are developed for real field operators. We propose a novel data reconstruction algorithm, which leads to effective performance by properly exploiting a small number of network coded messages along with signal priors at decoder.

Next, the problem of in-network interpolation tries to estimate function values at positions where none of the sensors is physically located. Sensors in networks are commonly irregularly distributed on a geometric manifold. Because of the large network coverage, the inherent manifold structure built by sensors can not be neglected. Commonly, the set of interpolation points is not known before data acquisition in sensor networks. Efficient existing methods for high dimensional and noisy data interpolation, known as graph-based variational methods [6], are however incapable of dealing with manifold data interpolation problems in cases where interpolation points are not known prior to the signal acquisition. Hence, we propose a new distributed interpolation algorithm which builds local estimates of the global network function, whose accuracy improves when more messages become available in the sensor neighborhood.

Overall, we propose in this thesis novel solutions to important instances of distributed signal processing problems in sensor networks. We present our proposed distributed sensor failure detection method that is based on the particular probabilistic message design and binary sparse failure signal assumption. It determines failure sensors by a simple distance decoder with high probability, given a small number of messages. Then, our new data reconstruction solution is based on the message passing algorithm that recovers the signal given its priors and incomplete messages, when messages are coded by combining the quantized measurements in finite fields of arbitrary size. We finally present a new distributed interpolation solution for arbitrary sets of points by solving a new nonlinear kernel regression optimization problem for noisy irregular manifold data. This thesis provides important insights in the design of large-scale sensor network architectures, with numerous applications.

## 1.2 Thesis outline

The thesis is organized as follows. We first provide an overview of the existing literature on topics studied in this thesis in Chapter 2. In particular, we discuss the related work in areas like detection, reconstruction and interpolation.

We then address in Chapter 3 the problem of **sensors' failure detection** in networks with a small number of defective sensors, whose measurements differ significantly from the neighbor measurements. We build on the sparse nature of the binary sensor failure signals to propose a novel distributed detection algorithm based on gossip mechanisms and on Group Testing (GT) ideas. Our new distributed GT algorithm estimates the set of scattered defective sensors with a low complexity distance decoder from a small number of linearly independent binary messages exchanged by the sensors. We first consider networks with one defective sensor and determine the minimal number of linearly independent messages needed for its detection with high probability. We then extend our study to the detection of multiple defective sensors by modifying appropriately the message exchange protocol and the decoding procedure. We show that, for small and medium size networks, the number of messages required for successful detection is actually very small. Finally, simulations demonstrate that the proposed method outperforms methods based on random walks in terms of both, the detection performance and the convergence rate.

Further on, in Chapter 4 we address the problem of **efficient distributed data gathering with reconstruction from quantized and incomplete messages**. We propose a distributed gathering method that encodes messages in finite fields of arbitrary size. A network message transmission starts at sensors located on borders of the network. Messages are transmitted distributedly along a directed path towards the receiver and their values are updated at sensors during the message transmission. At every sensor in the network, our novel algorithm forms a new message by combining the measurement of the sensor with the messages received from its neighbors. Weighted combinations are performed using arbitrary size finite field operations. We then study the resulting signal reconstruction problem, given the partial data collected by the proposed gathering framework. We analyze the decoding error probability for signal classes that have locally correlated values. We further investigate the properties of probabilistic coding matrices used in the network coding operations. Finally, we provide a constructive data reconstruction algorithm based on a Belief Propagation method. The results of the theoretical and experimental analysis demonstrate that the proposed scheme effectively gathers data in sensor networks and recovers them with high probability when simple signal priors are available at the receiver.

Then we study in Chapter 5 **distributed signal interpolation** for irregularly positioned sensors on a geometric manifold. We propose a novel interpolation algorithm that efficiently computes interpolation values for any set of points that are not given a priori. We then estimate the parameters of an unknown local function from available data. Once the function parameters are known, the interpolated values become functions of manifold distances to the sensor that performs interpolation. Our novel interpolation algorithm efficiently exploits geometry information and copes with irregular and noisy signal samples. We provide reconstruction results of the proposed manifold interpolation algorithm for two types of spherical manifold data, namely, catadioptric image and synthetic datasets with different sample distributions. We provide simulation results for noiseless and noisy environment that demonstrate higher accuracy of the proposed solution. In particular, our algorithm outperforms the Nearest Neighbor (NN) interpolation algorithm, both in terms of classical and perceptually oriented signal fidelity measures. We further study the convergence of our interpolation algorithm and show that the estimation performance gracefully increases with the number of neighbors' measurements that become available to the interpolating sensor.

Finally, concluding remarks and future perspectives are given in Chapter 6.

### 1.3 Summary of contributions

The main contributions of this thesis are summarized below.

- We solve the problem of distributed sensors' failure detection in networks, where every sensor may perform detection. This is the first fully distributed algorithm for detection of defective sensors. Failure detection ensures the robustness of networks and enhances the signal quality.
- We propose a novel efficient distributed data gathering method for signal reconstruction from incomplete and quantized network messages, given simple signal priors at decoder. In this method, sensors perform nonlinear modulo operations to combine quantized network data, for more efficient data transmission, which represents a novel form of data collection in networks.

- We propose a new distributed interpolation method for sensors data on geometric manifolds. Sensors locally solve a regression problem that incorporates manifold geometry, given the noisy data from sensors that are distributed irregularly on a geometric manifold. This represents a unique distributed interpolation algorithm for estimating data at a priori unknown locations on a manifold.





## Chapter 2

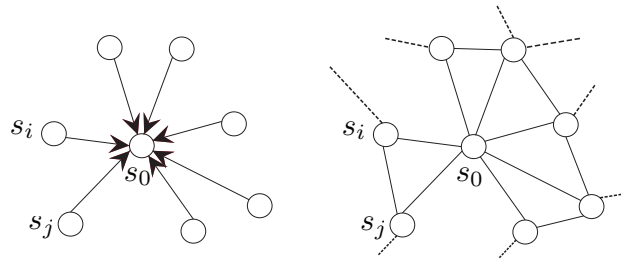
# Related Work

In this chapter, we first review the methods commonly used to improve the efficiency of distributed sensor networks, and then, we review the literature that is closely connected to general problems addressed in this thesis. First, in Section 2.1 we overview the methods for efficient data dissemination in networks, as well as the distributed source coding methods that reduce the redundancy in the network. Next, we overview methods which improve the network throughput. In Section 2.2, we provide a literature review of state of the art works on general detection problems, and we briefly provide its connections with coding theory. Next, in Section 2.3 we focus on literature that studies data gathering methods in sensor networks. In cases when the receiver gathers a partial data set, the decoder deals with an inverse problem. We discuss the classical and novel inverse problem approaches and we overview the recent inverse problem works in finite field spaces. Finally, in Section 2.4 we review the most common interpolation methods that are applicable to data acquired by sensors distributed on a smooth manifold. We note that each of the following chapters includes more detailed discussion of the works that are closely related to the methodologies proposed in this thesis, where required.

### 2.1 Network mechanisms

We discuss a few common methods used in this thesis that increase the overall network efficiency; in particular, we start with motivations for using distributed data gathering and processing rather than their centralized counterparts. Then, we discuss several common methods for efficient data dissemination that reduce the redundancy in the network and improve its throughput. Further, we provide the overview of efficient message dissemination algorithms. Finally, we review mechanisms which improve the network data throughput and increase the probability of forming innovative messages in the network.

**Distributed processing in sensor networks** The first generation of sensor networks consisted of a small number of sensors whose role is to capture and communicate data in a raw form directly to the central node. This node, often denoted as the *fusion center*, is responsible of analysis tasks of the signal, Fig. 2.1 (left). Nowadays, with the increase of the scope of sensor networks, raw data transmission to the fusion center becomes unpractical [7]. It is more efficient to split and share the computation load amongst sensors in the network in terms of communication and sensor



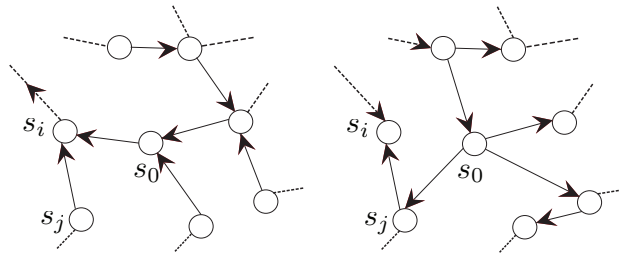
**Figure 2.1:** Illustration of data transmission centralized (left) and distributed (right) sensor networks. In centralized networks, a single node  $s_0$ , denoted as a fusion centre, collects data from sensors and performs signal analysis. Sensors in distributed systems transmit their measurements in adhoc way and very often perform computations at sensors (so called “in-network” processing) to distribute the computational load.

battery power. Such algorithms belong to a class of so called *distributed* algorithms [8] and they eliminate the need to transmit the raw data to the central node, Fig. 2.1 (right). They perform “in-network” computation and/or adhoc data transmission. For more detail and examples of distributed optimization algorithms, such as distributed classification, robust estimation or clustering refer to [9] and references within. However, the challenges imposed by distributed processing are numerous [2], as shown in the rest of this thesis for the problems of detection, reconstruction and interpolation in sensor networks.

**Random dissemination algorithms** To perform data transmission in the network, sensors collaboratively follow a certain protocol. A class of protocols for data spreading in distributed networks, which is modeled by mimicking the distribution of social network gossips [10], uses less communication resources than a simple broadcasting method (flooding). They are often denoted as *gossip (rumor)* or *epidemic* protocols [11], [12], [13]. These models were used for maintenance of replicated databases [14] and they became popular in applications related with data transmission. Problems of fast and robust distributed message spreading in networks, where the total number of messages is smaller than the number of sensors in the network is studied in [15] for a complete graph. The principle of random combination of messages along with their transmission, known as the *network coding*, is used for achieving higher robustness to packet losses. The work in [16] provides an analysis of information dissemination via network coding for arbitrary graphs by connecting the information dissemination with the spectral properties of the underlying graph topology. More recently, such algorithms are used for distribution of signal processing tasks. In particular, distributed linear parameter estimation, source localization and distributed compression are studied in [17].

To minimize the costs in networks and achieve flexible algorithms, distributed algorithms commonly use noncooperative protocols for data dissemination. In this thesis as a data transmission protocol we use the gossip transmission protocol with so-called “push” gossip scheme as a data transmission protocol, illustrated in Fig. 2.2. For detail on advantages of the “push” scheme compared to the “pull” one, refer to [11] (Sec. 2). In random regular graphs, the effect of the node degree on rumor spreading for “push” gossip mechanisms is analyzed in [18], where the authors provide the probabilistic analysis for message distribution to all the nodes.

**Distributed Source Coding** In the ideal case, message transmission is exclusively reserved



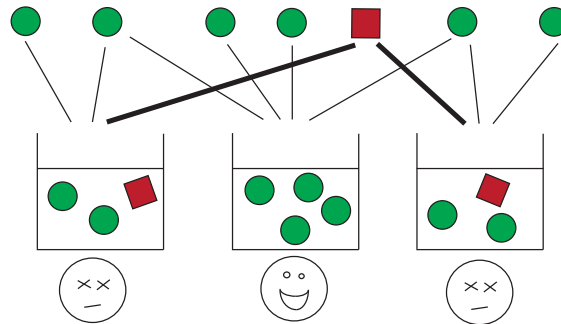
**Figure 2.2:** Illustration of “push” (left) and “pull” (right) gossip transmission protocols. Left: each sensor randomly chooses one sensor to whom it transmits the message. Right: each sensor randomly chooses the sensor from whom it requests the message.

for the innovative or critical information. To extract such an information, sensors should have knowledge of the local signal values, which is often not the case. A Distributed Source Coding (DSC) [19], [20] is commonly deployed to reduce the redundancy in networks. DSC is a signal compression method for correlated signals. Signal encoding is performed independently, while their decoding is performed jointly. DSC shifts the computation complexity from sensors towards the decoder side, which is important for network energy preservation in the network nodes. Unfortunately, DSC is difficult to implement in fully distributed systems because the sensors generally do not have a priori knowledge required for identification of data correlation.

**Network Coding** The Network Coding (NC) method [21], [22], [23], is a common method for improving the network throughput. It employs linear recombination of several messages available at network nodes and creates a novel message. The novel message is transmitted to the neighbor sensor, which further uses it to form novel messages in the following transmission rounds. The receiver decodes the messages only after receiving the information from all the sensors. Compared to the networks that simply forward information towards neighboring nodes, these networks have a high degree of robustness. Recently, DSC and NC techniques are jointly used for efficient data gathering in [24]. Correlated source signals are quantized, mapped onto the codewords and transmitted in the network using the NC principles. Signals are decoded jointly at the receiver using the Gaussian elimination method.

## 2.2 Detection of defective sensors

In general, the goal of detection is to determine a data set with different characteristics from the expected ones, for e.g., the outlier detection, or to identify a rare event from available data samples, like for natural disaster detection. This problem is common for many research fields, so numerous detection methods are proposed in the studies. One of the possible ways to classify the detection methods is based on the employed decoder in the method. Based on this criteria, we distinguish methods with: (i) statistical decoders [25] and (ii) detection based on Group Testing. We provide basic information on the class of statistical decoders, and focus on the second class of methods that is closely related to our problem. Detection algorithms generally employ statistical decoders [25] and often assume that the signal probability distribution and the network connectivity are known. For example, a Bayesian approach in [26] proposes to compute a detection score for a priori defined



**Figure 2.3:** Illustration of the Group Testing principle: the outcomes of carefully designed tests are sufficient to detect the defective element and the total number of tests is smaller compared to the scheme when each of the elements is tested separately. In the example, the goal is to detect the rotten bean bag marked with the red square by tasting three dishes. The rotten bean bag is uniquely detected given the tasting outcomes if the combination of bags used for cooking the dishes follows a particular design.

sets of hypothesis, which depends on the received messages. The hypothesis with the highest score drives the final decision. The binary event detection problem for hierarchically clustered networks is proposed in [27] where the cluster decisions are fused centrally to make a final decision. Surveys on similar methods can be found in [28, 29].

**Group Testing** Group Testing (GT) is a general method applicable to testing large populations subject to the same characteristics, Fig. 2.3. It originates from the work [30] that targets detection in medical applications. In particular, it proposes a simple idea of pooling blood samples to observe the viral presence in a set, instead of performing tests on every single blood sample separately. Typically, the main target is to minimize the number of *tests* required to identify all the infected samples, while keeping the detection procedure as simple as possible.

GT has been studied more recently in the context of sensor networks for detection of malicious events. Interested readers are referred to the survey [31] and references within. In summary, the GT literature can be divided into two main algorithm types: (a) deterministic and (b) probabilistic. Both approaches aim to minimize the number of tests for detection. The combinatorial GT [32] uses a deterministic test design, however, purely deterministic test designs are inappropriate for dynamic structures such as sensor networks. The probabilistic GT applies the knowledge of probability distribution of defective elements for the purpose of detection and these methods provide flexibility required for network applications. Based on the type of the failure, detection approaches differ in scenarios with errors, inhibitors or of their combinations and they can be rather naive [33]. In some works detection is performed by iterative elimination of identified non-defective items from the test outcomes. The detection time is typically of the order  $\mathcal{O}(SB)$ , where  $B$  is the number of tests and  $S$  is the total number of sensors.

Finally, even if the test designs are contingent to the communication limitations in sensor networks, not many works have considered connectivity constraints imposed by the network topology in GT methods. The authors in [34] propose to form tests by a random walk process on well-connected graphs. The minimal number of tests required for detection in this case depends on the random walk mixing time. A bipartite graph structure is considered in [35] with a two-stage

hybrid detection method. Here, a subset of defective items in the first stage is determined by pre-designed tests, while the remaining items are tested individually in the next step. Data retrieval for topology-adaptive GT is studied in [36] where a binary tree splitting algorithm is proposed. The above methods however use centralized decision algorithms which are not appropriate for large-scale sensor networks or networks with a dynamic topology because of the high communication costs. Contrary to the above related work, the proposed method in this thesis focuses on a resource efficient distributed failure detection algorithm, applied for robust monitoring networks.

**Connections to coding theory** Particular probabilistic GT test design methods in centralized systems improve the effective time for detection. For example, a useful test matrix property called  $K$ -disjunctness (i.e., the Boolean sum of every  $K$  columns does not result in any other column in this matrix), speeds up the decoding process. This property is used in code designs, e.g., for superimposed codes [37], [38] or in detection of malicious users [39]. Finally, the work in [40] proposes to impose structures into disjunct matrices for fast decoding. For this purpose, they use random, independently generated codes. Non-linear inner codewords are chosen to be random binary vectors, where each entry is chosen to be one with a probability inversely proportional to the upper bound of the expected defectives. Authors use the random efficient construction of disjunct matrices. The decoding time of disjunct matrices is proportional to  $\mathcal{O}(\pi(B) \cdot B \log^2 B + \mathcal{O}(B^2))$ , where  $B = \mathcal{O}(K^2 \log S)$  is the number of tests,  $S$  is the number of sensors and  $\pi(\cdot)$  denotes a polynomial. In our knowledge, this represents the state-of-the-art decoding performance for centralized detection of an arbitrary sparse signal.

The previous literature on detection can be mostly classified into fully centralized and semi-distributed methods. In the latter, nodes are grouped in static clusters that independently perform failure detection. Moreover, we should emphasize that the term “distributed” in earlier detection works generally refers to the information transmission process between neighboring nodes towards the fusion center that performs detection.

In this thesis, we however consider the problem of distributed detection, where each sensor in the network is able to perform a detection task. In other words, motivated by works that address distributed signal processing tasks, we target a framework where nodes both transmit and process data in a distributed manner. We target the problem of distributed detection, where each of the sensors in the network is able to perform detection tasks. To the best of our knowledge, no former analysis has been proposed for distributed detection methods of sparse binary test signals as proposed in this thesis in Chapter 3.

## 2.3 Inverse problems for signal recovery

In this section, we review classical gathering methods in networks, as well as methods for data recovery from incomplete set of data.

**Data gathering methods** The extensive overview of state of the art gathering methods is given in the survey [41] and references within. These methods target the exact signal recovery. Data collection algorithms are generally driven by factors such as the sensors’ power and the network topology, for example. In the literature, efficient distributed data gathering methods for spatially correlated signal reconstruction perform data decorrelation along the collection path to decrease the number of bits transmitted in the network. The transmission and computation power consumed

for data gathering of correlated and piecewise smooth signals is minimized in [42]. Network is segmented into several groups of sensors. Sensors that belong to the same group perform data decorrelation. One sensor per group collects a resulting decorrelated group data and transmits it in ad-hoc way towards the receiver. Another ad-hoc gathering approach that uses wavelet lifting transforms on a network graph [43] minimizes a transmission of raw data in the network. It divides sensors into two groups, where the first sensor group performs a raw data transmission and the rest of the nodes transmit decorrelated signal. The predicted value is computed based on the values of its direct neighbors and decorrelated data is a result of a subtraction of the sensor value and its prediction. Different to our work, the receiver in the above methods reconstruct signals upon receiving a full rank of linearly independent messages.

In general, the classical gathering problem is posed as the constrained optimization problem, where the set of constraints are network dependent. For instance, rate allocation algorithms for data collection are proposed in [44], while Distributed Source Coding (DSC) algorithms [19], [20] are used in order to reduce data redundancy in the network. The reconstruction is successful when all sensor messages are available at the receiver. Different to these works, we study cases where partial network information gathered by the proposed method is sufficient for data reconstruction with high probability.

**Classical inverse problems** We provide details on the class of problems that aim to recover signals given partial observed data. In the literature, they are often denoted as the *variational methods*. Typically, these problems are ill-posed. To solve a problem, additional assumptions, such as smoothness or sparsity of the solution, are required. We briefly review the most common optimization problems, namely, Tikhonov regularization and Total Variation (TV).

(a) *Tikhonov regularization* We assume that the observed data  $\mathbf{y}$  is not complete, so the problem  $\mathbf{y} = \Phi \tilde{f}$  is ill-posed. In such scenarios, a classical Least Square approach that minimizes  $\|\Phi \tilde{f} - \mathbf{y}\|^2$ , where  $\|\cdot\|$  is the Euclidean norm, does not provide the unique solution. Instead, the regularization term that gives preference to a particular solution with desirable properties is included and the optimization problem becomes

$$\tilde{f} = \arg \min_f \|\Phi f - \mathbf{y}\|^2 + \|\Gamma f\|_2, \quad (2.1)$$

where  $\Gamma$  is a suitably chosen Tikhonov matrix [45], [46]. For e.g., a Tikhonov matrix is commonly chosen as the identity or difference operator, which respectively provides the small norm or smooth solutions. Such problems are solved by classical convex optimization tools, such as the gradient descent, for example.

The pioneering work [47] used in signal restoration is designed to remove noise from images while preserving their high frequency components (in image signals, this corresponds to edges). It solves the following optimization problem

$$\tilde{f} = \min_f \int_{\Omega} |\nabla f| du, \quad s.t. \quad \|f - \hat{f}\|_2^2 \leq \sigma^2, \quad (2.2)$$

where  $\hat{f}$  represents a noisy image in domain  $\Omega$ ,  $\tilde{f}$  denotes a resulting denoised image,  $|\cdot|$  stands for the Euclidean distance operator and  $\sigma^2$  is a noise variance estimate. Other works [48], [49], [50] that followed established different problem-driven regularization methods.

The development of discrete differential operators [51] initiated usage of regularization methods in discrete domains, like for instance in semi-supervised learning [52]. The discrete form of the above regularization problems is of special interest, because it admits the usage of fast and efficient discretized algorithms. In the literature, there is a number of solutions to discretized version of the TV regularization problem in Eq. (2.2), defined by  $\tilde{f} = \arg \min_f \{ \|\nabla f\| + 1/2\lambda \|f - \hat{f}\|^2 \}$ , where  $\lambda$  is a tradeoff between the regularization and the data term. It is very common to solve its dual representation due to simplified computation, for example, by the iterative recursive algorithms in [53]. This problem can also be solved by proximal iterations, given in [54]. Study [55] proposes several solutions that are competitive with a Chambolle's method, that are based on the gradient projection algorithms. Their algorithms offer several adaptive solutions to the tradeoff between the accuracy and the desired convergence time.

**Compressive Sensing in real and finite fields** We briefly review the classical compressive sensing, provide literature with applications to sensor networks and cite recent works in finite fields that are conceptually very closely related with the Compressive Sensing framework.

A novel recovery method for sparse or compressible signals known as the Compressive Sensing (CS) [56], [57], [58] guarantees signal recovery with high probability given a small number of measurements, obtained by projection with a matrix  $\Phi$  of certain property. A sufficient condition that the matrix of measurements  $\Phi$  has to fulfill for this purpose is to preserve the metric. This condition is called the Restricted Isometry Property (RIP) and it is expressed as:

$$(1 - \delta)\|\tilde{f}\|_2^2 \leq \|\Phi\tilde{f}\|_2^2 \leq (1 + \delta)\|\tilde{f}\|_2^2, \quad (2.3)$$

where the coefficient  $\delta \in [0, 1]$ , vector  $\tilde{f} \in \mathbb{R}^p$  and  $p$  is given for all index sets  $i \subset \{1, \dots, S\}$ .

Let the unknown signal  $\tilde{f}$  of dimension  $S$  be  $k$ -sparse or compressible. In other words, if  $a$  denotes a set of  $k$  coefficients and  $\Psi$  denotes a basis function set, the signal that has a representation  $\tilde{f} = a\Psi$  can be represented with  $k \ll S$  significant coefficients. The signal is then encoded by the known matrix  $\Phi \in \mathbb{R}^{m \times S}$  by  $\mathbf{y} = \Phi\tilde{f}$ , where the obtained measurement vector  $\mathbf{y}$  is of dimension  $m$ ,  $m \ll S$ . This compressed signal version is transmitted to the decoder. Since the signal prior is given, the decoder solves the optimization problem

$$\tilde{f} = \arg \min_f \|f\|_{p_1}, \text{ subject to } \mathbf{y} = \Phi f. \quad (2.4)$$

The solution to this problem is known to be NP hard for  $p_1 = 0$  so the relaxed version of the optimization problem is solved by  $p_1 = 1$ . The algorithmic solutions to the optimization problems are well studied and solved by linear programming.

CS algorithms for sensor networks [59] and [60] locally compute random measurements at sensors, transmit them to neighbors and aggregate them at the receiver, which performs the reconstruction. If some signals are jointly sparse, like for instance signals that differ in a few components, distributed CS algorithms which exploit both intra- and inter-signal correlation structures [61] have a superior performance to the non-distributed ones.

In these works, signals take values from a real field and sensors perform linear data combinations in a real field as well. In practice, sensor measurements are often quantized and operations at sensors are optimized to preserve a sensor battery. For this reason, different to the previous works, data manipulation in the proposed method in this thesis is performed in a finite ring of arbitrary size,

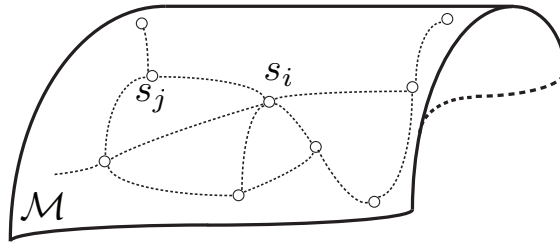
which restricts us from straightforward deployment of classical or CS algorithms.

A few recent works have analyzed finite field CS methods from the perspective of information theory. For example, the work [62] studies the error exponent value for sparse signal recovery given the partial signal observations, where signal values are elements of the finite field. These observations are the results of data combinations performed in a Galois fields (GF). The error exponent expression is developed under the assumption that the sampling matrix elements are i.i.d. and uniformly sampled from the Galois field. The problem of a low rank matrix recovery from its measurements given the matrix elements that take values from a Galois field is studied in [63], where the authors provide fundamental limits on sampling requirements and show that the decoding error bound is small for arbitrary matrices of a given rank. The binary GF inverse problems are the simplest finite field signal recovery problems. Group Testing (GT) [30] is a detection problem (see the previous section) in the Galois field  $GF(2)$ , where a pool of signal samples is jointly tested to minimize the number of performed tests. Decentralized GT approaches for sparse binary signals are studied in [34] and [64]. In these studies, message designs are motivated by the linear error correcting code designs [65], [66], [67]. In general, CS algorithms in finite fields are designed exclusively for Galois fields. To the best of our knowledge, no prior works analyze CS algorithms in a finite ring of arbitrary size.

**Coding matrix designs** When sensors take values from a limited and known set of input signals, the efficient gathering problem is dual to the coding matrix design problem. The construction of coding matrices for data gathering has similarities with the construction of probabilistic parity-check matrices for Low Density Parity Check (LDPC) codes. In general, parity check matrix values have binary values. Decoders that use higher order GF for LDPC code building however outperform their binary equivalents for binary Gaussian channels [67]. Also, the decoding complexity for finite alphabet elements is polynomial, as shown in [68]. In practice, Random Linear Codes (RLC) [69] are used in combination with NC to disseminate the innovative messages. Here, messages available at some sensor are linearly combined into a new message that is further transmitted in the network. Another class of matrices, called sparse coding matrices, has interesting properties from a decoder complexity point of view. A good overview of works on sparse coding matrices is given in [70] and references within, while more details on properties of sparse matrices in finite fields are provided in [71]. In the above works, decoding is performed when all the sensor measurements are observed by the decoder, which is not convenient for sensor network setups similar to ours.

Not many works address the gathering problem in finite fields. We emphasize that the existing works focus on gathering in Galois fields. We however consider the problem of gathering in a finite field of an arbitrary size and recovery from such data, where the size of the field may be adapted to the sensors' power. In other words, motivated by works of compressive sensing in Galois fields and techniques for distributed and random data decorrelation, we target a framework where sensors distributedly transmit and process data in arbitrary size finite fields. Our work specifically focuses on a resource-effective distributed gathering algorithm for signals with known priors, which can be deployed in robust monitoring networks.





**Figure 2.4:** Illustration of sensor network distributed irregularly on a geometric manifold  $\mathcal{M}$ .

## 2.4 Interpolation on geometric manifolds

The goal of interpolation is to estimate a set of function values at desired positions, given the noisy and irregular data. In classical signal processing, interpolation module is commonly deployed between the data acquisition step and the compression or analysis blocks. General interpolation problem has been studied over half of the century and numerous methods are proposed in the literature. However, the recent development of new application-driven sensing devices that capture signals on geometric manifolds imposed novel challenges in interpolation community. Interpolation of geometric manifold signal has to deal with its specific geometry and still to remain simple and efficient.

We distinguish the following two families for multivariate interpolation given a noisy scattered data on non-Euclidean manifolds. The first categorization is based on the regularity of the sampled data. We distinguish recovery given (a) regularly and (b) irregularly sampled data. The first case represents a classical case, studied for many decades. Classical Shannon's sampling theory provides effective solutions for signal reconstruction in shift-invariant spaces, where signals may be band limited. We provide details on recovery methods given irregularly sampled data. The second categorization is based on the model assumptions and we distinguish the works which use (a) meshing strategies [72] (e.g., Laplace, Poisson or thin-plate) and (b) mesh-free methods [73] (e.g., Shepard, MLS, NN, RBF) for signal recovery. The second family is closer to the method proposed in this thesis. We provide details on recovery given non-uniform or mesh-free interpolation methods and we connect the latter with methods in statistical learning, for learning the function from observed data.

**Non-uniform sampling** The challenging problem in sampling represents the reconstruction in shift-invariant spaces given non-uniform samples. The studies [74], [75] and [76] provide theoretical and practical solutions. The solutions are based on solving different types of variational problems to signal reconstruction, such as least squares or solutions that minimize a data fidelity term subject to a Tikhonov-like  $l_2$ -regularization to obtain the continuous space solutions. Numerically efficient methods for sampling and reconstruction of band-limited irregularly sampled data are based on iterative solutions for the original problem rewritten as the Toeplitz system with adaptive weights [77] or block-based solutions to variational methods approximated by B-splines [78].

**Mesh-free interpolation methods** Broadly utilized mesh-free strategies are mainly spline approximation techniques [79], [80], polynomial approximations [81], least-squares approximation techniques [82], [83], or approximations by shifts or radial basis functions [84], [85]. Mesh-free inter-

polation methods can be: (a) parametric and (b) non-parametric interpolation methods, depending if signals are modeled using the implicit interpolation function or if such function is approximated from data. Parametric methods use reconstruction filters to estimate interpolation values from the given local data set, determined by the filter size. Though this methods are very fast in regular grid setup, their performance is highly dependent on the filter parameters. In this work we use a nonparametric method for interpolation [86], where the values are estimated directly from a given data and we describe it below in more detail.

Several recent works deal with manifold data processing. For example, the nonparametric reconstruction method [87] assumes that an elastic membrane is attached to a set of unorganized set of data points. This approach is motivated with its dual in domain of electrostatics, where the membrane is enclosing a set of electric charges and it gets attracted or repelled by these charges in minimizing the electrostatic potential and the elastic energy. In [88] authors solve a variational problem on an implicit surface, represented as the zero level set of a higher dimensional embedding function. They show that the reconstruction accuracy is enhanced, although the dimensionality of the problem is increased. Recent work on local interpolation-related processing [89] gives models for different signal manifold patches and proposes a manifold regularization approach to solve an inverse problem. When a patch is a sparse structure, manifold-pursuit regularization method is solved by the iterative threshold methods with a sparsity prior.

Regression, and in particular linear regression modeling, is one of the most studied and widely used statistical tools. Initially, it is used for learning a manifold that represents the observed data in a compact form. In general, these algorithms first estimate the manifold parameters [90] and next, they perform regression. In [91], the estimate of the embedded tangent plane is computed by local principal component analysis. Then, it performs a local linear regression on the tangent plane. A nonparametric regression between two Riemannian manifolds is studied in [92] and similar kernel-based method is used to build a classifier on a Riemannian manifold in [93].

Distributed regression is proposed as the method suitable for modeling sensor network data in [94]. Since then, several methods, mainly based on linear regression, have been proposed in this context. A non-negative linear distributed regression for data inference in wireless sensor networks is proposed in [95]. Similar techniques are used for distributed function and time delay estimation in [96], [97], while the effects of quantization for nonparametric regression are studied in [98]. In the following, we provide more details on kernels in nonlinear regression.

We motivate the concept of data-adaptive weight functions in optimization problems by drawing the parallel between the characteristics of these weight functions in nonparametric regression methods [99] and Kernel functions defined in the field of statistical learning theory. Weight functions used in regression methods for signal estimation are non-negative and normalized. Such function characteristics in fact correspond to a probability distribution, which is used in the statistical learning theory for estimation of function values from its samples. If we draw in total  $q$  values uniformly and i.i.d. from the unknown distribution function  $p(\mathbf{m}, y)$ , where  $\mathbf{m}$  is the input vector and by  $y$  the output value, the goal is to find such a function  $\tilde{f}(\mathbf{m}_j)$  for which holds  $\tilde{f}(\mathbf{m}_j) = y_j$  for all  $j \in \{1, \dots, q\}$ . A usual approach to find such  $\tilde{f}$  is to minimize of the risk functional  $R(\tilde{f})$

$$R(\tilde{f}) = \int L(y, \tilde{f}(\mathbf{m}))p(\mathbf{m}, y)d\mathbf{m}dy, \quad (2.5)$$

where the value  $L(\cdot, \cdot)$  represents the prediction estimation error of the value  $\tilde{f}(\mathbf{m})$  instead of  $y$  [100]. A direct solution to solve the previous equation is to directly estimate the density  $p(\mathbf{m}, y)$ , which in most of the cases is not a feasible approach. Instead of solving the above risk function, the common practice is to estimate the empirical risk function from the data  $R_e(\tilde{f}) = \frac{1}{q} \sum_{j=1}^q L(y_j, \tilde{f}(\mathbf{m}_j))$ , for which it can be shown that if  $q \rightarrow \infty$ , the minimum of the function  $R_e(\tilde{f})$  tends to the minimum of  $R(\tilde{f})$ .

The Kernel methods are primarily used for parameter learning purposes and they replace the function  $\tilde{f}(\mathbf{m})$  by a weighted nonlinear transformation  $w_j \phi(\mathbf{m}_j)$ . Kernel methods are used to solve the previous equation by a general linear regressor method [101]. To avoid data overfitting, the Tikhonov regularizer is commonly added in a minimization problem to direct a tradeoff between the smoothness and the risk function minimization. If the function  $L(\cdot, \cdot)$  is convex (the Representer theorem holds), the optimal solution  $\mathbf{w}^*$  of Tikhonov regularizer is a linear combination of the observed measurements:  $\mathbf{w}^* = \sum_{j=1}^q \alpha_j \phi(\mathbf{m}_j)$ . The main advantage of using the Kernel function instead of the nonlinear transformation function  $\phi(\mathbf{m})$  is that the explicit function model of the nonlinear transformation function  $\phi(\mathbf{m})$  is not required.

Classical interpolation methods assume that the set of interpolation point positions is available prior to signal acquisition. Such assumption is not valid for interpolation performed by sensor network nodes and more flexible solutions are necessary. Interpolation problems have similarities with variational and statistical methods that estimate function values from available data. Discrete variational methods on graphs have very fast solutions, but they can not be applied to sensor network interpolation, since they require interpolation set position knowledge for graph building. To the best of our knowledge, other methods for estimation of the function values from data do not consider interpolation applications nor study the setups where sensors are distributed over a geometric manifolds. We rather provide interpolation method for networks that are flexible (arbitrary interpolation points), distributed (every sensor can perform interpolation of a value at its neighborhood) and that incorporate the geometry of the manifold data model.



## Chapter 3

# Distributed sensor failure detection in Sensor Networks

### 3.1 Introduction

Sensor networks are nowadays often built by inexpensive sensors. The unreliable data generated by sensors that are defective has unpredictable consequences on data interpretation tasks. To guarantee the stability and robustness of networks with dynamic architectures and loose coordination, it is very important to identify and eliminate the unreliable data generated by defective sensors. However, most of the previous detection methods are analyzed in centralized or semi-distributed settings, where one or more nodes have an additional functionality for handling failure detection. Contributed solutions however are not optimal in terms of robustness, nor coordination and communication costs. Hence, there is a clear need for novel distributed failure detection algorithms that are effective under network topology and communication constraints.

In this chapter, we address the problem of distributed sensors' failure detection in networks with a small number of defective sensors, whose measurements differ significantly from the neighbor measurements. We propose a novel distributed sensors' failure detection method that employs a simple distance decoder for sparse and binary signals. We consider that a sensor network is represented by a connected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where the vertices  $\mathcal{V} = \{s_i\}_{i=1}^S$  stand for the  $S$  sensors in the network and the edges  $\mathcal{E}$  determine sensors' connectivity. For instance, if two sensors  $s_i$  and  $s_j$  lie within each other's communication range, the edge  $e_{i,j} \in \mathcal{E}$  has a non-zero value. We assume that the function measured by sensors is smooth, so that neighbor sensors typically have similar measurements as long as sensors work correctly. We consider that at most  $K$  out of  $S$  sensors are defective, where  $K \ll S$  and that the defective sensors are scattered over the network. Therefore, the defective sensor identification problem boils down to a sparse binary signal recovery, where non-zero signal values correspond to defective sensors.

Our novel distributed detection approach builds on ideas used in Group Testing (GT) methods [30] that are commonly applied for centralized systems. The core idea is to perform low-cost experiments in the network, called *tests*, in order to detect the defective sensors. The tests (detection) are performed on pools of sensors by a set of sensors called *master* sensors. Each master sensor compares the sensor measurements based on a similarity measure (e.g., threshold-based similarity

measure) to detect the presence of defective sensors in its vicinity. The result of this test takes a binary value, which might be possibly altered by noise. The probabilistically designed tests and their outcomes together build *messages* at master nodes. They are communicated between the master nodes and their direct neighbors. Next, the messages are disseminated in the network with a gossip algorithm (rumor mongering) [17] that follows a pull protocol [14], where every sensor randomly picks a sender amongst its direct neighbors. Each time a new message reaches a sensor, its value is linearly combined with the current message at this sensor. This increases the diversity of information in the network, which is important for an accurate decoding. The message design and dissemination phases are repeated for several rounds. The specific probabilistic test design described later in this chapter ensures that the detection is successful with high probability, as long as the number of messages exceeds a given threshold (see Section 3.3.2). The decoder performs failure detection using a simple distance decoder method (e.g., Hamming decoder). Due to the distributed nature of our algorithm, the detection can be performed in any sensor and not only at a master node. We analyze the detection failure bounds and derive the conditions for successful failure detection in the case of a single defective sensor. Then, we provide the error bounds for detection of multiple defective sensors in the network. We show that the number of linearly independent messages required for detection is smaller in practice than the one given by the theoretical bounds obtained in our worst case analysis. We finally provide simulation results for regular and irregular networks. The experiments outline the advantages of the proposed detection method compared to other binary signal detection algorithms based on the random walk measurements gathering. Our algorithm outperforms random walk detection methods both in terms of detection accuracy and convergence rate, because it creates innovative messages with the higher rate and disseminates them faster.

The main contributions brought in this chapter can be briefly summarized as follows:

- We build on the sparse nature of the binary sensor failure signals to propose a novel distributed detection algorithm based on gossip mechanisms and on Group Testing (GT), where the latter has been used so far in centralized detection problems.
- The new distributed GT algorithm estimates the set of scattered defective sensors with a low complexity distance decoder from a small number of linearly independent binary messages exchanged by the sensors.
- We first consider networks with one defective sensor and determine the minimal number of linearly independent messages needed for its detection with high probability. We then extend our study to the multiple defective sensors detection by modifying appropriately the message exchange protocol and the decoding procedure. We show that, for small and medium sized networks, the number of messages required for successful detection is actually smaller than the theoretical value computed in a worst case analysis.
- Finally, simulations demonstrate that the proposed method outperforms methods based on random walks in terms of both detection performance and convergence rate.

The rest of this Chapter is organized as follows. Section 3.2 reviews the centralized Group Testing framework. Section 3.3 describes our novel distributed detection method. It presents

the message formation and dissemination processes in sensor networks and discusses the detection problem for single and multiple defective sensors. Section 3.4 presents the simulation results.

## 3.2 Centralized detection with probabilistic Group Testing

We first review the centralized detection of sensor failures with methods based on GT and then describe our novel distributed detection method. We adopt the following notation. Calligraphic letters denote sets,  $|\cdot|$  represents the number of elements in a set and the  $i$ -th column and row of the matrix  $\mathbf{W}$  are represented with  $\mathbf{W}_{:,i}$  and  $\mathbf{W}_{i,:}$ , respectively.

GT generally aims at detecting defective items in a set based on the outcome of binary tests. Non-zero entries of a  $S$ -dimensional binary vector  $\mathbf{f} \in \mathbb{F}_2^S$  indicate the defective sensors which is actually the unknown of our problem.  $\mathbb{F}_2$  is a finite field of size two and  $\mathbf{f}$  is a  $K$ -sparse signal, where  $K \ll S$ . The tests performed on sensor measurements are represented with a  $B \times S$  dimensional matrix  $\mathbf{W}$ , where  $B$  stands for the number of tests. The non-zero entries of  $\mathbf{W}_{i,:} \in \mathbb{F}_2^S$  refer to sensors that participate in the  $i$ -th test. The boolean matrix multiplication operator is denoted with  $\otimes$ . Then, the results of binary tests are denoted with the test outcome vector  $\mathbf{g} = \mathbf{W} \otimes \mathbf{f}$ , where  $\mathbf{g} \in \mathbb{F}_2^B$ .

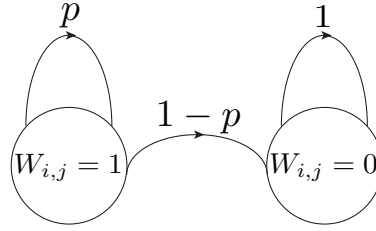
The design of the matrix  $\mathbf{W}$  is crucial for reducing the number of required tests for the detection of defective sensors. This design resembles the design of generator matrices of LDPC codes [65]. Motivated by this similarity, our test matrix  $\mathbf{W}$  is constructed as [102]:

$$W_{i,j} = \begin{cases} 1, & \text{with probability } q, \\ 0, & \text{otherwise.} \end{cases} \quad (3.1)$$

Hence, a sensor participates in a test with a probability  $q$ . Such a test matrix design assures that the matrix is *disjunct* with high probability ([102], Sec. IV, Def. 1). In other words, a matrix  $\mathbf{W}$  is called  $K$ -disjunct if no column  $\mathbf{W}_{:,i}$  of  $\mathbf{W}$  lies in the sub-space formed by any set of  $K$  columns  $\mathbf{W}_{:,j}$  with  $j \neq i$ . This property enables detection ([102], Sec. IV, Proposition 2) with a distance decoder (i.e., Hamming distance). The disjunct matrix parameter  $\epsilon$  represents the distance decoder threshold for detection. We define the support of the vector with the operator  $supp(\cdot)$ . For any column  $\mathbf{W}_{:,i}$  of the test matrix  $\mathbf{W}$  that is  $(K, \epsilon)$ -disjunct (Appendix A.4, Def. 1), the decoder verifies that

$$|supp(\mathbf{W}_{:,i}) \setminus supp(\mathbf{g})| \leq \epsilon, \quad (3.2)$$

where the operator  $\setminus$  is the set difference operator. In other words, the decoder counts the number of positions in the column  $\mathbf{W}_{:,i}$  that are different from the outcome vector  $\mathbf{g}$ . Then, the elements of  $\mathbf{f}$  are inferred as non-zero (marked as defective) iff the inequality (3.2) holds. In [102] (Theorem 4), the required number of measurements for successful decoding is equal to  $B = \mathcal{O}(K \log(S)/p^3)$  for centralized detection for any single set of defective sensors. In general, the noise model alters the non-zero entries in the test matrix  $\mathbf{W}$  with probability  $1 - p$ , as represented in Fig. 3.1. This noise model considers only this particular case. In the next section, we focus on distributed detection algorithms to provide robust solutions that have small costs.



**Figure 3.1:** Noise effect on binary symbols in the test message: this model assumes that only the non-zero values in the test matrix are flipped with probability  $(1 - p)$  to the value zero.

### 3.3 Proposed Distributed Detection

In this section, we propose a novel distributed failure detection algorithm and analyze its performance. In our framework, we assume that the signal under observation is smooth and that the measurements of locally distributed sensors are not significantly different when the sensors are not defective.

#### 3.3.1 Sensor network message design and dissemination

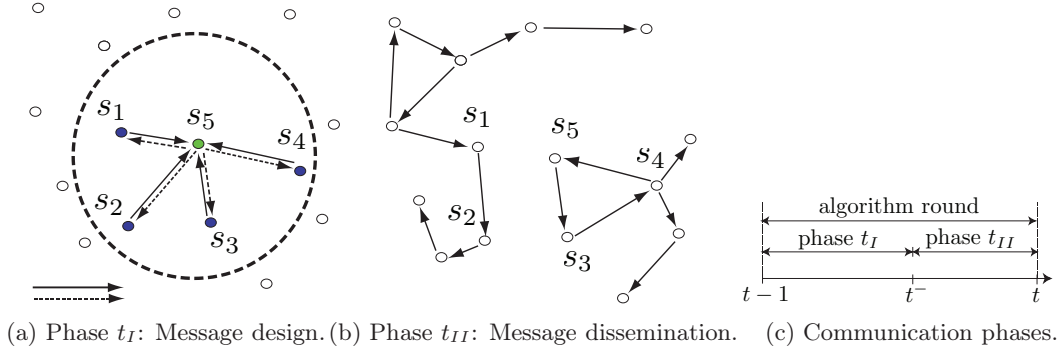
The algorithm is based on a novel test design and message dissemination strategy in a distributed GT framework. The sensors iteratively create and disseminate messages in two phases, denoted by  $t_I$  and  $t_{II}$ . One round of our iterative distributed detection algorithm consists of these two phases. During the first phase  $t_I$ , the sensors obtain *messages* that estimate the presence of defective sensors in their neighborhood. In the second phase  $t_{II}$ , the sensors linearly combine messages and exchange them employing a gossip mechanism. They are illustrated in Fig. 3.2 and described below in more detail.

#### Message construction phase

The first phase  $t_I$  in round  $t$  represents the message construction process illustrated in Fig. 3.2 (a). It proceeds as follows:

- Clustering of the sensors and measurement gathering  
 In the first stage,  $L$  master sensors are chosen either randomly or deterministically. The master sensors broadcast the message transmission request to all their neighbors. Sensors may receive multiple request messages from several master nodes, however, they can respond only to one request. The sensors forward their measurements to the master sensor. After this stage, the sensors form  $L$  disjoint subsets  $\mathcal{V}_l \subset \mathcal{V}$ ,  $l = 1, \dots, L$  and the master sensor collects the sensor measurements from the corresponding cluster.
- Test message design  
 In this stage, the master sensor designs the test message. Each sensor in the cluster randomly participates in the test with probability  $q$ , as given in Eq. (3.1). The master sensor compares the measurements of the sensors. It estimates the presence of the defective sensors (e.g., by





**Figure 3.2:** Illustration of the message design and dissemination through the sensor network. (a) Message formation based on local sensor measurements: Full and dashed arrows correspond to the steps of the message design, respectively. In the first step, the master sensor collects the sensor measurements from its neighbor sensors  $\{s_1, \dots, s_4\}$  and forms the message  $(g_l(t^-), \mathbf{W}_{l,:}(t^-))$ . In the second step, the message is propagated from the master sensor to its neighbor sensors. (b) Message dissemination based on a gossip algorithm with pull protocol, where the sensors request the messages from their neighbors chosen uniformly at random. (c) Rounds of communication in our iterative detection algorithm consist of the message design ( $t_I$ ) and the message dissemination ( $t_{II}$ ) phases.

thresholding) and attributes a binary value  $f(s_k) \in \mathbf{f}$  to each sensor in the neighborhood that participated in the test. The value  $f(s_k) = 1$  denotes that the sensor  $s_k$  is defective and  $\mathcal{K}$  marks the set of defective sensors. The influence of noise in the test is given as in Fig. 3.1 following a model used in centralized GT algorithm [102]. The test outcome at master node  $l$  is finally computed as:

$$g_l = \mathbf{W}_{l,:} \otimes \mathbf{f} = \begin{cases} 1, & s_k \in \mathcal{K}, \\ 0, & \text{otherwise,} \end{cases} \quad (3.3)$$

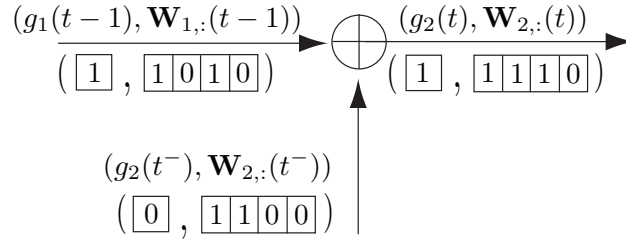
where the binary matrix operator  $\otimes$  is composed by the bitwise OR  $\odot$  and the bitwise addition  $\oplus$  operators. The matrix  $\mathbf{W} = [\mathbf{W}_{1,:}(t); \dots; \mathbf{W}_{B,:}(t)]$  (Matab notation) that marks sensors participating in tests is of size  $B \times S$ . The message formed by a master sensor  $l$  during the phase  $t_I$  consists of the outcome  $g_l$  and the test participation identifier  $\mathbf{W}_{l,:}$ .

- Broadcast of the designed test messages by the master sensors  
The master nodes sends the message  $(g_l(t^-), \mathbf{W}_{l,:}(t^-))$  at time  $t^- \in [t-1, t]$  to all the neighbor sensors, which concludes the phase  $t_I$ .

### Message dissemination phase

During the phase  $t_{II}$  the messages are disseminated within the network as illustrated in Fig. 3.2(b). This phase proceeds as follows:

- Message transmission request  
Every sensor  $s_i \in \{1, \dots, S\}$  requests the message obtained in the previous round from one of its neighbors chosen uniformly at random, following a gossip mechanism with pull protocol. We denote the chosen sensor is  $s_j$ .



**Figure 3.3:** The message formation at sensor  $s_2$  in round  $t$ . We assume that  $s_2$  pulls the sensor  $s_1$  to send its message, created in the previous round (round  $(t-1)$ ). We assume that the sensor  $s_3$  is defective  $\mathbf{f} = [0010\dots]$ . The outcome value and the test identifier vector are formed by bitwise OR operator.

- Sensor response  
The sensor  $s_j$  responds to the request by sending a message created in the previous round. This process is performed only once per round.
- Message computations  
The sensor  $s_i$  further combines these messages as:

$$\begin{aligned}
 g_i(t) &\leftarrow g_i(t^-) \oplus g_j(t-1), \\
 \mathbf{W}_{i,:}(t) &\leftarrow \mathbf{W}_{i,:}(t^-) \oplus \mathbf{W}_{j,:}(t-1),
 \end{aligned} \tag{3.4}$$

where  $g_j(t-1)$  denotes the sensor outcome value of the neighbor  $s_j$  at the previous round  $(t-1)$ . The vector  $\mathbf{W}_{i,:}(t)$  represents the test indicator vector at the sensor  $s_i$  in round  $t$ . Since the messages are created probabilistically, the message combination in the different rounds assures that an innovative message reaches sensors at every round with high probability. A toy example of the dissemination phases is illustrated in Fig. 3.3. In this example the sensor  $s_2$  at round  $t$  pulls the message from the sensor  $s_1$  and constructs a new message according to Eq. (3.4).

In a matrix form, the process of message formation and transmission in  $B$  rounds at any sensor in the network is represented as:

$$\mathbf{g} = \mathbf{W} \otimes \mathbf{f}. \tag{3.5}$$

This equation resembles to the outcome of computations in the centralized GT case. However, in distributed GT, the tests represent linear combinations of test vectors that build disjunct matrix with high probability, as given in Eq. (3.1).

We provide here an example to make a clear distinction between test matrices in distributed and centralized setups. We assume that an oracle has a direct access to the readings of the master nodes. A vector representation of tests performed at master nodes observed by an oracle at the round  $t = i$  is denoted by  $\mathbf{C}_{i,:}$ . Then, the matrix  $\mathbf{C} = [\mathbf{C}_{1,:}; \mathbf{C}_{2,:}; \dots \mathbf{C}_{B,:}]$  (Matlab notation) represents the test matrix over  $B$  collection rounds. Observe that the matrix  $\mathbf{C}$  is by construction disjunct (Appendix A.4, Definition 1), while  $\mathbf{W}$  is built on the boolean addition of rows of  $\mathbf{C}$  as in Eq. (3.4). The values in  $\mathbf{W}$  thus depend on the random message propagation path, which is obviously not the case in the centralized GT algorithm. Note that, for an arbitrary network, the number of network

rounds required for collecting a particular number of linearly independent tests varies and depends on the network topology, the number of master nodes  $L$  and the test participation probability  $q$ . Once every sensor has gathered test messages, it independently computes the binary vector  $\mathbf{f}$  that satisfies the tests in Eq. (3.5). This process is analyzed in more detail below.

### 3.3.2 Detection of one defective sensor in the network

We first analyze the case of a single defective sensor (case  $K = 1$ ) in the network and study the detection probability of our distributed algorithm. To recall, the distance decoder used for detection computes the Hamming distance between vectors. The element-wise distance is equal to one if element values differ, and to zero otherwise. To avoid false alarms, the decoder threshold  $\epsilon$  is set to a value higher than the expected number of noise-induced bit flips per columns in the disjunct matrix  $\mathbf{C}$  [102]:

$$\epsilon = (1 + \delta)(1 - p)qB. \quad (3.6)$$

where  $\delta > 0$  is a small constant and  $B$  is the number of rows in  $\mathbf{C}$ . The columns of  $\mathbf{C}$  have in average  $qB$  non-zero elements and non-zero elements are flipped with probability  $(1 - p)$ , so the expected number of flips per column is:

$$\mu = (1 - p)qB. \quad (3.7)$$

Recall that the matrix  $\mathbf{C}$  is by construction a disjunct matrix. The detection problem is resolved for tests that form a disjunct test matrix ([102], Proposition 2). However, the messages available at sensors in the network form a test matrix that is obtained by linear combinations of disjunct matrix rows, and not directly by disjunct matrix rows. Nevertheless, we show below that the distance decoder detects a defective sensor with high probability in this case under certain conditions.

The formal propositions for detection with high probability are given below. The Proposition 1 and the first part of Proposition 2 that refer to designing a  $(K, \epsilon)$ -disjunct matrix with high probability are similar to the Proposition 2 in [102], which is however derived for centralized detection. They represent the starting point of the analysis of our distributed algorithm. First we show that, for networks with a single master node the proposed distributed detection algorithm designs a  $(K, \epsilon)$ -disjunct matrix  $\mathbf{C}$  during the phase  $t_I$ . Next we show that linear combinations of rows of  $\mathbf{C}$  preserve distances between the test outcome and the column of the defective sensor in the test matrix. We then build on the first two propositions to analyze the number of messages needed for the distributed detection of a single defective sensor, which is given in Proposition 5.

Below we show that networks with a single master node ( $L = 1$ ) and probabilistic message design in  $t_I$  build a  $(K, \epsilon)$ -disjunct matrix  $\mathbf{C}$  with high probability. This case boils down to the centralized collection of data given in Proposition 2 in [102].

**Proposition 1.** *For a single-cluster network, the message design over the phase  $t_I$  of our proposed method builds a  $(K, \epsilon)$ -disjunct matrix  $\mathbf{C}$  with high probability for an arbitrary  $K$  and for  $\epsilon$  defined as  $\epsilon = (1 + \delta)(1 - p)qB$ , where  $\delta > 0$  is a small constant,  $B$  is the number of rows in  $\mathbf{C}$ ,  $q$  is a probability to have a nonzero entry in  $\mathbf{C}$ , where non-zero elements are flipped with probability  $(1 - p)$ .*

**Proof:** We denote by  $G$  the number of rows with a good disjunctness property. We show that

the probability that rows in  $\mathbf{C}$  have disjunctness property is smaller than  $\epsilon$  and we follow the development proposed in [102]. A row of the matrix  $\mathbf{C}$  given by  $\mathbf{C}_{i,:}$  is considered to have a good disjunctness property if a single symbol “1” occurs, while the rest  $K-1$  values are equal to zero. The probability of such an event is equal to  $\mu_1 = q(1-q)^{K-1}$ , where  $q$  is the probability to participate in a test (see Eq. (3.1)). Then the random variable that marks the total number of rows with such a property is denoted with  $G$ . The distribution of  $G$  is binomial with a mean value  $\mu_2 = \mu_1 B$ . We show that the probability of having less than  $\epsilon$  rows with a good disjunctness property is small under the assumption that  $\epsilon < \mu_2$ . We limit this probability by a Chernoff bound:

$$P(G < \epsilon) \leq e^{-\frac{1}{2} \frac{(\mu_2 - \epsilon)^2}{\mu_2}} = e^{-qB \frac{[(1-q)^{K-1} - (1-p)(1+\delta)]^2}{2(1-q)^{K-1}}}. \quad (3.8)$$

We mark the exponential term as  $\gamma = \frac{[(1-q)^{K-1} - (1-p)(1+\delta)]^2}{2(1-q)^{K-1}}$ . Since  $2 < e < 3$ ,  $2^{-m} \geq e^{-m} \geq 3^{-m}$  and constant  $m \geq 0$ , it follows that  $\gamma$  is bounded. For the parameter choice in [102]  $(\delta, m) = (\frac{p}{2}, \frac{p}{8})$ , its value becomes  $\gamma = \mathcal{O}(p^3)$ . Therefore this probability can be designed to be arbitrary small:  $P(G < \epsilon) \leq e^{-B\gamma/K} = e^{-\mathcal{O}(Bp^3/K)}$ .  $\square$

Then we show that linear combinations of rows of  $\mathbf{C}$  for networks with one master node preserve the Hamming distance only between the column of matrix  $\mathbf{W}_{:,k}$  that corresponds to the defective sensor  $s_k$  and the outcome vector  $\mathbf{g}$ .

**Proposition 2.** *Let  $\mathbf{C}$  be the  $(K, \epsilon)$ -disjunct matrix created over  $B$  consecutive rounds in a single-cluster network during the phase  $t_I$ . Linear combinations of messages generated during the phase  $t_{II}$ , performed as in Eq. (3.4), preserve the Hamming distance between the column of the resulting matrix  $\mathbf{W}_{:,k}$  that corresponds to the defective sensor  $s_k$  and the outcome vector  $\mathbf{g}$ .*

**Proof:** We first analyze the number of value flips that lead to a decoding failure for  $(K, \epsilon)$ -disjunct matrices, following a development similar to [102]. Next, we prove that linear combinations of rows in such matrices preserve vector distances between the outcome vector and the column of  $\mathbf{W}$  that corresponds to the defective sensor.

A decoding failure with a distance decoder occurs in a  $(K, \epsilon)$ -disjunct matrix when the number of flips of column elements of  $\mathbf{C}$  is higher than  $\epsilon$ . The probability of an occurrence of a single flip is equal to  $\mu_3 = q(1-p)$ . Let  $F$  denotes the number of flips in the columns of the matrix. Hence, the expected number of flips per column is given in Eq. (3.7). We want to compute the lower bounds for the event that more than  $(1+\delta)\mu$  flips occurred in the column of the matrix, for  $\delta > 0$ . Applying the Markov inequality  $P(F \geq (1+\delta)\mu) \leq \inf_{d>0} \frac{\prod_{i=1}^S E[e^{dF_i}]}{e^{d(1+\delta)\mu}}$  for a constant  $d > 0$  and plugging the probability of the single flip event

$$P(F_i) = \begin{cases} 1, & \text{with probability } (1-p)q, \\ 0, & \text{with probability } 1 - (1-p)q, \end{cases} \quad (3.9)$$

into the term  $E[e^{dF}i]$ , we write the Markov inequality as

$$P(F \geq (1+\delta)\mu) \leq \inf_{d>0} \frac{\prod_{i=1}^S [(1-p)q(e^d - 1) + 1]}{e^{d(1+\delta)\mu}}. \quad (3.10)$$

Using simple computations and setting  $x = (1 - p)q(e^d - 1)$  and  $1 + x < e^x$  we obtain:

$$\begin{aligned} P(F \geq (1 + \delta)\mu) &\leq \inf_{d>0} \frac{\prod_{i=1}^{m_i} e^{(1-p)q(e^d-1)}}{e^{d(1+\delta)\mu}} \\ &= \inf_{t>0} \frac{e^{(1-p)qm_i(e^d-1)}}{e^{d(1+\delta)\mu}} = \inf_{t>0} \frac{e^{\mu(e^d-1)}}{e^{d(1+\delta)\mu}}. \end{aligned} \quad (3.11)$$

We plug the constant  $d = \log(1 + \delta)$  to have

$$P(F \geq (1 + \delta)\mu) \leq \left(\frac{e^\delta}{(1 + \delta)^{(1+\delta)}}\right)^\mu = e^{\mu\delta - \mu(1+\delta)\log(1+\delta)}. \quad (3.12)$$

Observing that  $\log(1 + \delta) > \frac{2\delta}{2+\delta}$ , we finally obtain:

$$P(F \geq (1 + \delta)\mu) \leq e^{\frac{-\mu\delta^2}{2+\delta}}. \quad (3.13)$$

The outcome value  $\mathbf{g}$  depends on the presence of a defective sensor  $s_k$  in the test. We prove next that the distance between  $\mathbf{g}$  and the  $k$ -th column  $\mathbf{W}_{:,k}$  does not increase more than  $\epsilon$  during  $t_{II}$ , while this is not true for the rest of the columns. When the sensor  $s_j$  sends its message to the sensor  $s_i$  during the round  $t$ , we have:

$$\begin{aligned} \text{dist}\left(g_i(t), W_{i,k}(t)\right) &= \text{dist}\left(g_i(t^-) \oplus g_j(t-1), W_{i,k}(t^-) \oplus W_{j,k}(t-1)\right) \\ &= \text{dist}\left(g_i(t^-), W_{i,k}(t^-)\right) \oplus \text{dist}\left(g_j(t-1), W_{j,k}(t-1)\right), \end{aligned} \quad (3.14)$$

where the first equality results from Eq. (3.4). The second equality follows directly from the fact that the values of  $\mathbf{g}(t^-)$  and the columns  $\mathbf{W}_{:,k}(t^-)$  are identical for the defective sensor due to Eq. (3.5). Since these two columns initially may differ at  $\epsilon$  positions due to noise flips, the overall distance between the vectors  $\mathbf{g}(t^-)$  and  $\mathbf{W}_{:,k}(t^-)$  is at maximum  $\epsilon$ , which is defined by Eq. (3.6).  $\square$

We now consider networks with  $L$  master sensors and an hypothetical centralized data collection. We assume that  $L$  master nodes cluster the sensor network in disjoint subsets, where every sensor belongs to exactly one cluster. The master nodes perform message design over the rounds  $t_I$  as proposed by our algorithm. We show that the tests gathered from the  $L$  different clusters build a disjunct matrix, where each cluster relates to a  $(K, \epsilon)$ -disjunct matrix.

**Proposition 3.** *The diagonal matrix  $\mathbf{C} = \text{diag}(\mathbf{C}_1, \dots, \mathbf{C}_L)$  obtained from  $(K, \epsilon)$ -disjunct matrices  $\mathcal{C} = \{\mathbf{C}_i\}_{i=1}^L$  is at least  $(K, \epsilon)$ -disjunct.*

**Proof:** The proof follows directly from the disjunctness property of matrices in  $\mathcal{C}$  (Propositions 1 and 2).  $\square$

We analyze now the influence of the message gathering process over successive rounds of our detection algorithm. Uniform gathering of linearly combined messages at  $L$  clusters by a hypothetical centralized decoder enables the detection of the defective sensor with high probability when the number of received messages is sufficient.

**Proposition 4.** When the  $(K, \epsilon_i)$ -disjunct matrices  $\mathcal{C} = \{\mathbf{C}_i\}_{i=1}^L$  are linearly combined as in Eq. (3.4), where  $\epsilon = \sum_{i=1}^L \epsilon_i$  and  $q = \sum_{i=1}^L q_i$ , the resulting test matrix permits the detection by a distance decoder with high probability as long as  $B \geq \mathcal{O}(K \log(S)/p^3)$  messages are collected from randomly chosen clusters to build this matrix.

**Proof:** We first show that a diagonal matrix constructed from  $(K, \epsilon_i)$ -disjunct matrices of the set  $\mathcal{C}$  is  $(K, \epsilon)$ -disjunct. Let the number of rows for all matrices be  $B = \mathcal{O}(K \log(S)/p^3)$ . The parameters  $\epsilon$  and  $\epsilon_i$  are defined in Eq. (3.6) and  $\epsilon = \sum_{i=1}^L \epsilon_i = (1 + \delta)(1 - p)Bq$ , so the diagonal matrix built out of  $(K, \epsilon_i)$  matrices is  $(K, \epsilon)$  disjunct. The next part of the proof follows from Proposition 2, which states that a matrix whose rows are formed by linear combinations of rows of  $(K, \epsilon)$ -disjunct matrices, guarantees a high probability detection with a distance decoder. Finally, we prove that for a given matrix  $\mathbf{C}$  the disjunct property holds if at least  $B$  messages are available. For this purpose, we follow a development similar to [102] and consider that the number of sensors in the clusters is equal to the total number of sensors  $n = S$ . The probability bound given in Proposition 1 should hold for all possible choices of a fixed set of  $T$  out of  $S$  columns:  $\cup_T P(G \leq \epsilon) \leq S e^{-Bq\gamma}$ . This probability can be arbitrary small, e.g., in case  $B \geq \frac{K \log S}{m\gamma} = \mathcal{O}(K \log S/p^3)$ . Further on, the condition in Eq. (3.13), which gives the probability bound that the number of flips in any  $K$  out of  $T$  columns exceeds a threshold value  $\epsilon$  is also bounded. It reads:

$$\cup_K P(F \geq (1 + \delta)\mu) \leq K e^{\frac{-\delta^2}{2+\delta}\mu} = K e^{\frac{-\delta^2}{(2+\delta)p^3}(1-p)qK \log(S)},$$

where the last equality is obtained from Eq. (3.7). This probability is small for a sufficiently large value of  $B = \mathcal{O}(K \log(S)/p^3)$ .  $\square$

We now analyze the proposed distributed algorithm and consider the detection requirements for every sensor in the network. We show that the test messages collected by the sensors during the transmission rounds enable failure detection by the distance decoder with high probability if the number of messages is sufficient, where the decoder operations are performed locally at sensors.

**Proposition 5.** We assume that  $L$  master sensors partition the sensor network in disjunct parts. Test realizations within a cluster form test vectors. Over the rounds, these vectors create  $(K, \epsilon)$ -disjunct matrices  $\mathcal{C} = \{\mathbf{C}_i\}_{i=1}^L$  whose elements:

$$c_{i,j} = \begin{cases} 1, & \text{with probability } q_i = \alpha_i, \\ 0, & \text{otherwise,} \end{cases} \quad (3.15)$$

where  $q = \sum_{i=1}^L q_i$ . The messages  $(\mathbf{g}_i, \mathbf{W}_{i,:})$  arrive at all the sensors in the network in our proposed algorithm, as described in the previous section. If the above assumptions hold and if the number of linearly independent messages received from each cluster at every sensor in the network is at least  $B/L$ , where  $B \geq \mathcal{O}(K \log(S)/p^3)$ , the probability that sensors fail to detect the defective sensor by the distance decoder tends to zero as  $S \rightarrow \infty$ .

**Proof:** The message collection method does not influence the decoder performance, since the number of per-cluster measurements is sufficient for decoding with high probability. Therefore, the proof follows from the proof of Proposition 4.  $\square$

### 3.3.3 Detection of multiple defective sensors in the network

We analyze now the distributed detection of multiple defective sensors where  $K \ll S$  holds. We propose here to slightly modify our distributed algorithm and to limit the decoder search space to be able to apply the Hamming distance decoder. The protocol modification and the adaptation of the distance decoder are described below. We assume that sensors have a knowledge about sensors belonging to each cluster and that at most one defective sensor is located in a given cluster. This knowledge limits the size of the decoder search space.

The proposed protocol is first modified to deal with multiple defective sensors. A decoder error occurs when two or more messages with positive test outcomes are combined together during the phase  $t_{II}$ , since the distance preserving property defined in Eq. (3.14) is not guaranteed in this case. Since the number of defective sensors is very small compared to the total number of sensors, this event however occurs rarely. We explain the protocol modification by a simple example. Let the sensor  $s_i$  pull a message from the sensor  $s_j$ , where both sensor test outcomes have non-zero values. Instead of combining the messages as in Eq. (3.4), we simply buffer the new message of  $s_i$  and consider the message from  $s_j$  at previous round as the final outcome of the phase  $t$ :

$$\begin{aligned} g_i(t) &= g_j(t-1), \\ \mathbf{W}_{i,:}(t) &= \mathbf{W}_{j,:}(t-1). \end{aligned} \quad (3.16)$$

At the first subsequent round  $\tau \geq t+1$  of our distributed algorithm where both messages  $g_i(\tau)$  and  $g_j(\tau-1)$  have non-zero values as test outcomes,  $g_i(\tau)$  is replaced by the message buffered in the node  $s_i$ . The rest of the protocol remains unchanged.

The decoding proceeds in two main steps. First, the appropriate unions of test matrix columns are created to form the search set space and second, the Hamming distance between the test outcome vector and the vectors of the search set are computed. The minimum Hamming distance indicates the solution of the detection problem. The outcomes  $\mathbf{g} = [\mathbf{g}_0^T \ \mathbf{g}_1^T]^T$  collected at some sensor are divided into two sets, i.e., the negative and positive outcome vectors  $\mathbf{g}_0$  and  $\mathbf{g}_1$ , respectively. Subsequently, the rows of the test matrix  $\mathbf{W}$  form two sub-matrices  $\mathbf{W}_0$  and  $\mathbf{W}_1$  so we rewrite Eq. (3.5) by

$$\begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{W}_0 & 0 \\ 0 & \mathbf{W}_1 \end{bmatrix} \begin{bmatrix} \mathbf{f}_0 \\ \mathbf{f}_1 \end{bmatrix}. \quad (3.17)$$

We eliminate non-defective sensors from  $\mathbf{W}_1$  using the knowledge from  $\mathbf{W}_0$  and obtain  $\mathbf{W}'_1$ . Recall that master nodes partition sensors into clusters. The columns of interest are those columns of  $\mathbf{W}'_1$ , which contain at least one non-zero value, since they mark the participation of the potential defective sensors in tests. We build the total search space  $\mathcal{U}$  by taking into account the total or partial knowledge of sensor cluster partition. When the full cluster partition is known, the columns of the sensors that belong to the same cluster  $l$  are grouped together in a set  $\mathcal{H}_l$ ,  $l = \{1, \dots, L\}$ , where  $L$  is the total number of clusters (master nodes). Given a knowledge about the sensor cluster partition for a subset of sensors, the cardinality of the set  $|\mathcal{H}| = |\{\mathcal{H}_l\}_{l=1}^{L^*}|$  is  $L^* > L$ . The search space  $\mathcal{U}$  consists of vectors that are obtained by element-wise OR addition of up to  $K$  vectors that are chosen from different sets  $\mathcal{H}_l$ , because of the assumption that at most one defective sensor exists in each cluster. For instance, let the number of defective sensors and clusters be  $(K, L) = (2, 2)$ . Let  $\mathcal{H}_1$  contain  $h_1$  and let  $\mathcal{H}_2$  contain  $h_2$  columns. Then the search space size

has in total  $|\mathcal{U}| = h_1 h_2 + h_1 + h_2$  elements, where  $h_1 h_2 = \binom{h_1}{1} \cdot \binom{h_2}{1}$  denotes the number of unions of  $K = 2$  columns and single column subsets are chosen in  $h_1 + h_2$  ways. Distance decoding is performed between  $\mathbf{g}_1$  and elements of the set  $\mathcal{U}$ , starting from the vectors that are created as unions of  $K$  columns towards the smaller number of column unions. If no solution exists for a particular value of  $K$ , we perform the decoding for vectors built from  $K - 1$  column unions of  $\mathcal{H}_l$ . If no unique solution is found, we encounter a decoding failure.

We now analyze the number of messages that are required for the detection of multiple defective sensors with high probability.

**Proposition 6.** *Under the assumption that at most one defective sensor occurs in each cluster, that the number of available linearly independent messages at all sensors is at least  $B/L$  per cluster with  $B \geq \mathcal{O}(K \log(S)/p^3)$  and that sensors know what sensors belong to each of the clusters in the network, the distance decoder detects defective sensors at all sensors in the network with high probability.*

**Proof:** To recall, the transmission protocol ensures that the assumptions imposed by Proposition 5 hold for one defective sensor. Then, due to the assumption that at most one defective sensor is present in one cluster and that there is at most one defective sensor active in the test, we form the set of solutions for the multiple defective case, which has a unique solution. The distance decoder between the outcome vector and a limited set of vectors that form a full search space can therefore find the appropriate solution. In other words, this procedure is identical to the case of per-cluster decoding, where each cluster has at most one defective element, so the Proposition 5 can be applied here.  $\square$

**Proposition 7.** *Under the assumption that at most one defective sensor is present in each cluster, that the number of available linearly independent messages at all sensors in the network is at least  $B/L$  per cluster, where  $B \geq \mathcal{O}(K \log(S)/p^3)$  and that sensors know the partial set of identifiers of the clusters in the network, the distance decoder detects defective sensors at all sensors in the network with high probability.*

**Proof:** The search space  $\mathcal{U}$  created in this case is larger, but it contains the solution. Here the proof is identical to that in the previous proposition.  $\square$

The assumption that at most one defective sensor occurrence per cluster is reasonable when  $K \ll S$ . It is possible to bound the probability that at least two defective sensors occur within any cluster with the particular combination of  $(S, L, K)$  parameters. We provide this analysis in the following for the sake of completeness.

### 3.3.4 Probability of at least two defective sensors in a cluster

We show now that our assumption that at most one defective sensor occurs per cluster is reasonable. We here bound the probability that at least two defective sensors occur within any cluster. An erroneous message is generated in a cluster that contains more than one defective sensor when only a fraction of defective sensors participate in the test actively and we denote the probability of such an event  $E$  with  $P(E)$ . If defective sensors participate in the test, the distance within the column



that relates to these vectors and the outcome result does not change. The same occurs if none of the defective sensors participate in a test. Due to the protocol modification described above, only one cluster may generate the erroneous message per round. In total we assume that there are  $m \in \{2, \dots, K\}$ ,  $m \leq n$  defective sensors and that clusters contain  $n = \frac{S}{L}$  sensors. Then, the probability of a decoding error in one cluster,  $P_{cl}(E)$ , is equal to:

$$P_{cl}(E) = \sum_{m=2}^K P(n, q|m)P(m) = \sum_{m=2}^K P(n|m)P(q|m)P(m), \quad (3.18)$$

due to independence of parameters  $n$  and  $q$ .  $P(m)$  represents the probability that some cluster contains  $m$  defective sensors,  $P(n|m) = \binom{n}{m}$  is a probability of choosing  $m$  defective sensors within a cluster with  $n$  sensors and  $P(q|m)$  denotes the conditional probability of the error occurrence in a cluster with  $m$  defective sensors and a test participation probability  $q$ . We assume<sup>1</sup> that  $m$  takes a value from the set  $\{2, \dots, K\}$  with uniform distribution, so  $P(m) = \frac{1}{K-1}$  (in total,  $K-1$  value). Next,  $P(q|m) = 1 - q^m - (1-q)^m$  (see Appendix A.1). The total error probability for  $L$  clusters is bounded by  $P(E) \leq L \cdot P_{cl}(E)$ , so:

$$P(E) \leq L \frac{1}{K-1} \sum_{m=2}^K \frac{1 - q^m - (1-q)^m}{\binom{n}{m}}. \quad (3.19)$$

We use the well known binomial coefficient inequality  $\binom{n}{m} \geq \left(\frac{n}{m}\right)^m$  that holds for  $n, m > 0$  where  $m < n$  and  $1 - q^m - (1-q)^m \leq 1$ ,  $q \in \{0, 1\}$  to bound the value:

$$\frac{1 - q^m - (1-q)^m}{\binom{n}{m}} \leq \frac{1 - q^m - (1-q)^m}{\left(\frac{n}{m}\right)^m} < \frac{1}{\left(\frac{n}{m}\right)^m}, \quad (3.20)$$

We rewrite  $\left(\frac{n}{m}\right)^m$  by using a well known inequality as  $\left(\frac{n}{m}\right)^m = \left(1 + \frac{n-m}{m}\right)^m \leq e^{n-m}$ . Plugging these expressions into the previous expression and performing simple calculations we finally obtain:

$$P(E) < \frac{L}{K-1} e^{2-n} \frac{e^{K-1} - 1}{e - 1}. \quad (3.21)$$

For example, for the network values  $(S, L, K) = (70, 5, 3)$  this probability is bounded with  $P(E) < 1.1 \cdot 10^{-4}$ .

The distance decoder error probability due to our assumption that only one defective sensor is present in the network is thus very small. In addition, the decoder threshold value can be updated to increase the robustness. We finally increase the value of threshold parameter as  $\epsilon' = \epsilon + \delta_\epsilon$ , where  $\delta_\epsilon = P(E)E(\mathbf{g}_1)$  and  $E(\mathbf{g}_1)$  is the expected number of non-zero test outcomes. It is set to the total number of observed positive test outcomes.

---

<sup>1</sup>There is more than one defective element, so the minimum value of  $m$  is two.

## 3.4 Experimental results and discussion

### 3.4.1 Simulation setup

In this section, we investigate the performance of our distributed detection method denoted as GP in various scenarios. We first examine the influence of different network parameters on the rate of message dissemination. Next, we examine the decoding probability for defective sensor(s) detection. The number of system rounds required to collect the necessary number of messages for accurate decoding varies with the network topology. The simulations are performed for fully connected,  $k$ -connected and irregular graphs. Finally, we discuss the number of required linearly independent measurements for successful detection and compare it with the theoretical value.

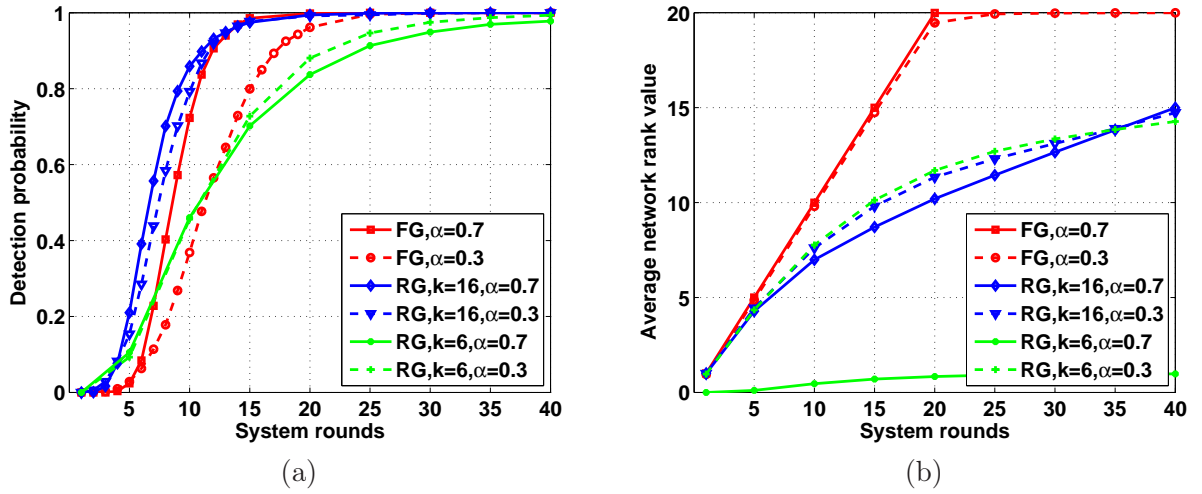
We also analyze the performance of several alternative schemes, namely a Random Walk method that employs a Gossip mechanism with pull protocol (RWGP) and a classical Random Walk (RW) detection. A random walk determines the path of successive random dissemination message exchanges between neighbor sensors. In the RWGP method, the random walk is initiated at  $L$  sensors (equivalent to the master sensors in the GP method) and terminates after a pre-determined number of rounds. The sensors create messages from the sensor measurements collected along the random walk path. These messages are transmitted with the gossip algorithm that uses a pull protocol. Note that, for identical choice of the sensors over rounds, RWGP and GP are identical. The RW method initiates the raw (uncompressed) measurements collection at  $L$  random sensors and completes it in a given number of rounds. Every sensor that lies along the random walk path stores the values of all sensors whose measurements are collected along the transmission path. When all the sensors receive all data, the process terminates.

The proposed GP algorithm is also compared with a Store-and-Forward (SF) and a Greedy Store-and-Forward (GSF) methods that employ a pull protocol. Both algorithms disseminate raw sensor measurements. For the SF method, upon receiving a message request, a node responds by forwarding randomly chosen messages from the available set of messages. In GSF, each sensor randomly requests the innovative measurements in a greedy manner from its randomly chosen neighbor sensors. This procedure involves an additional message exchange among sensors in every round.

We analyze the performance of these algorithms in fully connected,  $k$ -regular and irregular networks, respectively. To obtain irregular topologies, we randomly position sensors in a unit square area. Sensors that lie within a certain radius may exchange messages directly. In each case, we build ten different network realizations and for each such realization we perform 100 independent simulations. The results are averaged over all the simulations.

### 3.4.2 Detection performance

We first consider the case of a single defective sensor ( $K = 1$ ). The detection probability and the average rank evolution over rounds are examined for a fully connected (FG) and  $k$ -connected regular networks (RG) with sensors degree  $k \in \{6, 16\}$ . Here, the network consists of  $S = 20$  sensors. Fig. 3.4 shows that networks with higher number of connections faster disseminate the innovative messages. We also note that high connectivity value  $k$  is beneficial, but itself it cannot drive the performance of our detection scheme. It should be combined with the appropriate choice of network parameters, as discussed in more detail in Section 3.4.3. For example, random master



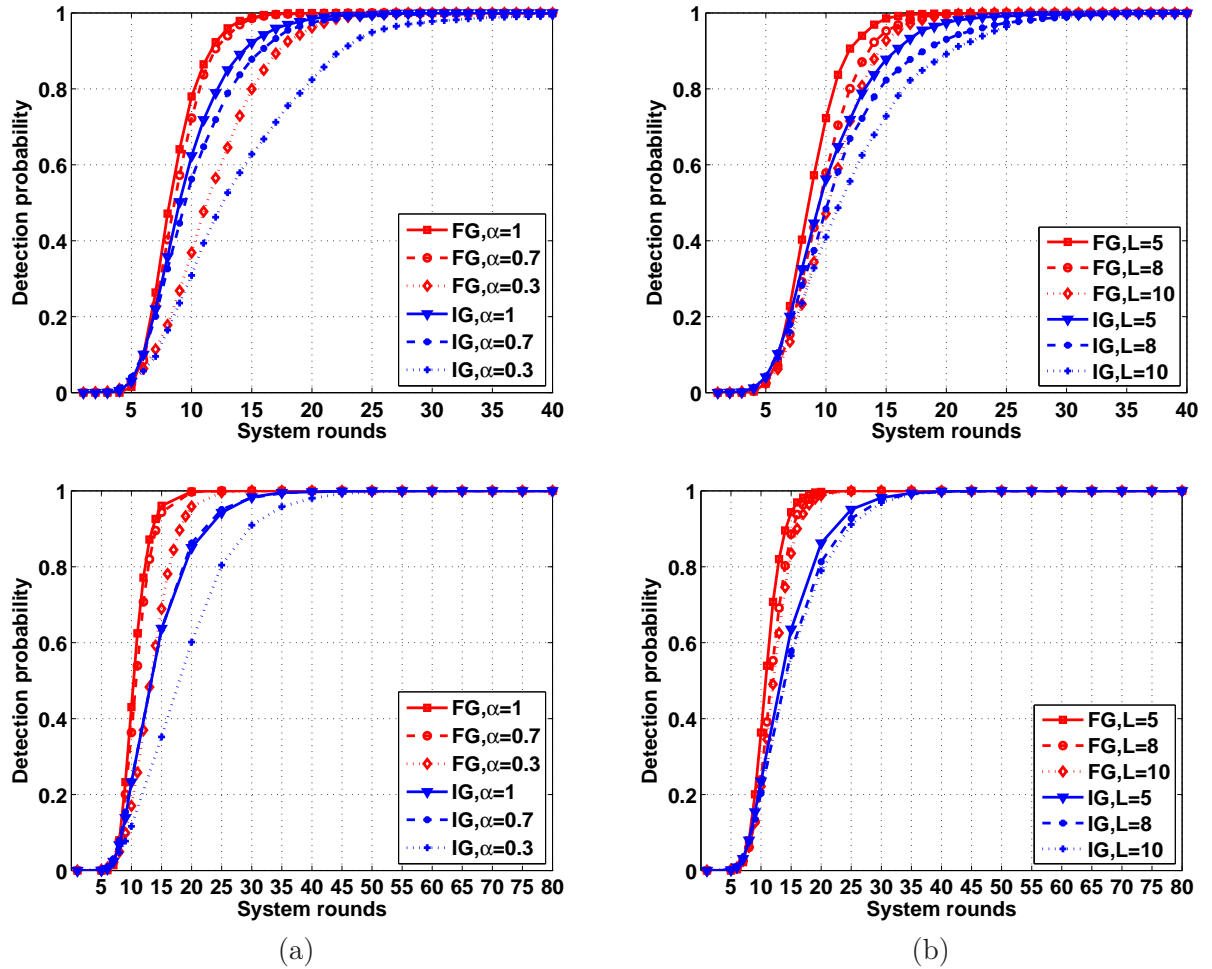
**Figure 3.4:** Simulation results for fully connected (FG),  $k = 16$ -regular connected (RG,  $k = 16$ ) and  $k = 6$ -connected graphs (RG,  $k = 6$ ) and sensor participation constant  $\alpha = qK$  with  $S = 20$  sensors,  $K = 1$  and a random selection (RM) of  $L = 5$  master sensors: (a) Probability of defective sensor detection; (b) Average rank of messages received per sensor.

(RM) sensor selection for  $k$ -connected networks ( $k = 16$ ) achieves better detection performance, compared to that of fully connected graphs.

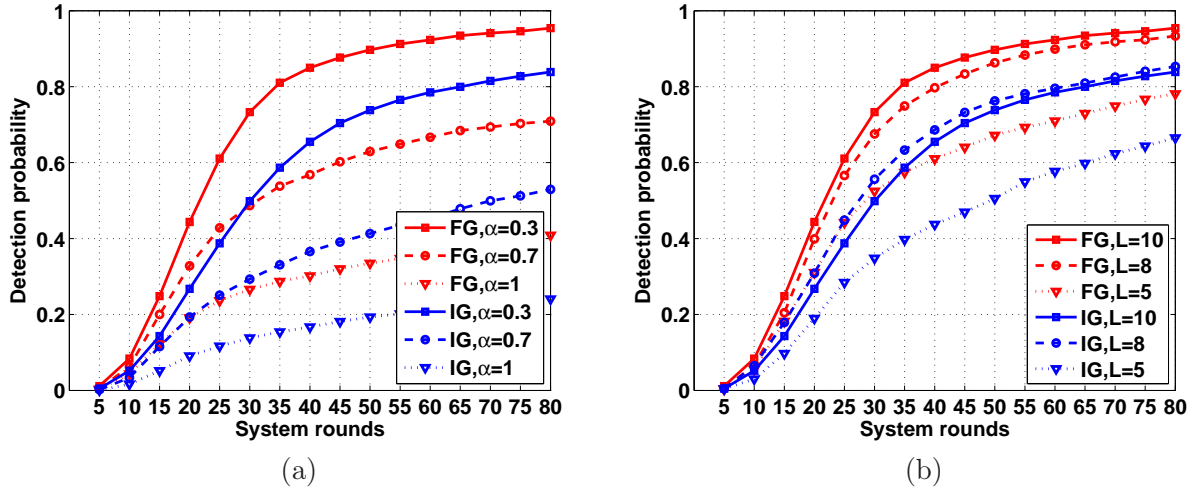
In Fig. 3.5, we illustrate the detection probability for random graphs with parameters  $(S, K) = (20, 1)$  in the top row and  $(S, K) = (70, 1)$  in the bottom row. The minimum sensors' degree is  $k \geq 3$ . We observe that random networks require more rounds in average for successful detection, as expected. Also, we observe that the detection performance decreases because of the limited message diversity (smaller probability of receiving innovative messages) and the lower connectivity.

We then consider the case of multiple defective sensors. In Fig. 3.6, we present results for the cases with  $K = 2$  defective sensors in networks with 70 sensors. The results are given in terms of the average detection probability over dissemination rounds, for both fully and irregularly connected graphs. The master sensors are selected deterministically (DM) due to the decoder design for multiple defective sensors identification. We focus on the evolution of the decoding probability and the average number of messages collected over rounds. From the evaluation it is clear that the detection performance is reasonable when the selected parameters values ( $L, \alpha$ ) favor diverse message generation.

In [102], a centralized system has been proposed, which can be considered similar to fully connected networks with centralized tests (single master sensor that covers all the network). For comparison reasons, we compute the required number of measurements for networks with the following settings:  $(S = 20, K \in \{1, 2\}, p \in (0.9 - 1), q \in (0.15 - 0.3), pf_1 = 0.01, pf_2 = 0.01)$  and  $(S = 70, K \in \{1, 2\}, p \in (0.9 - 1), q \in (0.15 - 0.3), pf_1 = 0.01, pf_2 = 0.01)$ . Here,  $pf_1$  and  $pf_2$  defined in [102] denote the probability that the bit flips (the non-zero message weight elements are flipped to value zero) occur in the test matrix and the probability that the matrix  $\mathbf{C}$  is not disjunct



**Figure 3.5:** Probability of defective sensor detection; Simulation results for irregular graphs ( $k > 3$ ) and random selection (RM) of (Top  $S = 20$ ) and (Bottom  $S = 70$ ) sensors,  $K = 1$ . (a)  $L = 5$  master sensors; (b) sensor participation constant  $\alpha = qK = 0.7$ .



**Figure 3.6:** Probability of defective sensor detection for fully connected (FG) and irregular graphs (IG),  $d > 3$  with  $S = 70$  sensors,  $K = 2$  and deterministic selection (DM) of master sensors. (a) fixed number of clusters  $L = 10$ ; (b) fixed sensor participation constant  $\alpha = qK = 0.3$ .

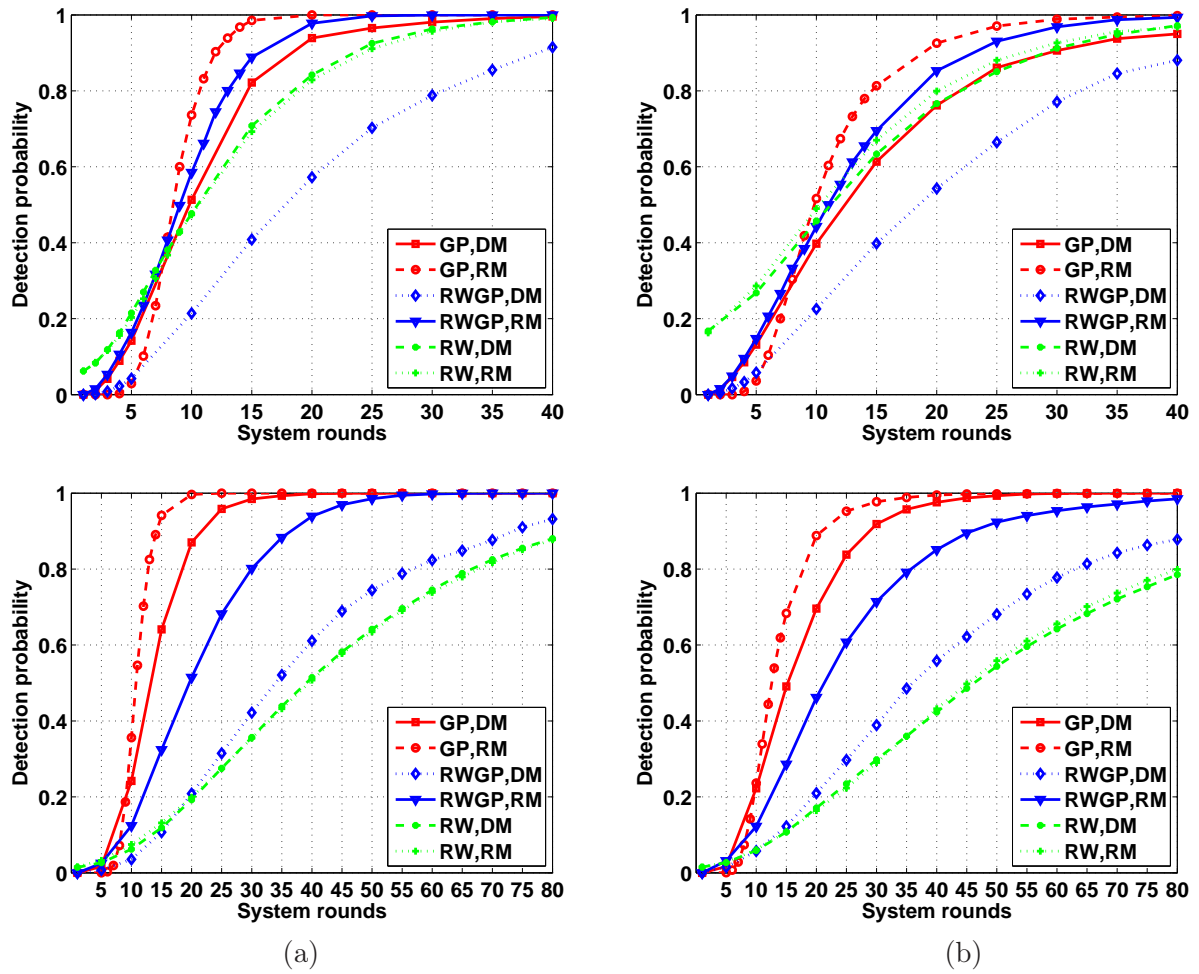
**Table 3.1:** Theoretical measurement rounds requirements for networks with  $S$  sensors.

	S=20		S=70	
	$K = 1$	$K = 2$	$K = 1$	$K = 2$
$p \in (0.9 - 1)$	130	(115-244)	(174-217)	(125-284)

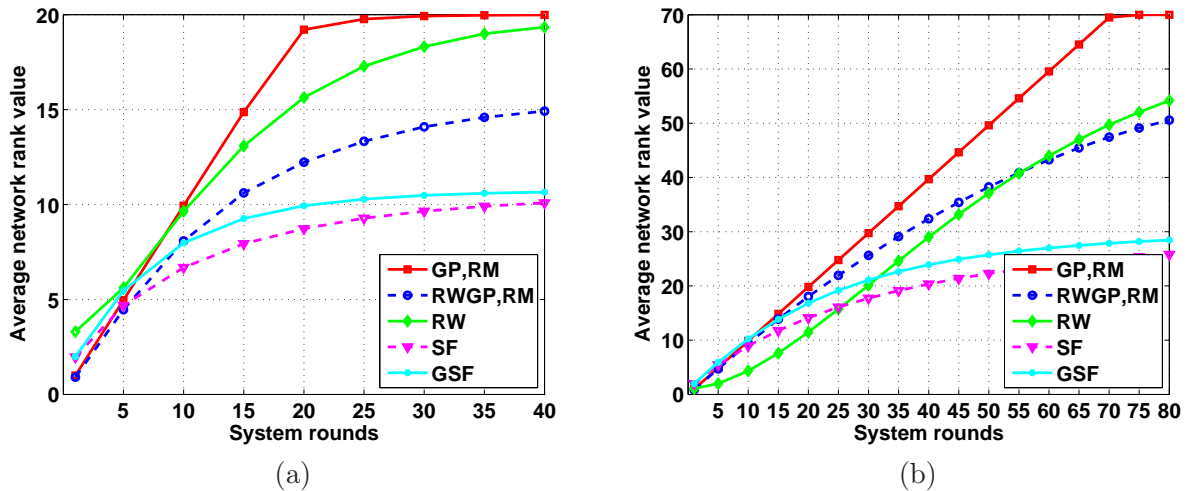
by construction (see Proposition 1). The results are reported in Table 3.1. We observe that the worst case analysis leads to a higher number of dissemination rounds than the real ones. However, these values decrease relatively to the growth of number of sensors in the network. Simulations show that in practice the number of required measurements is significantly smaller.

The detection probability of the proposed method and of several random walk detection methods is illustrated in Fig. 3.7 for  $S = 20$  sensors in the top row and  $S = 70$  sensors in the bottom row. The proposed scheme outperforms all other methods under comparison. Note that the number of required rounds in RWGP scheme for a high probability detection is large compared to the other schemes, while RW requires higher communication overhead for dissemination due to the transmission of raw (non-compressed) sensor measurements. Average rank values over the network rounds are illustrated in Fig. 3.8. We observe that for the fixed detection probability  $p = 0.9$  for the network with  $S = 70$  sensors the average number of system rounds required for the proposed method lies in [17, 20]. The number of system rounds required by the other algorithms to reach the same probability of performance is higher.

We note here that the class of defective sensor signals, which can be successfully detected is smaller in GP than in the centralized GT method because of the additional assumption that the defective sensors are scattered in the network. The proposed approach may be partially recover those sparse signals whose non-zero values are localized, due to the cluster prtitioning. However,



**Figure 3.7:** Detection performance for networks with (top  $S = 20$ ) and (bottom  $S = 70$ ) sensors and  $L = 5$  master sensors. Abbreviations: GP: Proposed method, RWGP: Random Walk gossip dissemination algorithm with pull protocol, RW: Random Walk in the network initiated at  $L$  sensors. (a) fully connected sensor network; (b) irregular sensor network.



**Figure 3.8:** Average rank value for irregular sensor networks with  $L = 5$  master sensors: (a)  $S = 20$  sensors (b)  $S = 70$  sensors. Abbreviations: GP: Proposed method, RWGP: Random Walk gossip dissemination algorithm with pull protocol, RW: Random Walk in the network initiated at  $L$  sensors, SF: pull store-and-forward algorithm with a random message transmission, GSF: pull store-and-forward algorithm with a greedy message transmission.

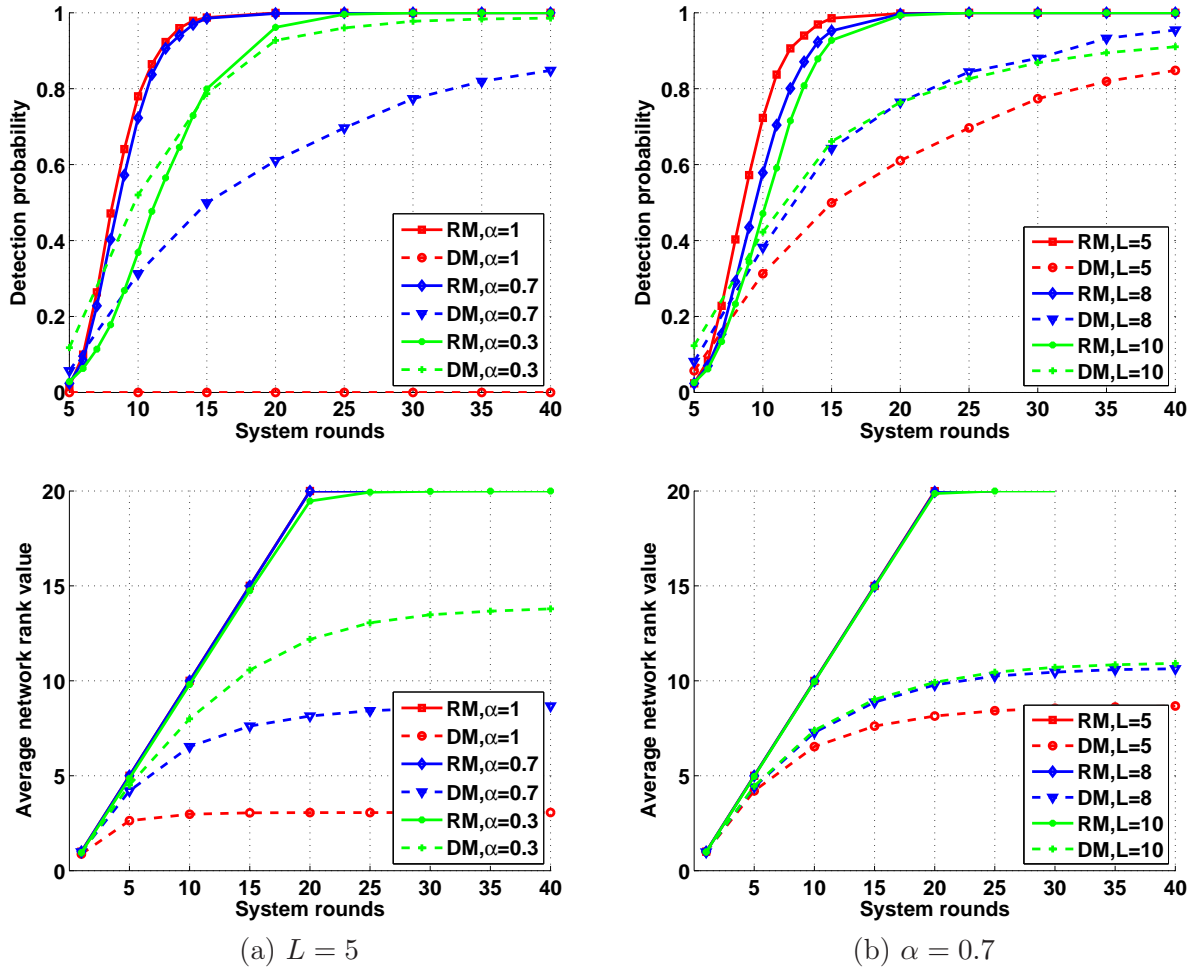
the cardinality of the class of locally distributed sparse signals is much smaller than the number of realizations of scattered sparse signals that this algorithm detects. Also, we highlight that the main advantages of the proposed distributed algorithm compared to the centralized one are reflected in lower communication costs and higher robustness to network topology changes. We finally note that any sensor can act as a decoder, which is useful in scenarios with one or more mobile decoders.

While we do not attempt to optimize the parameter values in this work, we briefly discuss the influence of the parameters below.

### 3.4.3 Influence of the master node selection process

We study the influence of networks' capability to generate innovative messages on the decoder performance. We consider two different methods for selecting master sensors: random master sensor selection (RM) and deterministic master sensor (DM) selection. Fig. 3.9 illustrates the detection probability and the achieved average rank with respect to the number of message dissemination rounds, for fully connected graphs with  $S = 20$  sensors and one ( $K = 1$ ) defective sensor. We observe that the performance depends on  $L$  and  $\alpha = qK$  for both RM and DM. These values should be selected properly in order to maximize the information diversity in the network. Specifically, we observe that RM achieves the maximum message diversity for  $\alpha = 1$  (maximum value) since the diversity of messages in this case is maximized by construction (see Fig. 3.9).

We can also note that the number of clusters does not affect significantly the detection performance of RM. On the contrary, for DM both parameters  $L$  and  $\alpha$  are important. Small values of  $\alpha$  guarantee high message diversity. This is due to the fact that DM requires more rounds to receive

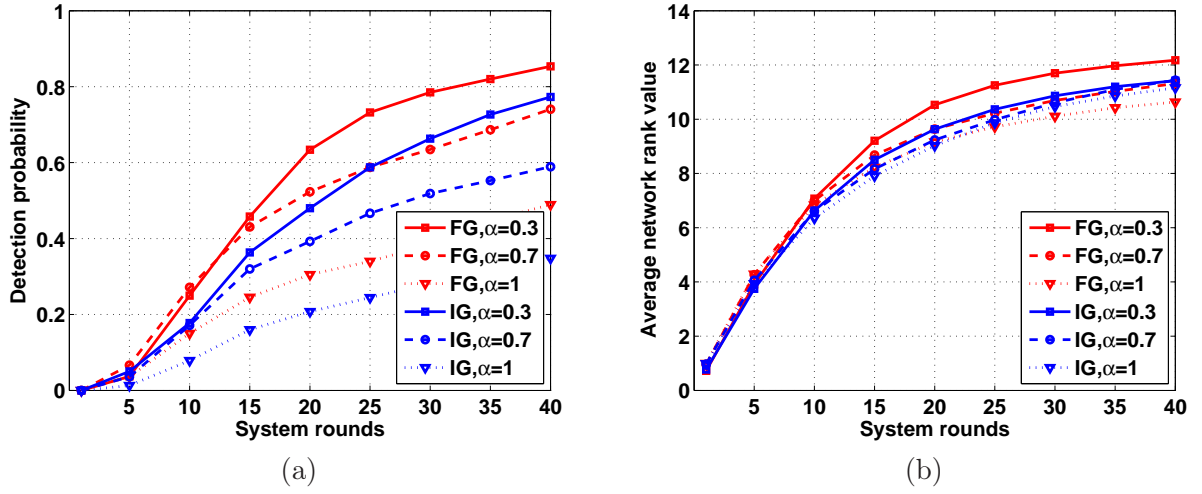


**Figure 3.9:** Simulation results for fully connected graphs with  $S = 20$  sensors,  $K = 1$ , where RM and DM denote the random and deterministic selection mode of master sensors, respectively. Top row: Probability of defective sensor detection. Bottom row: Average rank of messages received per sensor. Column (a): fixed values of the master sensors ( $L = 5$ ). Column (b): fixed values of the sensor participation constant ( $\alpha = qK = 0.7$ ).

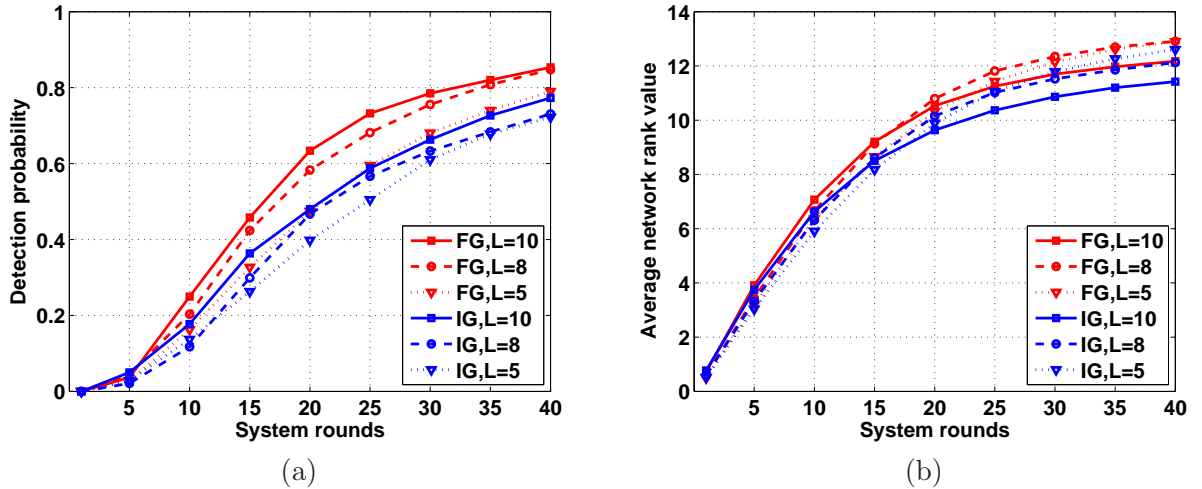
enough messages for detection. In the following, we focus on RM selection where possible (that is, for  $K = 1$ ), as it provides higher probability of creating innovative messages.

In Figs. 3.10 and 3.11 we present results for  $K = 2$  defective sensors and small networks with 20 sensors. The results are given in terms of the average detection probability over the dissemination rounds, for both fully and irregularly connected graphs. Master nodes are selected deterministically (DM) due to decoder condition for multiple defective sensors identification. Note that this example violates the condition  $K \ll S$  and the performance drops.





**Figure 3.10:** Simulation results for fully connected (FG) and irregular graphs (IG),  $d > 3$  with  $S = 20$  sensors,  $K = 2$  and deterministic selection (DM) of  $L = 5$  master nodes: (a) Probability of defective item detection; (b) Average rank value.



**Figure 3.11:** Simulation results for fully connected (FG) and irregular graphs (IG),  $d > 3$  with  $S = 20$  sensors,  $K = 2$  and deterministic selection (DM) of master nodes,  $\alpha = 0.3$ : (a) Probability of defective item detection; (b) Average rank value.

### 3.4.4 Communication overhead

For the sake of completeness, we analyze the communication costs of the proposed gossiping protocol and compare them with all other schemes under comparison. Let  $R_d$  and  $I_d$  denote the number of bits needed for raw measurements transmission and sensor identifier, respectively. Recall that

the tuple  $(S, L, L_n, n, \tau)$  stands for the number of sensors in the network, the number of master sensors (clusters), the number of master sensor neighbors, the average number of sensors per cluster ( $n = S/L$ ) and the total number of transmission rounds, respectively.

During the first phase of GP, the master sensors receive raw measurements from their neighbors. Thus,  $L_n \cdot R_d$  bits are used for communicating these values. Further, the master sensors create binary messages and send them to their neighbors. Every neighbor requires the knowledge of the test sensor identifiers, thus the cost is  $I_d \cdot \lceil q(L + L_n) \rceil$  bits, plus an additional bit in each message for sending the binary outcome result. Hence, the overall bit consumption is  $L_n R_d + L_n (I_d \lceil q(L + L_n) \rceil + 1)$ . Then,  $(S + 1)$  bits out of  $S(S + 1)$  bits in the message exchange phase are reserved for the test outcome and the test matrix row  $\mathbf{W}$ . Note that this analysis includes the full vector size and it can be further compressed. The overall number of transmitted bits over  $\tau$  rounds is:

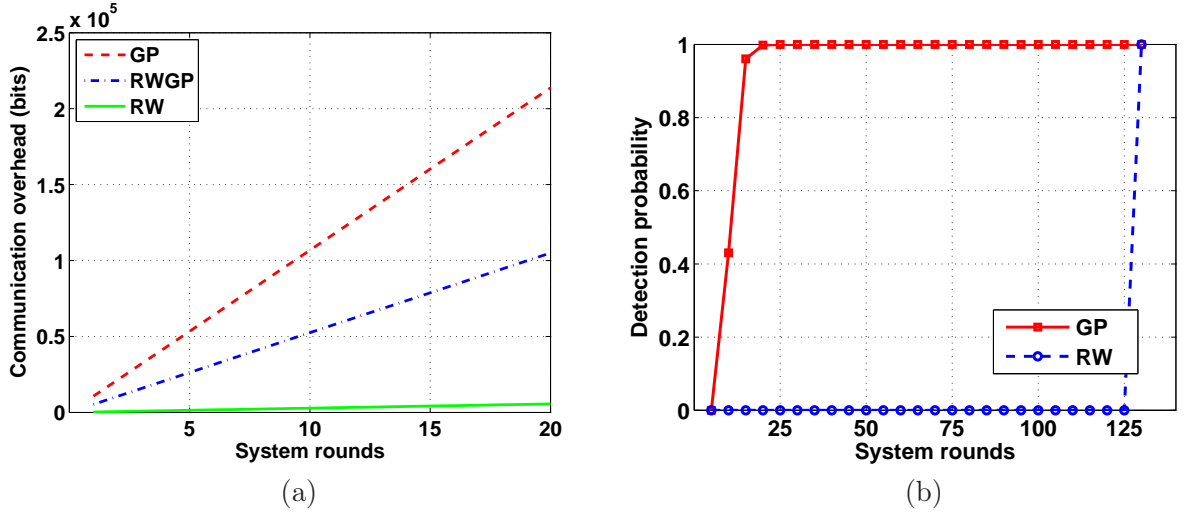
$$n_{GP}^b = \tau [L_n \{R_d + I_d \lceil q(L + L_n) \rceil + 1\} + S(1 + S)]. \quad (3.22)$$

We compare the communication costs of GP with the ones of RWGP that also has two phases. The first phase represents the random walk message collection, while the second is equivalent to the GP algorithm. Note that in the special case when RWGP and GP collect exactly the same data, they have identical decoding performance. However, if RWGP visits some sensors several times (this is more probable in irregular networks with a smaller connectivity degree), it performs worse than GP. In typical simulations, a random walk of RWGP terminates after  $n$ -th transmission round, where  $n$  is the number of elements per cluster in GP. RWGP transmits raw measurements, which results in  $\frac{(1+n)R_d}{2}$  bits. Therefore, the communication costs for RWGP is given by:

$$n_{RWGP}^b = \tau \left[ \frac{(n+1)R_d}{2} L + S(1 + S) \right]. \quad (3.23)$$

The bit transmission requirements for the *RW* algorithm is equivalent to that of the first step of RWGP, since it also transmits raw data. The detection is performed at nodes by comparison of known sensor values at that moment, excluding the message design step. The number of transmitted bits is equal to:  $n_{RW}^b = \tau \frac{(n+1)R_d}{2} L$ . Recall that one requires  $\log S$  transmissions for a message dissemination to all the nodes in a fully connected graph. Therefore, the SF algorithm requires in total  $n_{SF}^b = \tau R_d \log S$  bits.

The comparison between the proposed method and all other schemes regarding the bits spent in communication is illustrated in Fig. 3.12 for a fully connected graph. Note that the proposed algorithm in this setup requires only  $t = 15$  rounds for detection, but it consumes approximately three times more communication overhead compared to that of RWGP algorithm. However, due to the specific collection approach (hops), the duration of one transmission round of RWGP lasts ten times longer than that of the proposed algorithm. From the figure we can observe that the RW algorithm has very small communication overhead. However, it requires significantly higher number of rounds ( $S \log S \approx 130$  rounds) compared to the detection time of the proposed GP algorithm. Overall, the proposed GP scheme is able to compete with the other schemes in terms of used bits until detection.



**Figure 3.12:** (a) Comparison of the communication overhead for several algorithms, for the following parameter values:  $(S, L, L_n, \alpha, R_d, I_d, \tau) = (70, 5, 50, 0.7, 7, 7, 80)$ . Graph is fully connected. Abbreviations: GP: Proposed method, RWGP: Random Walk rounds with gossip algorithm and pull protocol dissemination, RW: Random Walk in the network initiated at  $L$  sensors. (b) Comparison of detection vs. number of rounds of the distributed detection scheme.

### 3.5 Conclusions

In this chapter, we have addressed the problem of distributed failure detection in sensor networks. We have proposed a novel distributed algorithm that is able to detect a small number of defective sensors in networks. We have designed a probabilistic message propagation algorithm that allows the use of a simple and efficient distance decoder at sensors. The transmitted messages are formed from local sensor observations and they are communicated using a gossip algorithm. For the worst case scenario we have derived the lower bound on the required number of linearly independent messages per cluster that sensors need to collect to ensure detection of one defective sensor with high probability. We have shown experimentally that this number is quite smaller in practice, even for the small size networks, which confirms the validity of the theoretical bound. The experimental results have shown that the proposed method outperforms other detection schemes in terms of successful detection probability. Finally, the convergence rate is very fast, which largely compensates for the higher communication overhead.



## Chapter 4

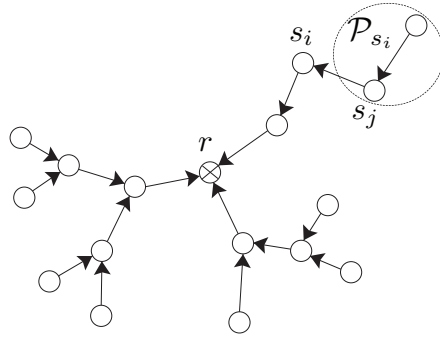
# Distributed data gathering and reconstruction

### 4.1 Introduction

Distributed sensor networks are often built by sets of low cost sensors. In general, the main task of such networks is to gather, recover and analyze sensor signals. Inexpensive sensors have a limited computational and transmission power. Therefore, the number of bits used for network message representation should be limited and the number of messages transmitted in the network should be limited. However, most of the common gathering and recovery algorithms collect a full set of messages from all the sensors and the number of bits per message in most algorithms is not carefully considered. For example, robust systems that perform random linear combinations of messages at sensors collect full sets of messages at the decoder and the decoder typically recovers the signal using the Euclidean elimination algorithm. This method is called Network Coding (NC) [22], [103], [104] and it is commonly used since it increases the message throughput in the network. Other algorithms perform signal decorrelation at sensors to increase the transmission efficiency, so only the innovative messages that are useful for decoding are transmitted towards the receiver. In such a setup, the decoder recovers the signal given the decorrelation design once that the full set of messages from all the sensors are available [42], [43].

In this chapter, we propose a novel and efficient distributed data gathering and reconstruction method for ad-hoc sensor networks that specifically takes into account system limitations in terms of number of messages and number of bits per message. In our framework the sensors use simple modular arithmetic to build network observations, which are represented with a limited number of bits. The proposed reconstruction algorithm then uses signal priors for recovery.

A sensor network is represented by a directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , as shown in Fig. 4.1. The set of vertices  $\mathcal{V}$  represents  $S$  sensors  $\{s_i\}$ ,  $1 \leq i \leq S$  and the edge set  $\mathcal{E}$  defines the sensor connectivity. We assume that the considered network is connected. The set of directed edges  $\mathcal{E}$  is defined by an asymmetric adjacency matrix, where the element  $e_{i,j}$  has a nonzero value if the sensor  $s_i$  communicates data towards  $s_j$ . The directed edges form data collection trees and a central receiver  $r$  collects the data. The parents  $\mathcal{P}_{s_i}$  of the node  $s_i$  are the neighbors of  $s_i$  with a larger hop distance to the receiver  $r$ . The sensors measure values of a scalar function  $f(s_i)$  that lives on



**Figure 4.1:** Directed sensor network. Parents  $\mathcal{P}_{s_i}$  of sensor  $s_i$  are sensors positioned farther than  $s_i$  from the receiver  $r$ .

the sensor network graph. These measurements can represent a physical phenomenon (pressure, temperature) or control data, for example. The central receiver  $r$  collects the set of measurements and reconstructs the function  $f$ .

In general, sensor measurements may be correlated. In order to minimize the communication costs, the sensors should ideally transmit only data that are innovative with respect to the information from other sensors. This is however hard to achieve in realistic settings since sensors have limited computational power and only a local view of the network. We therefore design a novel gathering and reconstruction framework. We exploit NC mechanism, which in a general scenario builds novel network messages by computing random linear combinations of the observed messages at sensors, which increases the robustness of the network. This motivated us to propose a system where the sensors randomly combine their measurement with the messages received by their parents, where the combinations performed at sensors use modular arithmetic. Messages are transmitted between the sensors along the directed path towards the receiver. Such gathering method uses a limited number of bits for message representation which reduces the communication costs and it ensures that the innovative information is conveyed to the receiver with high probability.

In the literature, message encoding is commonly performed in a ring  $F_q$  of size  $q$ , where a value  $q$  is equal to the power of a prime number (an existence of a unique multiplication element simplifies computations in this case). We remark that in this work the value of the modulo divisor  $q$  can be chosen arbitrarily, which increases the flexibility of the system. Though the optimization of the value  $q$  is out of the scope of our work, we use several examples to explain the reasoning behind building systems with an arbitrary  $q$  value. Modular operations ensure that the result takes values from the known interval of values, so the representation of sensor network messages requires a limited number of bits. The value of  $q$  may be chosen as the value that optimizes the trade-off between the consumed network energy and the reconstruction quality. Another way to choose  $q$  is related with the research efforts to design more efficient multi-level memory cells for information storage. Such memory cells have more than two states per bit. Development of reliable sensors that use these novel technologies could cause the replacement of the currently used 2-logic system by a multiple valued logic [4]. For sensors with multi-level memory cells, a logical choice of  $q$  is the power of the number of memory level states, where data processing operations at sensors perform the corresponding  $q$ -valued (or power of  $q$ ) logic.

We assume in this work that the sensor measurements  $x_i$ ,  $1 \leq i \leq S$ , are uniformly quantized and mapped by injection to the interval of integer values  $[0, q - 1]$ . Then, the data collection is performed as follows. We assign a coding coefficient (*weight*)  $w_i \in \{0, \dots, q - 1\}$  to every sensor  $s_i$ . Sensors at the border of the network (i.e., sensors that do not have any incoming link) initiate the data gathering. Messages are collected synchronously along the data collection paths. Each sensor performs weighted combinations of measurements using modular arithmetic. In particular, sensors perform modulo- $q$  addition and multiplication of the elements in the finite set  $\{0, 1, \dots, q - 1\}$  of size  $q$ , which builds a ring algebraic structure  $F_q$  (see def. of a ring algebraic structure in Appendix A.2). For instance, the message  $y_i \in F_q$  created at the sensor  $s_i$  is the encoding result of its current measurement and the messages  $y_j$  received from its parents  $\mathcal{P}_{s_i}$ . It reads

$$y_i = (w_i \odot x_i) \oplus \sum_{j \in \mathcal{P}_{s_i}} y_j. \quad (4.1)$$

Similarly, the message  $y_j$  has been constructed as the weighted combination of the measurement of the sensor  $s_j$  and data received from its parents  $\mathcal{P}_{s_j}$ . Note that the operators  $\odot$  and  $\oplus$  respectively represent the element-wise modulo- $q$  multiplication and addition. We remark that modular arithmetic builds messages whose bit representation is limited by construction. The data gathering terminates when  $M$  messages have reached the receiver. Finally, the messages at the receiver can be represented in a matrix form as

$$Y = \mathbf{W} \otimes X, \quad (4.2)$$

where  $X \in F_q^{S \times 1}$  represents a vector of sensor measurement values  $x_i$ ,  $1 \leq i \leq S$  and the vector  $Y \in F_q^{M \times 1}$  contains the received messages. The matrix  $\mathbf{W} \in F_q^{M \times S}$  describes the coding operations, while the operator  $\otimes$  denotes the operator that represents the combination of the operators  $\oplus$  and  $\odot$ . Remark that the  $i$ -th row of the coding matrix  $\mathbf{W}_{i,:}$  represents the coding vector used for building the  $i$ -th message  $y_i$ .

In general, the receiver collects partial network data,  $M < S$ . The reconstruction problem at the receiver thus consists in reconstructing the sensor data from a small number of messages with help of priors about the signal under observation. In other words, the decoder has to determine the vector  $X$  that both satisfies the constraints from Eq. (4.2) and fits the data model. In the following sections, we study the decoding error for the proposed data gathering system. We analyze the properties of the random coding matrix and implement a practical message-passing algorithm for signal reconstruction.

In summary, the contributions of this chapter are the following:

- We propose a novel framework for effective gathering of the sensor data with network coding, which ensures signal recovery with high probability. Sensors form messages by combining their measurements with messages received from neighbor nodes using modulo operations and transmit them along directed communication paths. The receiver reconstructs the data from an underdetermined system of coded messages with help of sensor data priors.
- We develop an analytic decoding error expression as a function of the number of collected messages and the partition function values.
- We compute the system performance when sensor data is sparse and locally correlated. We

show that the decoding error stays small even if a small number of messages are collected by the receiver.

- We next discuss the coding strategies that increase the decoder accuracy and relate them to the decoding error expression.
- We propose a novel signal recovery algorithm based on the Belief Propagation (BP) [105] decoder. To estimate sensor measurements, the decoder uses the data collected by the receiver, the sensor priors and the signal correlation model dependent on the network topology.
- Simulation results that are provided for several types of network topologies and signals demonstrate that the proposed algorithm outperforms or competes with non-encoded and wavelet lifting [43] gathering algorithms in terms of the reconstruction quality or bit consumption.

The rest of this paper is organized as follows. The analytic decoder error bound is developed in Section 4.2. In Section 4.3, we analyze and discuss the design of the coding strategy that decreases the decoding error. In Section 4.4 we provide a practical reconstruction algorithm based on a Belief Propagation algorithm. Finally, in Section 4.5 we provide the simulation results and discussions.

## 4.2 General decoder performance bounds

In this section we analyze the probability of the decoding error given the messages collected by the proposed gathering framework. We assume that the sensor signal  $X \in F_q^{S \times 1}$  and the decoded signal  $\hat{X} \in F_q^{S \times 1}$  belong to a class of signals denoted by  $\mathcal{F}$ . This class represents the full set of signals that match the data model.

We assume that sensors have correlated measurements with their spatially close neighbors. Then, for a given topology correlation model based on the sensor distances, we define  $\mathcal{F}$  as the set of all the signal function vectors with locally correlated values. For example, one class of signals that fit our model is the set of all the signals with  $K$  nonzero elements, where these nonzero values are measured by a spatially close neighbors.

A reconstruction error occurs when exists such a signal  $\hat{X}$  in  $\mathcal{F}$  that matches the coding conditions, i.e.,  $Y = \mathbf{W} \otimes X = \mathbf{W} \otimes \hat{X}$ , where  $\hat{X}$  is different from  $X$ . This analysis corresponds to the worst case approach and the decoding error probability reads

$$p(\hat{X}|X) = p\left(\hat{X} \in \mathcal{F}, \text{ s.t. } Y = \mathbf{W} \otimes \hat{X} \text{ and } \hat{X} \neq X\right) \leq \sum_{\hat{X} \in \mathcal{F}} p\left(\hat{X} \text{ s.t. } Y = \mathbf{W} \otimes \hat{X}\right). \quad (4.3)$$

The work in [62] uses a similar setup for expressing the general decoding error probability when the coding is performed in a Galois field, where the  $K$ -sparse signal and coding coefficients have i.i.d. uniform probability mass function (pmf). We study here the decoding performance for a more generic framework for rings of arbitrary size and with an arbitrary coding matrix distribution. The computation of the performance bounds is quite different in such a generic setup.



We assume that messages in the network are encoded independently. Then, the decoder error probability defined by Eq. (4.3) reads

$$p(Y = \mathbf{W} \otimes \hat{X}) = \prod_{i=1}^M p\left(y_i = \mathbf{W}_{i,:} \otimes \hat{X}\right), \quad (4.4)$$

where  $\mathbf{W}_{i,:}$  represents the  $i$ -th row of the coding matrix  $\mathbf{W}$ . We now analyze the error message encoding event, that occurs if  $\mathbf{W}_{i,:} \otimes \hat{X} = y_i = \mathbf{W}_{i,:} \otimes X$ . By subtracting the most left and the most right part of the last expression, the error event becomes  $\mathbf{W}_{i,:} \otimes (\hat{X} \ominus X) = 0$ . Thus, the error event probability reads

$$p\left(y_i = \mathbf{W}_{i,:} \otimes \hat{X}\right) = p\left(\sum_{s=1}^S W_{i,s} \odot (\hat{x}_s \ominus x_s) = 0\right). \quad (4.5)$$

Therefore, for messages collected by the receiver, the decoder error occurs if

$$\sum_{s=1}^S W_{i,s} \odot (\hat{x}_s \ominus x_s) = 0 \quad (4.6)$$

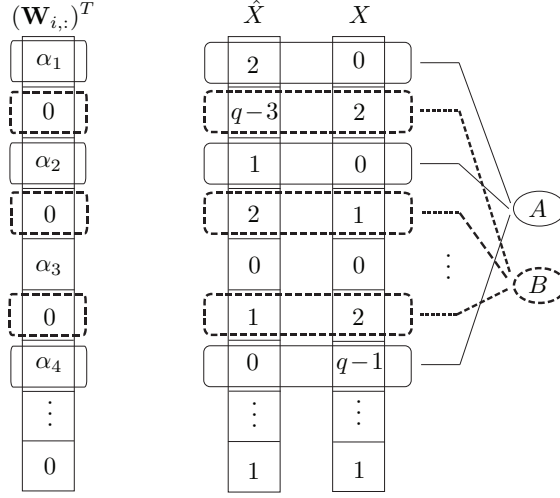
holds. Computing the decoder error probability boils down to building systematically the total error event. Once the full set of error events is known, the decoder error probability is computed as the sum of the probabilities of events in this set that generate a decoding error.

To simplify the analysis of the error event in Eq. (4.6), we consider that the coding matrix belongs to the general family of random matrices whose elements take values

$$c_W = \begin{cases} \alpha, & \text{with probability } p_w, \\ 0, & \text{with probability } 1 - p_w, \end{cases} \quad (4.7)$$

where  $\alpha \in \{F_q \setminus 0\}$  is any nonzero element of the ring and  $p_w \in ]0, 1[$ . We denote the encoding matrix element  $W_{i,j}$  by  $c_W$  to simplify the notation. We distinguish two instances that build the error event, illustrated in Fig. 4.2. The first instance occurs when the nonzero coding elements multiply nonzero values of the difference vector (full line ellipsoids in Fig. 4.2) and the number of such coding elements that form the  $i$ -th message is  $A$ . The second error event instance occurs when the zero coefficients multiply the nonzero elements of the difference vector of  $X$  and  $\hat{X}$ . The number of such coding matrix events is  $B$  and they are illustrated by dashed ellipsoids in Fig. 4.2. Then, the error event is built by  $A$  nonzero and  $B$  zero summands in  $F_q$  whose total sum is equal to zero in modular arithmetic. Since in our model there are in total up to  $k \in \{1, \dots, K\}$  differences between the statistically known model and the signal in  $\mathcal{F}$ , two vectors in the class of signals  $\mathcal{F}$  differ in up to  $2k$  positions and the values of  $A$  and  $B$  are given by  $A \in \{1, \dots, 2K\}$ ,  $B \in \{0, \dots, 2K\}$ .

We further denote by  $e_A$  and  $e_B$  the events corresponding to the cases where  $A$  non-zero summands and  $B$  zero summands build the error event. The size of these events is denoted by  $|\mathcal{S}(e_A)|$  and  $|\mathcal{S}(e_B)|$ , respectively. Then, we compute the probability that an error event  $e_i$  occurs



**Figure 4.2:** To simplify the error event probability calculation, we split the analysis into the cases depending on the values of the encoding matrix elements that multiply the signal elements at the positions where the vectors  $X$  and  $\hat{X}$  differ. Here,  $A$  is the number of nonzero coding matrix coefficients that multiply the elements of vectors  $X$  and  $\hat{X}$  in positions where they differ. The value  $B$  is the number of zero encoding coefficients that multiplies the nonzero elements of the vector obtained by subtracting the vectors  $X$  and  $\hat{X}$ .

as the sum of all the combinations of the events  $e_A$  and  $e_B$  as

$$p(e_i) = \sum_{k=1}^K \sum_{A=1}^{2k} \left\{ \left( p(c_W \neq 0) \right)^A |\mathcal{S}(e_A)| \left( p(c_W = 0) \right)^{2k-A} |\mathcal{S}(e_B)| \right\}, \quad (4.8)$$

where the coding matrix coefficients are defined by Eq. (4.7). We now compute the expected size of the sets  $|\mathcal{S}(e_A)|$  and  $|\mathcal{S}(e_B)|$ . Once they are computed, the decoding error  $p(\hat{X}|X) \leq (p(e_i))^M$  is known.

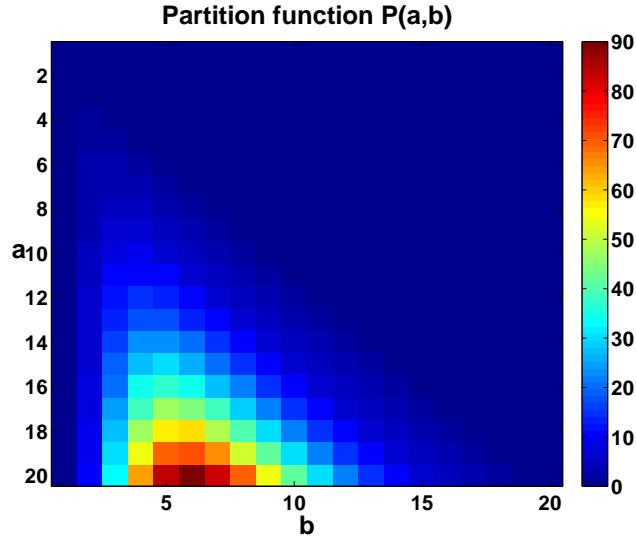
The size of the set  $\mathcal{S}(e_B)$  is simply given as

$$|\mathcal{S}(e_B)| = |\hat{\mathcal{X}}_B|, \quad (4.9)$$

which is the cardinality of the set of possible vectors in  $\mathcal{F}$  with  $B$  arbitrary values at positions multiplied by zero coefficients in the coding vector. Note that the actual signal values at corresponding positions of the event  $e_B$  are arbitrary, since their values are multiplied by zero coefficients  $c_W$ .

We compute the expected size of the set  $\mathcal{S}(e_A)$  as

$$|\mathcal{S}(e_A)| \leq |\hat{\mathcal{X}}_A| \sum_{m=1}^A \sum_{l=1}^A \sum_{n=1}^{q-1} \frac{p\left(\mathcal{C}(P((m-1)q, n, l))\right)}{p\left(\mathcal{C}(P((m-1)q, n))\right)}, \quad (4.10)$$



**Figure 4.3:** The partition function values  $P(a, b)$  for the pairs of values  $a$  and  $b$ , where  $a, b \in \{1, \dots, 20\}$  define the number of representations of the number  $a$  with  $b$  summands. Equivalently, they define the number of representations of the number  $a$  with an arbitrary number of summands, where the maximum summand is equal to  $b$ . These values form the lower triangular matrix.

where  $|\hat{\mathcal{X}}_A|$  counts the total number of vectors in  $\mathcal{F}$  with  $A$  nonzero values at positions where  $X$  and  $\hat{X}$  differ. It multiplies the expected number of error event realizations for each particular  $\hat{\mathcal{X}}_A \subset \mathcal{F}$  to give a bound on the size of  $e_A$ . The error events depend on the partition function values. The value of the partition function  $P(a, b)$  gives the number of possible representations of the integer  $a$  with  $b$  integer summands. Equivalently, it defines the number of representations of  $a$  with arbitrary summands where the biggest summand is equal to  $b$ . This function is nonlinear and its values  $P(a, b)$  for  $a \in \{1, \dots, 20\}$  and  $b \in \{1, \dots, 20\}$  are illustrated in Fig. 4.2. Next, the function  $P(a, b, c)$  represents the restricted partition of the representations of the number  $a$  with  $c$  summands, where the biggest element in the sum is  $b$ .

We illustrate the values of the partition function and the restricted partition function in a few examples. Number four can be represented by a following list of summands

$$\begin{aligned}
 4 &= 4 \\
 &= 3 + 1 \\
 &= 2 + 2 \\
 &= 2 + 1 + 1 \\
 &= 1 + 1 + 1 + 1,
 \end{aligned}$$

so the partition function value of the number 4 with 2 being the biggest summand is  $P(4, 2) = 2$ , which corresponds to the cases  $4 = 2 + 2$  and  $4 = 2 + 1 + 1$ . This value also gives the number of representations of the number 4 with two summands, which corresponds to the cases  $4 = 3 + 1$

and  $4 = 2 + 2$ . Also, the value of the restricted partition value  $P(4, 3, 2) = 1$  gives the partitions number for the number four with two summands, where the biggest summand value is three. This case corresponds to the case  $4 = 3 + 1$ . Similarly,  $P(4, 3, 3) = 0$ .

In Eq. (4.10), the function  $\mathcal{C}(P)$  defines the number of all the possible combinations of summands, for a given number of realizations of a partition function and a given number of summands of each representation. For example,  $\mathcal{C}(P(3, 3, 2)) = 2$  indicates that exist two ways to order the summands for representing the value four with two summands, where the biggest summand is equal to three. These representations are given by  $4 = 3 + 1$  and  $4 = 1 + 3$ .

As a consequence of the circularity of the modulo product in a ring of arbitrary size, certain values occur with higher probability. In the previous equation,  $p$  denotes the probability mass function (pmf) of the product of two random variables  $\mathbf{V} = \mathbf{W} \mathbf{X}$ , which are determined by the distributions of the coding vector and signal vector values. To compute the pmf of the random variable  $V$ , we perform real field operations, which are then mapped to the values of the ring. Therefore, the pmf of  $\mathbf{V}$  is computed in two steps. In the first step, the pmf of  $\mathbf{V}$  is computed by transform techniques (details available in Appendix A.3), as if it would represent the product of two random variables in the real field. Then, the values of the resulting real field random variable  $\mathbf{V}$  are mapped to the set of values  $\{0, \dots, q - 1\}$  using the modular arithmetic. The probabilities of the random variable realizations that have the same congruent modulo  $q$  (same reminders) are summed up together. Remark that the influence of the data model  $\mathcal{F}$  is not explicitly visible in the error term. It however drives the number of partitions and it constraints the values in  $\hat{\mathcal{X}}_A$ .

By combining Eqs. (4.8 - 4.10), we obtain the probability  $p(e_i)$  of the error event  $e_i$ . Finally, by inserting this result into Eq. (4.3) we can compute the error probability bound by

$$p(\hat{X}|X) \leq p(e_i)^M. \quad (4.11)$$

Recall that, even if the influence of the data model is not explicit in this last relation, the form of the data in the set  $\mathcal{F}$  drives the probability  $p(e_i)$  as given in Eqs. (4.9) and (4.10).

If two vectors from the class of signals  $\mathcal{F}$  differ in exactly  $2K$  positions (instead of up to  $2K$ , as considered so far), the interval of values used in computing the decoder error changes and it reads

$$p(e_i) = \sum_{A/2=1}^K \left\{ \left( p(c_W \neq 0) \right)^A \mathcal{S}(e_A) \left( p(c_W = 0) \right)^{2k-A} \mathcal{S}(e_B) \right\}. \quad (4.12)$$

This error is generally smaller than the decoding error for signals that have up to  $2K$  differences, as expected.

### 4.3 Coding matrix analysis

The coding strategy in our data gathering framework influences the decoding error, as shown by the decoding error expression in the previous section. We here investigate the important characteristics of the coding matrix  $\mathbf{W}$  that decrease the value of the decoding error and we discuss some practical aspects in its design. Coding matrix design has been studied extensively for half of the century in the coding, sampling and information theory communities [106] generally under the communication

constraints (e.g., maximum rate or capacity). Here, we rather consider the encoder design from a sampling theory perspective.

In classical sampling theory, the *stability* is the main property of a good sampling (coding) operator  $\mathbf{W}$ . This property itself implies *invertibility*. However, the stability condition does not hold for matrices  $\mathbf{W}$  that perform modulo- $q$  operations in  $F_q$ , as in our gathering framework, because the modulo encoding operations are nonlinear. In general, we study the problem of sensor signal recovery in the network with  $S$  sensors given  $M < S$  messages collected by the receiver, where the signal priors are given. Therefore, the encoding matrix is underdetermined. Instead of the invertibility property, we study the rank and the separability property of the encoding matrices whose elements are in  $F_q$ . We show that a good encoding matrix has a high rank value and that the separability property improves the decoding. These properties bring constraints on building the matrix rows (high rank property) and columns (separability property). Intuitively, the encoding matrices with a high rank are favorable since they ensure that every message collected by the receiver is innovative. However, since  $\mathbf{W}$  is underdetermined, the encoding pattern of sensors in the messages needs to differ. This is achieved by forming separable encoding matrices. Additionally, if the sensor and signal priors are available, the encoding matrix can be optimized to maximize the decoding performances. In this section we therefore study the rank and separability property of matrices that perform encoding in a ring algebraic structure  $F_q$ . We show that, if the coding matrix has a high rank and it is separable with high probability, the decoder error probability  $p(\hat{X}|X)$  decreases.

### 4.3.1 Matrix rank

We study the dependence between the rank of the coding matrix  $\mathbf{W}$  and the decoding error. We recall that matrix rows encode network messages, so independent rows of the encoding matrix guarantee building of innovative messages. We first recall the classical matrix rank definition that holds for matrices in the real field and we provide a definition that holds for matrices in  $F_q$ . We then show that the matrix rank value depends on the partition function values, which also drives the decoding error expression.

In a real field, the matrix rank is equal to the number of linearly independent rows (or columns) of the matrix. In the ring  $F_q$  with modulo addition and multiplication, the matrix rank is equal to the maximal number of rows whose weighted sum is different than zero in modulo- $q$  arithmetic. We now study the rank value of a  $j \times S$  matrix  $\mathbf{W}$ , where  $j \in \{1, \dots, M\}$ ,  $M < S$ , where the matrix elements are chosen as in Eq. (4.7). For this purpose, we compute the probability that amongst  $j$  rows some of the rows are dependent

$$p(\text{rank}(\mathbf{W}) < j) = p\left(\sum_j d_j \odot W_{j,:} = 0\right), \quad (4.13)$$

where  $\mathbf{d} = [d_1 \dots d_j] \in F_q^{j \times 1}$  is a random vector that has at least one non-zero value, while the operations are modulo- $q$  multiplication and addition. Let  $k_L$  be the expected value of the number of dependent rows (or columns) in the  $M \times S$  matrix  $\mathbf{W}$ . Let  $k_r$  and  $k_c$  denote the expected number of dependent rows and columns in a ring  $F_q$ , respectively. Then, the value  $k_L \in \{k_r, k_c\}$  and the expected matrix rank is equal to  $\text{rank}(\mathbf{W}) = \min(M - k_r, S - k_c)$ .

A zero value in modulo  $q$  space may correspond to a real value in  $\{0, q, \dots, \lceil MS p_w \rceil q\}$ , where  $\lceil MS p_w \rceil$  is the expected number of nonzero elements in the  $M \times S$  matrix  $\mathbf{W}$ . Recall that  $p_w$  denotes the probability that a coding matrix element chosen as in Eq. (4.7) takes a nonzero value. The expected number of nonzero elements in a matrix row and a matrix column are  $\lceil S p_w \rceil$  and  $\lceil M p_w \rceil$ , respectively. Then, the probability that, amongst  $j \leq M$  rows, some of them are dependent is equal to

$$p(\text{rank}(\mathbf{W}) < j) = \sum_{A=1}^{\lceil j \cdot S p_w \rceil} \sum_{m=1}^A \sum_{n=1}^{q-1} p(\mathcal{C}(P(q(m-1), n, A))), \quad (4.14)$$

and the expected value of the number of dependent rows in the matrix is then given by

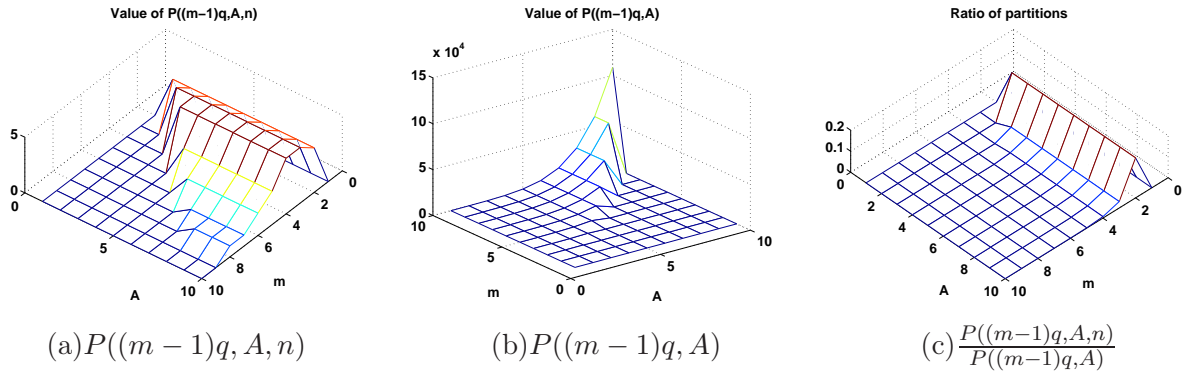
$$k_r = \sum_{j=1}^M \sum_{A=1}^{\lceil j \cdot S p_w \rceil} \sum_{m=1}^A \sum_{n=1}^{q-1} p(P(q(m-1), n, A)) \cdot \mathcal{C}(P(q(m-1), n, A)), \quad (4.15)$$

where  $\mathcal{C}(\cdot)$  lists all the combinations of the partition function values that can occur. We similarly compute the expected number of linearly dependent columns, where the probability of the number of dependent columns is

$$p\left(\sum_j d_j \odot W_{:,j} = 0\right) = \sum_{j=1}^S \sum_{A=1}^{\lceil j \cdot M p_w \rceil} \sum_{m=1}^A \sum_{n=1}^{q-1} p(\mathcal{C}(P(q(m-1), n, A))).$$

Now, the rank maximization problem is equivalent to the minimization of  $k_r$  (or  $k_c$  if  $S - k_c < M - k_r$ ). The expected number of dependent rows in Eq. (4.15) is proportional to the partition function. Since we showed in Section 4.2 that the decoder error probability is dependent on the partition function values, maximization of the matrix rank decreases the decoder error rank. Therefore, to maximize the rank of the matrix whose elements take values from the set  $\{0, \dots, q-1\}$  decreases the decoding error since it imposes an innovative design of messages collected by the receiver.

The partition function builds the expression for the computation of the number of dependent rows and columns. We now illustrate its values for a fixed set of parameters to understand its nonlinear behavior. We assume that the random variable  $\mathbf{V} = \mathbf{W}\mathbf{X}$  is given and that the pmf of the partition function is uniform. The random variables  $\mathbf{W}$  and  $\mathbf{X}$  are determined by the distributions of the coding vector and signal vector values. We compute the partition function values and its ratios for multiple pairs of parameters  $(K, q)$ , where  $K$  is the number of nonzero elements and  $q$  is the size of the ring  $F_q$ . We observe that the resulting partition function value has a similar behavior for different parameter pair values. We illustrate the values of the partition function ratios in Fig. 4.4 for  $(K, q) = (5, 8)$ , where we see that the partition function is maximal when the message values match the interval values  $(q, 2q-1)$ . We want to avoid the resulting message value to lie in this interval since this case maximizes the decoding error probability. Similarly, Fig. 4.5 shows that the ratio  $P(a, b, c)/P(a, b)$  has the maximum value if the biggest ring element of the ring element  $(q-1)$  takes part in the partition of the number  $(m-1)q$ . We conclude from these figures that the coding matrix should not encode multiple correlated sensor readings within a single message, since this event most probably causes an error. The second observation is that the error is highest when the message value maps to the biggest ring value  $(q-1)$ ; this can be exploited for the design



**Figure 4.4:** Dependence of the partition functions and their ratio  $\frac{P((m-1)q, A, n)}{P((m-1)q, A)}$  from the set of parameters  $(A, m)$ , plotted for the set of parameters  $(K, q) = (5, 8)$ . Note that  $A$  defines the number of nonzero summands in the partition function and  $m$  defines the value of number to be partitioned (equal to  $(m-1)q$ ). The value  $m = 3$  ( $m-1 = 2$ ) defines the real field interval with the highest values of the partition function in (a) and the ratio in (c).

of the coding matrix when the signal properties are known a priori, as shown in Sec. 4.3.3.

### 4.3.2 Disjunct and separable matrices

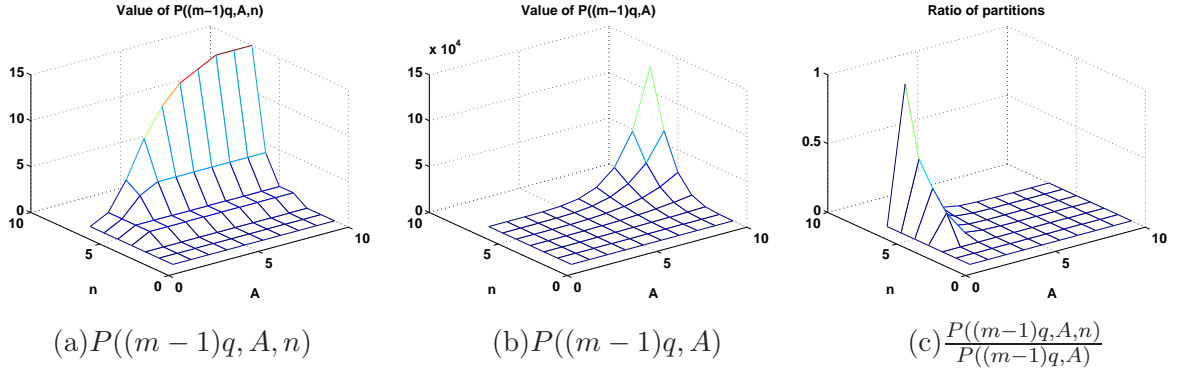
We have shown previously that the matrix rank influences the value of the decoding error. The high rank guarantees that all the messages collected by the decoder are novel. We now study the matrix properties which ensure different sensor encoding patterns in messages as well as their relation with the decoding error probability expression. We show that the disjunct and separable matrices represent good candidates for coding matrix designs whose elements take values from the ring  $F_q$ .

In coding theory, binary coding matrices that lead to high probability decoding have often disjunct or separability properties [107], [108]. In short, a binary matrix is  $K$ -disjunct if the Boolean sum of every  $K$  columns does not contain any other column in the matrix, while the  $K$ -separable property is fulfilled if the Boolean sum of every  $K$  columns is unique. Note that  $K$ -disjunct matrices are also  $K$ -separable, while the reverse does not hold. For the sake of completeness, we provide formal definition of such properties for binary matrices and operators in the Appendix A.4. Note that binary matrices with  $q = 2$  designed by Eq. (4.7) are disjunct with high probability.

For matrices whose elements take values from the ring  $F_q$  where  $q > 2$ , disjunct definition does not hold. We thus define now the separability property for matrices in arbitrary size of the ring  $F_q$  and analyze its influence on decoding performance.

**Definition 1.** *Separable matrix in the ring  $F_q$  of arbitrary size: A  $M \times S$  matrix  $\mathbf{W}$ , whose elements take values from the ring of arbitrary size  $q$ , is  $K$ -separable if and only if  $\forall \mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V} = \{s_1, \dots, s_S\}$  and  $|\mathcal{S}_1| = I, |\mathcal{S}_2| = J, I + J \leq K$ , it holds that*

$$\sum_{i \in \mathcal{S}_1} d_i \odot W_{:,i} \neq \sum_{j \in \mathcal{S}_2} d_j \odot W_{:,j},$$



**Figure 4.5:** Dependence of the partition functions and their ratio  $\frac{P((m-1)q, A, n)}{P((m-1)q, A)}$  from the set of parameters  $(A, n)$ , plotted for the set of parameters  $(K, q) = (5, 8)$ . The parameter  $A$  defines the number of nonzero summands in the partition function and  $n$  determines the maximum summand value used for partitioning the number  $(m-1)q$ . The error occurs most probably for  $n = q - 1$ , which is the highest value that  $n$  can take in a ring  $F_q$  of size  $q$ .

where  $i \in \mathcal{S}_1$  and  $j \in \mathcal{S}_2$ , vectors  $\mathbf{d}_I = [d_1 \dots d_I]$  and  $\mathbf{d}_J = [d_1 \dots d_J]$  are non-zero vectors whose elements take values from the interval  $\{0, \dots, q-1\}$ .

In other words, for a matrix to be  $K$ -separable, any weighted sum of up to  $K$  columns in modulo- $q$  arithmetic should differ from any other column so that the following holds

$$\sum_{k=1}^{I+J} d_k \odot W_{:,k} \neq 0. \quad (4.16)$$

Matrices with this property encode novel messages with high probability, so intuitively, the decoder error decreases. Therefore, the matrix is separable with high probability if the probability  $p(\sum_{k=1}^{I+J} d_k \odot W_{:,k} = 0)$  is small. This expression provides constraints on the columns of the encoding matrix, while the condition in Eq. (4.13) sets constraints on the matrix rows. We conclude that matrices with high rank or matrices that are separable in rings of arbitrary size minimize the decoding error with high probability.

### 4.3.3 Coding matrix design with a given signal pmf

We now discuss the problem of the design of the coding matrices which keeps the general decoder error probability low, where the sensor measurement priors  $\{p_{s_i}\}_{i=1}^S$ , the set  $X \in \mathcal{F}$ , the size of the ring  $q$  and the network topology are given. We first provide details about computations of the pmf  $p_V(\mathbf{V})$  of the random variable  $\mathbf{V} = \mathbf{X}\mathbf{W}$ , where the random variable  $\mathbf{X}$  corresponds to the random variable that models the sensor values and  $\mathbf{W}$  the coding matrix value. Remark that the realizations of the random variable  $\mathbf{V}$  take values from the interval  $(0, q-1)$  and not in  $(0, (q-1)^2)$ , since the message encoding is performed using a modulo- $q$  arithmetic. To simplify the computations of  $p_V(\mathbf{V})$  we compute the realizations of the random variable  $\mathbf{V}$  and its probability as for a real random variable and then we map the realizations and the corresponding probabilities



to the interval  $(0, q - 1)$  using the modular arithmetic. We now provide details of the described method. This can be used to compute the values of the coding matrix given the signal priors, which minimizes the decoder error probability.

We first compute the distribution of a random variable  $V = XW$ . The formal distribution of a product of two random variables, as given in Appendix A.3, requires that those variables are defined on positive values. Therefore, we shift the variable values interval from  $\{0, \dots, q - 1\}$  to  $\{1, \dots, q\}$  without loss of generality. We perform all the operations considering these shifted values and in the final step we map the results back to their appropriate values. With this mapping, our novel random variable takes values in the range  $(1, q^2)$  instead of  $(0, q^2 - 1)$ . Its values are computed as follows:

$$\begin{aligned} p_V(v = 1) &= \sum_{x=1}^q p_X(x)p_W\left(\frac{1}{x}\right) = p_X(1)p_W(1), \\ &\dots \\ p_V(v = q^2) &= \sum_{x=1}^q p_X(x)p_W\left(\frac{q^2}{x}\right) = p_X(1)p_W(q^2) + \dots + p_X(q)p_W(q), \end{aligned} \quad (4.17)$$

Since the distributions are discrete, the non-zero values of  $p_W(\frac{v}{x})$  are those that fulfill  $\frac{v}{x} \in \mathbb{Z}$ . Next, we perform modulo  $q$  summations of the values of  $p_V$ , because the values of our new random variable are from the range  $(0, q - 1)$ . We obtain the novel distribution value  $Q(p_V)$  and we use its pmf in computations. To build a coding matrix pmf that can minimize the decoding error probability  $p(\hat{X}|X)$ , we minimize the probability of the occurrence of an error event  $e$ . We define  $e$  as a the event that the sum of  $k \in \{1, \dots, 2K\}$  weighted differences of two vectors in  $\mathcal{F}$  is equal to zero in modulo  $q$  arithmetic, see Fig. (4.2). The goal is to minimize the probability of the decoding error. We first provide the example how to compute the probability of the error event when  $k = 2$ ,  $p(e)_{k=2}$ . Let  $v_1$  and  $v_2$  be the realizations of the random variable  $V$ . We have

$$p(e)_{k=2} = 2(p(v_1 = 1, v_2 = q - 1) + \dots + p(v_1 = \frac{q}{2}, v_2 = \frac{q}{2}))$$

which sums the probabilities of all the the decoding error events (here  $k = 2$ , so all the pairs of elements whose modulo- $q$  sum equals zero causes the decoding error). We rewrite the joint probabilities as  $p_V(v_1, v_2) = p_V(v_1)p_V(v_2)$  because they are independent. Each of the probabilities  $p_V(v = i)$  can be rewritten as the probabilities of pairs of  $(x, w)$ , where  $i = xw$ , so that

$$p(e)_{k=2} = 2\left(\sum_{x=1}^q p_X(x)p_W\left(\frac{1}{x}\right) \sum_{x=1}^q p_X(x)p_W\left(\frac{q-1}{x}\right) + \dots + \sum_{x=1}^q p_X(x)p_W\left(\frac{\lfloor \frac{q}{2} \rfloor}{x}\right) \sum_{x=1}^q p_X(x)p_W\left(\frac{\lfloor \frac{q}{2} \rfloor}{x}\right)\right). \quad (4.18)$$

Similarly, we compute the probabilities for  $e$ ,  $k > 2$ . The optimization of the coding matrix consists in choosing the coefficient that minimize the probability of an event  $e$  and reads

$$\arg \min_W p(e)_{\forall k}.$$

We find the optimal values for weight matrix by computing the discrete derivatives of  $p(e)_{\forall k}$  over

all the coefficients

$$\begin{aligned} \frac{\Delta p(e)_{\forall k}}{\Delta p_{\mathcal{W}}(w_0 = 0)} &= 0, \\ &\dots \\ \frac{\Delta p(e)_{\forall k}}{\Delta p_{\mathcal{W}}(w_q = q - 1)} &= 0, \end{aligned} \quad (4.19)$$

where  $\Delta$  is a discrete derivative operator. These equations provide a determined system of linear equations that can be solved for every particular network. This computations help to analyze good matrix properties for different signal classes.

#### 4.3.4 Probabilistic encoding matrix design

In this section we build the encoding matrix that we use in simulations. We want to build the matrix that is separable with high probability and has a high rank.

A probabilistic encoding matrix design propagates the probabilistic message encoding at sensors. It allows a lot of flexibility in the selection of its coefficients (it can be performed directly by the sensors). The number of linearly dependent rows and columns in matrices should still be minimized (maximal rank). We have seen previously that the decoding error is smaller if a limited number of measurements from locally distributed sensors are combined together, see Fig. 4.4 (separability property). Therefore, we want to build a matrix with a high rank that encodes the sensor measurement from different correlation sets, which are imposed by the topology correlation model. In our work, we assume that the signal follows the topology correlation model. For example, it can be represented by all up to  $K$  locally positioned sensors. Once the correlation model is defined, we can build the set of possible correlated values. Ideally, sensors whose measurements take part in message encoding should belong to different correlation sets. This problem may be observed as an covering problem. The covering problem consists in defining  $M$  different ways to cover the area using different patterns. Here, the value  $M$  is the number of encoded messages collected by the receiver. The covering area corresponds to the signal space in the encoding problem. Every covering attempt corresponds to a single message encoding. A single covering patch corresponds to one clique set (a patch is used for covering a particular area). Since a single sensor can belong to several signal correlation realizations, several patches are designed to cover the same area (patches may overlap). The message encoding problem becomes the problem of covering the area by different sets of randomly chosen patches, where no patches can overlap. Once the space is covered, the patches are removed and the next covering attempt takes place. In the next covering attempt we can not choose the elements that have been used already.

By design, the covering problem is a binary problem. For binary matrices, one of the convenient ways to build the  $M \times S$  matrices, where  $M < S$  whose rank is close to  $M$  with high probability is a disjunct matrix design. In our case, the nonzero values of the binary matrix are replaced by the nonzero elements of a ring  $F_q$ . Then, matrix elements are chosen probabilistically by:

$$c_W = \begin{cases} \alpha, & \text{with probability } \frac{p_w}{q-1} \text{ and } \alpha \in \{1, \dots, q-1\} \\ 0, & \text{with probability } 1 - p_w, \end{cases} \quad (4.20)$$

where  $\alpha \in \{F_q \setminus 0\}$  leads to the finite field disjunct matrix construction.

---

**Algorithm 1** PD Algorithm: probabilistic coding matrix design

---

**Input**  $(K, p_w, M)$   
**Initialize**  $\mathcal{C}_L$   
**while**  $i < M+1$  **do**  
    Choose the subset  $\mathcal{C}_l$  from  $\mathcal{C}_L$  that covers the signal space using the criteria  $\mathcal{C}$   
    Assign a weight value  $c_W$  to each sensor according to the Eq. (4.20)  
     $\mathcal{C}_L \leftarrow \mathcal{C}_l$  (remove the subset  $\mathcal{C}_l$  from  $\mathcal{C}_L$ )  
     $i \leftarrow i + 1$   
**end while**  
**Return** coding matrix  $\mathbf{W}$

---

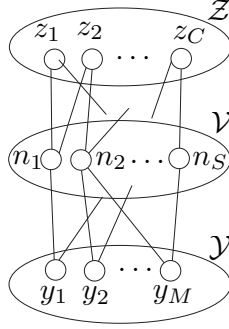
The algorithm is described in Algorithm 1. The input values of the algorithm are the sparsity value  $K$ , the probability to choose a nonzero field element  $p_w$ , the topology correlation model (the union of all possible correlated sets builds the set  $\mathcal{C}_L$ ), the number of messages  $M$  that are collected by the receiver and the covering problem criteria  $\mathcal{C}$  (minimal number of elements, maximal number of elements, etc. can be designed for different networks). Based on this criteria, we choose the subset of patches  $\mathcal{C}_l$  used for the message encoding. Then, any sensor in the patch is chosen randomly and the value of the coding matrix is drawn randomly accordingly to Eq. (4.20). The subset of patches  $\mathcal{C}_l$  is removed from  $\mathcal{C}_L$  and the process is repeated  $M$  times. Such design of the encoding matrix decreases the decoding error, since it avoids the error events that are illustrated by high values of the functions in Figs. 4.4 and 4.5.

We now discuss the coding matrix properties in more details. Intuitively, sparse and probabilistic coding matrices have interesting properties for efficient data gathering, especially if the signal is sparse itself. However, the sparsity condition alone does not assure recovery of the signal from a small number of messages. The encoding matrix should have a good covering property. On the other side, good covering properties assure that the signal space is spanned, but does not provide sparsity sampling matrix guarantees. Overall, the probabilistic design is particularly useful for distributed sensor networks and it designs matrices with sufficiently high ranks.

## 4.4 Modified Belief Propagation reconstruction algorithm

In the previous sections, we have developed the decoding error expression based on the worst case analysis and discussed the encoding matrix properties that minimize the decoding error. Since the decoder search space depends on the given set of vectors  $\mathcal{F}$ , it may be very large. To recall, the decoder forms the search space that consists of all the pairs of vectors in  $\mathcal{F}$  that differ in up to  $K$  positions and it performs the exhaustive search. Since the search space grows quadratically with the number of elements in the set  $\mathcal{F}$ , the decoding is computationally expensive. Therefore, in this section we propose a constructive decoding algorithm for signal reconstruction.

We propose a novel algorithm for signal reconstruction based on a Belief Propagation (BP) decoder. The layer BP algorithm denoted by BPL reconstructs a sensor signal, given the sensor priors, the topology model and the network coded messages that are gathered by the receiver node.



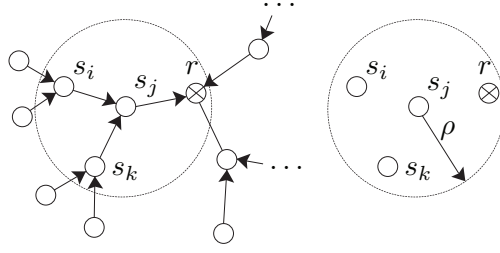
**Figure 4.6:** The structure of Belief Propagation algorithm denoted used for signal reconstruction.

Comparing with the classical BP, the novel part of the BPL algorithm represents the layer of nodes that introduces constraints on the topology correlation model and we describe it below in more details.

The layer BP algorithm architecture consists of variable nodes, two types of factor nodes and the edges between them, as shown in Fig. 4.6. The variable nodes  $\mathcal{V} = \{n_i\}_{i=1}^S$  represent the sensors themselves. The total set of the factor nodes  $\mathcal{U}$  consists of the measurement check nodes  $\mathcal{Y}$  and the structure check node set  $\mathcal{Z}$ , so  $\mathcal{U} = \bigcup (\mathcal{Y}, \mathcal{Z})$ . The measurement check nodes  $\mathcal{Y} = \{y_b\}_{b=1}^M$  examine if the decoded values satisfy data constraints given by Eq. (4.2) and the structure check nodes  $\mathcal{Z} = \{z_c\}_{c=1}^C$  verify the solution feasibility based on the topology correlation model. The connections between  $\mathcal{V}$  and  $\mathcal{Y}$  are defined by the nonzero elements of the coding matrix  $\mathbf{W}$ , while the connections between  $\mathcal{V}$  and  $\mathcal{Z}$  are determined by a topology correlation set. The messages transmitted over the graph edges represent beliefs that a particular sensor takes a certain value. We first provide details about the topology correlation model used for constructing the nodes  $\mathcal{Z}$  and their connections with  $\mathcal{V}$ . We then provide definitions of messages formed at the nodes  $\mathcal{V}$ ,  $\mathcal{Y}$  and  $\mathcal{Z}$  and finally, we describe the algorithm. We note that the messages computed at nodes  $\mathcal{V}$  are used for decoding the signal.

We briefly describe the topology correlation model that is used to define the connectivity between the structure check nodes  $\mathcal{Z}$  and variable check nodes  $\mathcal{V}$ . We assume that locally distributed sensors may have correlated measurements. We use this model to define a total set of potentially correlated neighbor sensor sets, called cliques. A graph *clique*  $\mathcal{C}$  [109], [110] represents the subset of neighboring vertices  $\mathcal{C} \subset \mathcal{V}$  and provides a consistent and convenient modeling of dependent entities, e.g., sensor measurements. In the sensor network, the set of neighbors  $\mathcal{N}_{s_j} = \{s_i \in \mathcal{N} | \text{dist}(s_i, s_j) \leq \rho, s_j \neq s_i\}$  of the sensor  $s_i$  represents sensors located at a maximal distance  $\rho \in \mathbb{R}^+$  from  $s_j$ . The total set of graph cliques  $\mathcal{G}_{\mathcal{C}} = (\mathcal{V}^*, \epsilon^*)$  is given by a union of cliques  $\mathcal{C}$  of different degrees, as illustrated in Fig. 4.7. Using the concept of cliques, we group together locally distributed sensors that could have correlated values. We denote this set by  $\mathcal{F}$ . Extraction of this set is network dependent and therefore expensive.

We describe below the layer BP algorithm illustrated in Fig. 4.6. Message exchanges between nodes are performed iteratively. We represent symbolically the message exchanges between the nodes in a single iteration of the BP algorithm by the sequence  $\mathcal{V} \rightarrow \mathcal{Y} \rightarrow \mathcal{V} \rightarrow \mathcal{Z} \rightarrow \mathcal{V}$ . The iteration starts at variable nodes that send messages towards the measurement check nodes, which



**Figure 4.7:** The topology correlation model is based on the assumption that the sensors which are locally positioned may have correlated values. The neighborhood set  $\mathcal{N}_{s_j} = \{s_i, s_k, r\}$  build sensors positioned within a radius  $\rho$  from the sensor  $s_j$ . In this illustration  $\text{dist}(s_i, s_k) < \rho$  and  $\text{dist}(s_i, r) > \rho$ . Thus the collection of clique sets in this neighborhood corresponds to a union of the set of single elements  $\mathcal{C}_1 = \{s_i, s_j, s_k, r\}$ , pairs  $\mathcal{C}_2 = \{(s_i, s_j), (s_j, s_k), (s_j, r)\}$  and triplets  $\mathcal{C}_3 = \{(s_i, s_j, s_k), (s_k, s_j, r)\}$ . The pair  $(s_k, r)$  is not correlated in the considered neighborhood, since it does not contain the sensor  $s_j$ .

compute new beliefs and send them back to the variable nodes. The variable nodes update their values and forward messages towards the structure check nodes. They attempt to decode the signal using the topology-based correlation model and send their values back to the variable nodes. This concludes one iteration of the decoding algorithm. The full algorithm is listed in Algorithm 2 and we describe it in further detail below.

The BPL algorithm starts by the initialization of the variable node messages with sensor priors  $\mu_{s_i \rightarrow y_j} \leftarrow p(s_i)$ , while the check node and the structure check node messages are set to zeros  $\mu_{y \rightarrow s_i} \leftarrow \mathbf{0}$  and ones  $\mu_{z \rightarrow s_i} \leftarrow \mathbf{1}$ ,  $i \in \{1, \dots, S\}$ , respectively. The messages are sent from the variable nodes  $\mathcal{V}$  towards the check nodes  $\mathcal{Y}$ . In the first iteration, the messages sent to the check nodes are equal to the prior sensor values. In other iterations, the message transmitted from the variable node  $s_i$  towards the check node  $y_j$  depends on the subset of neighboring check node messages, different to the current destination node. It reads

$$\mu_{s_i \rightarrow y_j} = \gamma_{s_i} \prod_{y_k \in \{\mathcal{N}(s_i) \setminus y_j\}} \mu_{y_k \rightarrow s_i} \prod_{z_t \in \{\mathcal{N}(s_i) \setminus z_r\}} \mu_{z_t \rightarrow s_i}, \quad (4.21)$$

where  $\gamma_{s_i}$  normalizes the outgoing messages at sensor  $s_i$ . We denote by  $\mathcal{N}(y_j) \setminus s_i$  the neighbors of the check node  $y_j$  that belong to the set of variable nodes different than  $s_i$ . The belief messages at measurement check nodes are calculated by

$$\mu_{y_j \rightarrow s_i} = \sum_{\sim s_i} f(\mathcal{N}_{y_j}) \prod_{s_l \in \{\mathcal{N}(y_j) \setminus s_i\}} \mu_{s_l \rightarrow y_j}, \quad (4.22)$$

where the binary function  $f(\mathcal{N}_{y_j})$  is set to one only if the measurement check node conditions are satisfied. The symbol below the summation,  $\sim s_i$ , denotes that the summation is performed over all the sensors different than  $s_i$ . The novel variable node messages depend on the messages from

**Algorithm 2** Layer Belief Propagation algorithm

---

**Initialize** variable and check node messages  $\mu_{s_i \rightarrow u_j}(s_i) \leftarrow p(s_i)$ ,  $\mu_{u_j \rightarrow s_i} \leftarrow \mathbf{0}$   
**Initialize** constants (flag  $\leftarrow 0$ , iter  $\leftarrow 0$ , maxit), structure set  $\mathcal{F}$  and sets  $\mathcal{L}, \mathcal{H} \leftarrow \emptyset$   
**while** iter < maxit or flag = 0 **do**  
  **Update** messages at measurement check nodes  $\mathcal{Y}$ , Eq. (4.22);  
  **Update** messages at variable nodes  $\mathcal{V}$ , Eq. (4.23);  
  Find uniquely decoded sensor values at variable nodes from the set  $\mathcal{L} \leftarrow \text{Unique max}(\mu_{s_i \rightarrow z_j}(s_i))$   
  **if**  $|\mathcal{L}| = S$  and  $\mathcal{L} \neq \emptyset$  **then**  
     $\hat{X}(s_i) = \max(\mu_{s_i \rightarrow z_j}(s_i))$ ; flag  $\leftarrow 1$ ; **break**  
  **else if**  $0 < |\mathcal{L}| < S$  **then**  
    Compute  $\mathcal{H} = \{\mathcal{F} | \text{valid structure subset } \mathcal{L}\}$   
    **for**  $h \in \{1, \dots, |\mathcal{H}|\}$  **do**  
      Find the sensor values for the missing set of sensors  $\mathcal{L}_h$  by matching the structure priors that take into account the existing set of decoded sensor values  
      **if**  $|\mathcal{L} \cup \mathcal{L}_h| = S$  **then**  
        Update structure messages  $\mu_{z_t \rightarrow s_i}$  as in Eq. (4.24) and variable messages  $\mu_{s_i \rightarrow y_j}$  defined by Eq. (4.21); flag  $\leftarrow 1$   
      **end if**  
    **end for**  
     $\hat{X}(s_i) = \max(\mu_{s_i \rightarrow u_j}(s_i))$ ; **break**  
  **end if**  
  iterate  $\rightarrow$  iterate + 1  
**end while**  
**return**  $\hat{X}$

---

the measurement messages

$$\mu_{s_i \rightarrow z_r} = \gamma_{s_i \rightarrow z_r} \prod_{y_k \in \{\mathcal{N}(s_i) \setminus y_j\}} \mu_{y_k \rightarrow s_i}, \quad (4.23)$$

The structure check nodes receive messages from the variable nodes and use the correlation model and the subset of formerly decoded values to compute

$$\mu_{z_k \rightarrow s_i} = \sum_{\sim s_i} f(\mathcal{N}_{z_j}) \prod_{s_l \in \{\mathcal{N}(z_k) \setminus s_i\}} \mu_{s_l \rightarrow z_k}, \quad (4.24)$$

which is then sent to the variable nodes. The novel variable node messages are computed as in Eq. (4.21) and the decoded value is finally

$$\hat{X}(s_i) = \arg \max_{s_i} \mu_{s_i \rightarrow y_j}(s_i). \quad (4.25)$$

The algorithm terminates if the signal is decoded. If this is not the case, the iterative process defined by Eqs. (4.21)-(4.25) carries on until one of two termination conditions are fulfilled: the unique signal is determined or the maximum iteration step is reached.

For the signals collected by the proposed gathering method, the convergence of classical BP algorithms is not guaranteed. The algorithm diverges in certain cases because the collected messages form an underdetermined system of equations at decoder. Hence, our modified reconstruction algorithm uses information about topology and signal, where the latter is typically a smoothness or sparsity prior. With this modification, the performance of the decoder is improved, even if the number of collected messages stays low. We note that BPL decoder has the limitation imposed by the encoding error. The number of computations grow exponentially with the number of non-zero values in rows of the coding matrix. We limit this value in practice.

## 4.5 Simulation results

In this section we compute the bounds for the analytic decoder error and discuss them. Next, we analyze the performance of our layer BP decoder and compare it with several state-of-the-art gathering methods.

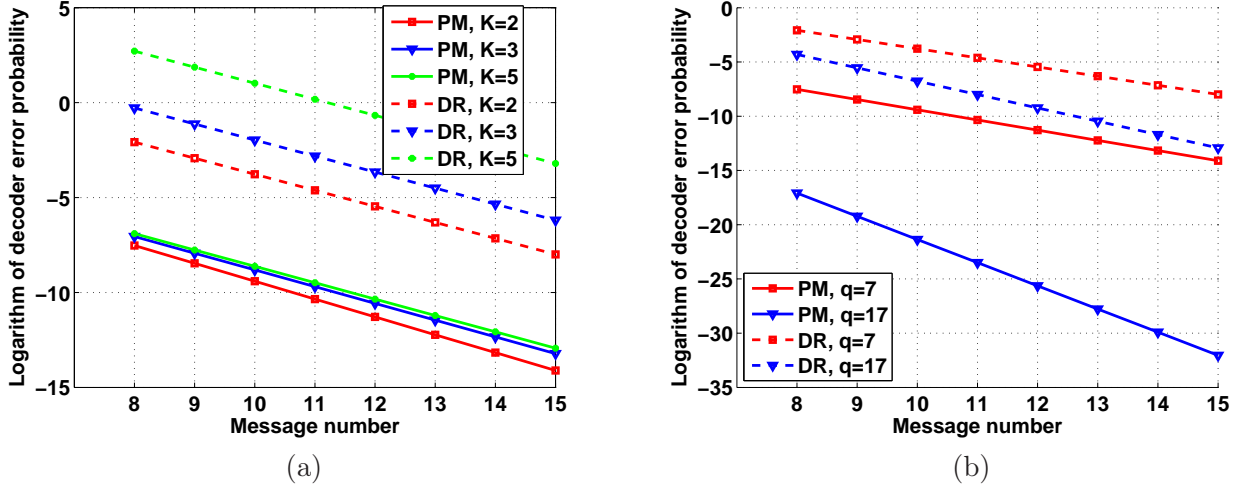
### 4.5.1 Simulation setup

We first describe the experimental setup used in our simulations. We consider networks with a line array (LA) or an irregular topology. In the latter case, the sensors are randomly distributed geographically and the data gathering paths represent the shortest path criteria between the sensors and the receiver in terms of hop distance. We append the coding vector to every sensor in the networks as in Eq. (4.20) in order to keep track of the coding operations.

For all the networks, several classes of signal priors are used in the experiments. We denote by  $U_q$  the uniform discrete probability mass function (pmf) defined on the interval  $(0, q - 1)$  that represents the distribution of values of each sensor. Alternatively, the sparsity pmf  $S_q(p_w)$  generates non-zero values with a small probability  $p_w$  and those non-zero values are distributed uniformly in the interval of nonzero ring  $F_q$  values  $(1, q - 1)$ . Signals in sensor networks used for environmental monitoring very often have a small number of non-zero values that are locally positioned, while the rest of the values are zeros. Therefore, we adopt this data model in our simulations and we assume that the class of signals of interest  $\mathcal{F}$  is formed by signals with up to  $K$  non-zero values that are grouped locally on the graph. We then consider a Gaussian model (DG) where the set of measurements take values in the finite set of elements, with a Discrete Gaussian probability mass function (pmf) defined on the interval  $(0, q - 1)$  according to a shifted and re-normalized discretized Gaussian distribution. Finally, we consider a Discrete Laplace (DL) pmf for the signal values, which preserves the useful properties of the Laplace distribution [111] such as simple closed form and stability. This distribution is widely used for modeling events in environmental sciences, signal processing (speech and image) and medicine, to name a few.

### 4.5.2 General decoder performance bounds

We first study the decoder error bound and analyse the performance of the proposed framework in different settings. We assume that the values of the coding matrix are chosen uniformly at random, similarly to the coding matrix design used in [62] and we consider that the signal has at most  $K$  non-zero values that form local clusters in the network.



**Figure 4.8:** Error decoding probability (log scale) for line array sensor network with parameters  $(S, K, q, \text{sparse signal distribution})$ . PM: proposed method bound; DR: Draper et al. bound (a) PM and DR are calculated using parameter values  $(20, [2, 3, 5], 7, \text{Uniform})$ . (b) PM and DR are calculated using parameter values  $(20, 2, [7, 17], \text{Uniform})$ .

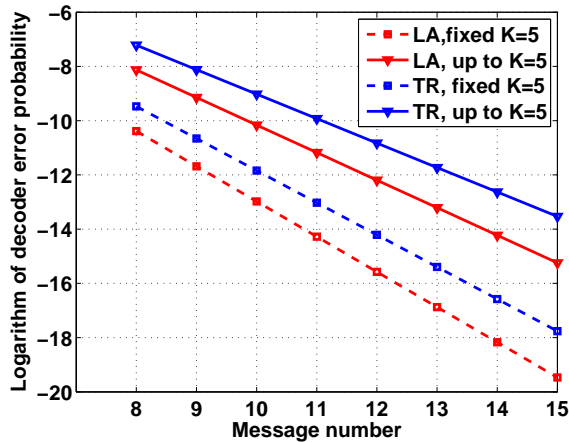
First, we study the evolution of the performance bound as a function of the number of messages, for different sparsity levels (i.e., different values of  $K$ ). In order to help the evaluation, we also show the bound given in [62]

$$p(\hat{X}|X) \leq K 2^{SH_B(K/S)} (q-1)^K q^{-M}, \quad (4.26)$$

where  $H_B(K/S)$  is the binary entropy function. This decoding error has been developed for a framework where linear combinations are performed in  $GF(q)$  for prime values of the field size  $q$ , where the elements of the matrix  $\mathbf{W}$  are chosen uniformly at random in a Galois field  $GF(q)$  and the signals are chosen uniformly from the set of sparse signals. Note that our bound in Eq. (4.11) can be seen as a generalization of the bound in Eq. (4.26), since we allow network combinations to be performed in a ring of arbitrary size  $q$  using modulo  $q$  operations. Remark that, for the same set of the assumptions (parameters, linear combinations in Galois field and data model) both bounds actually match.

Figure 4.8(a) illustrates the error decoder performance of the method in [62] denoted by DR and for our setup (PM), for different instances of the parameter set  $(S, K, q, \text{signal model})$ . The lower error values for PM is a consequence of set cardinality: the cardinality of the set  $\mathcal{F}$  is smaller when data is locally correlated than in the case where the set of  $K$ -sparse signals have non-zero values that are arbitrarily distributed. We compare these two bounds for different sizes of the ring. We clearly see that the decoding error decreases for larger values  $q$  of the field size. Note that we chose prime values for the field size  $q$  for the sake of comparison with the framework in [62]; however, our framework could use any value for  $q$  and the results actually follow the same tendency as the one shown in Figure 4.8(b).



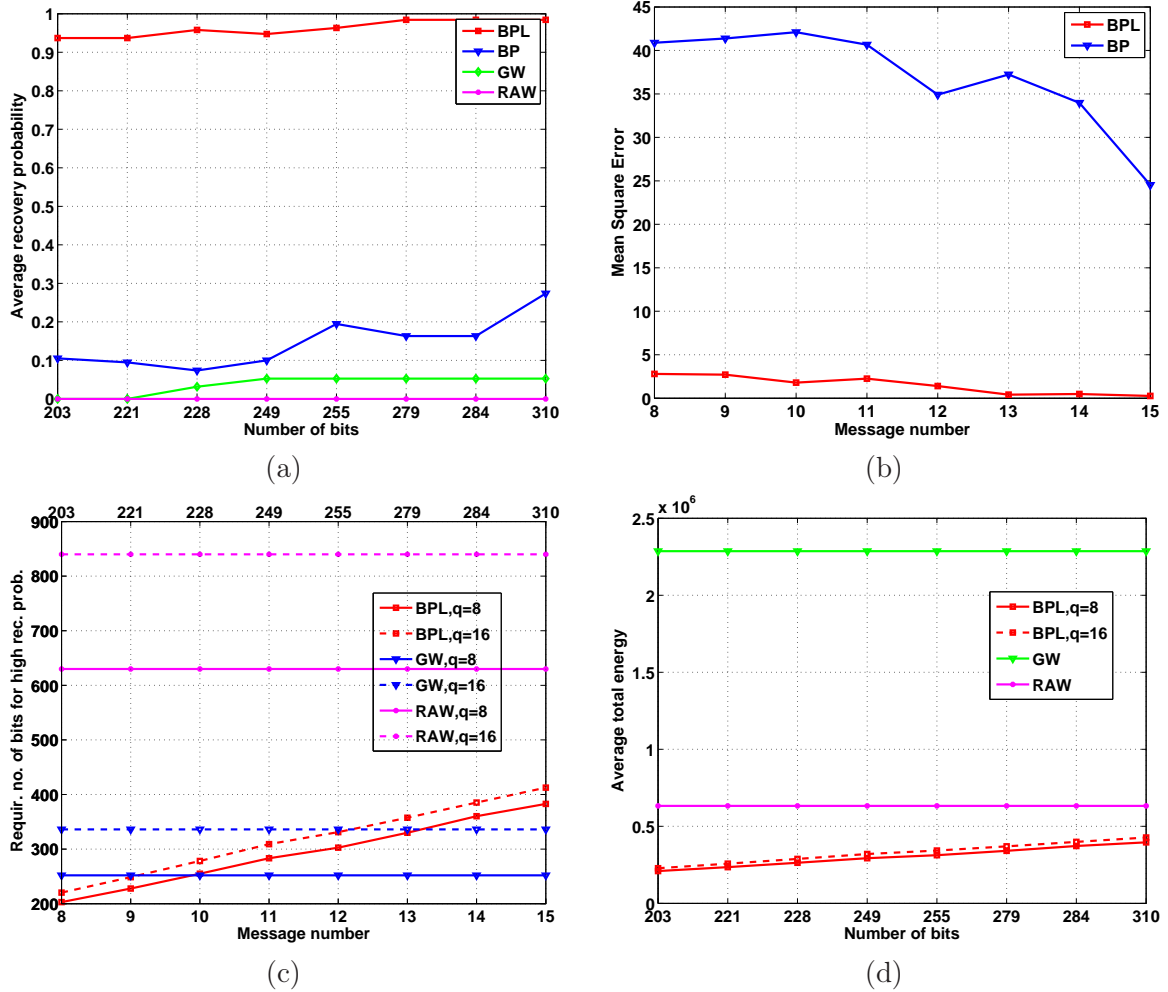


**Figure 4.9:** Error decoding probability (log scale) for line array and tree-based sensor network with parameters:  $(S, K, q, \text{sparse signal model}) = (20, 5, 8, \text{DiscreteLaplacian})$ . The error is calculated for sparsity of  $K$  and up to  $K$ .

Finally, we want to study the influence of the data model on the performance bounds. We consider a second class of signals, where the number of non-zero values is exactly  $K$ . These non-zeros values are again grouped locally on the graph. The computations of our performance bounds in Section 4.5.2 is adapted in this case by putting  $k = K$  exactly, and by choosing  $A$  even in  $\{2, \dots, 2K\}$  (i.e., two different vectors with fixed  $K$  may differ only in an even number of positions). Fig. 4.9 illustrates the decoding error probability for both signal models and for line array (LA) and tree (TR) network. The tree network has roughly 30% more connections between sensors than the line array network. As expected, we see that the error is smaller for signals whose sparsity is fixed, as the set of possible signals is smaller in this case, hence the decoding error is reduced. For the same reasons, smaller values of sparsity (i.e.,  $K$ ) leads to smaller decoding error probabilities. Finally, we observe that, in all cases, the error bounds decrease exponentially with the number of messages, which is very important for building effective data gathering solutions.

### 4.5.3 Layer BP performance

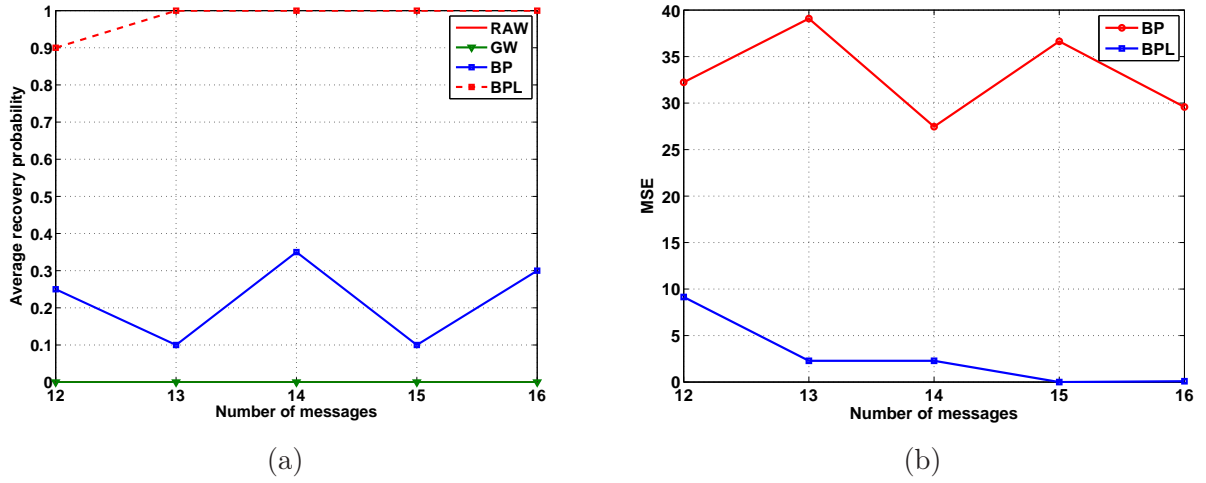
We now provide simulation results to illustrate the performance of our novel BP decoder, which is able to consider topology and signal priors to improve the signal results. We study the recovery as a function of the number of bits transmitted in the network. We first consider a sensor network with  $S = 20$  sensors, where  $K = 2$  non-zero values are locally distributed. The simulation results represent the mean value of  $N = 10$  random network realizations. The probability that the coding vector elements take values different to zero is heuristically fixed to  $p_W = 0.2$ . The connections between the variable nodes and the measurement check nodes in the BP decoder correspond to the non-zero ring elements of the coding matrix  $\mathbf{W}$ . The modulo operations are performed in the ring  $F_q$ , where  $q \in \{8, 16\}$ . The proposed layered BP reconstruction algorithm is denoted by BPL, while the classical BP based reconstruction for signals of fixed sparsity is denoted by BP. We compare



**Figure 4.10:** Performance analysis for LA networks with parameters  $(S, K, N, q, p_w) = (20, 2, 10, 8, 0.2)$ . (a) Average recovery probability; (b) Average Mean Square Error; (c) Bit consumption; (d) Average energy consumption.

the proposed method with the RAW and GW algorithms. In the RAW algorithm, the transmitted messages represent non-compressed quantized sensor measurements that are simply forwarded by the nodes towards the receiver.  $GW^1$  represents gathering method based on wavelets, proposed in [43]. In general, we give the simulation results for a small number of sensors in the network, since the complexity of the BPL algorithm grows exponentially with the number of non-zero values in rows of the coding matrix, which is a general property of the classical BP algorithms. However, the benefits of using the proposed layer BP show for such networks in terms of the bit consumptions. Current ongoing analysis for medium networks is promising.

<sup>1</sup>Thanks to S. Narang, G. Chen and A. Ortega for the code. Provided GW results in our work are obtained by adaptation of the original code to our network conditions.

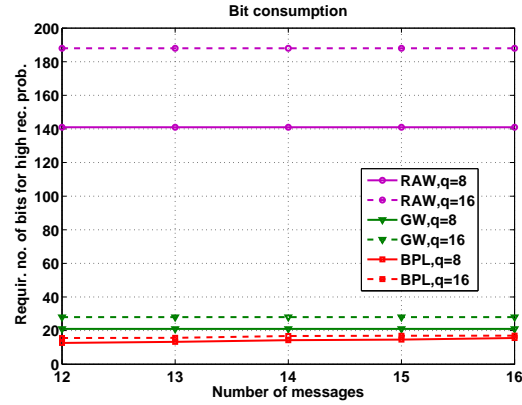


**Figure 4.11:** Performance analysis for TR networks with parameters  $(S, K, N, q, p_w) = (20, 2, 10, 8, 0.2)$ . (a) Average recovery probability; (b) Average Mean Square Error;

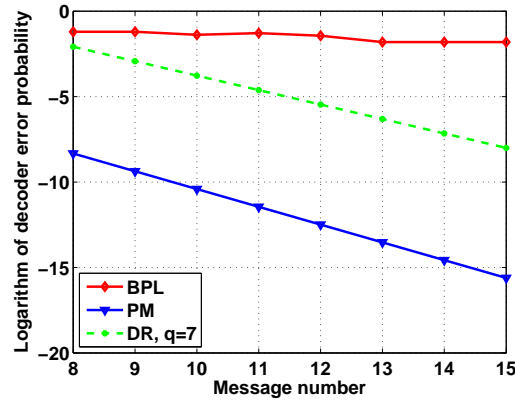
In Fig. 4.10 we illustrate the decoding performance for line array networks. The values given on the  $x$ -axis represent the number of messages (bits) that are transmitted in the network and used in reconstruction by the proposed BPL method. In particular, the number of messages from the set  $\{8, \dots, 15\}$  correspond to the number of bits in  $\{203, \dots, 310\}$  used for decoding. Fig. 4.10(a) shows the reconstruction probability. Recovery probability of RAW and GW methods is small since they require a fixed number of messages for signal recovery whose bit representation is larger than the number of messages (bits) gathered by layered and classical BP based algorithms. The average Mean Square Error (MSE) value for the BPL method is smaller than for the BP, as illustrated in Fig. 4.10(b). Fig. 4.10(c) then shows required number of bits for recovery with high probability vs. the number of bits consumed by the proposed BPL method. The methods RAW and GW are able to recover the signal only upon receiving all the messages, therefore, they require a fixed number of bits for recovery. These methods require larger number of bits than the proposed BPL algorithm. Finally, we analyze the total energy requirements that are modeled as in [112]. It considers that energy is spent on the channel message transmission as well as at the sensors for receiving and transmission of messages. The total energy consumption for the proposed method is lower than for the other systems, as shown in Fig. 4.10(d).

In Figs. 4.11 and 4.12 we illustrate the results for networks with tree topologies. The proposed BPL method recovers the signal with high probability, where the number of transmitted bits is quite smaller than for GW, BP and RAW message gathering algorithms. The average recovery probability for of the RAW and GW methods are zero, because they require a full set of messages for recovery. This value is larger than the number of bits for recovery by BPL algorithms. For tree networks, the number of message transmission hops needed for reaching the receiver drops compared to the values in line arrays with the same number of sensors. Therefore, the number of bits per message transmission decreases.

In Fig. 4.13 we illustrate the recovery probability values for theoretical bounds and the exper-



**Figure 4.12:** Bit consumption for TR network with parameters  $(S, K, N, q, p_w) = (20, 2, 10, 8, 0.2)$ .



**Figure 4.13:** Theoretical and experimental values of recovery probability. The considered setup is defined by the set of parameters  $(S, K, N, q, p_w, \text{signal prior}) = (20, 2, 10, 8, 0.2, \text{Discrete Laplacian})$ .

imental BPL method. The BPL algorithm however does not have convergence guarantees. The relatively high error value for small number of messages is caused by the practical limitations of the BPL algorithm, since the number of iterations of the algorithm are limited. Also, the complexity of BP-related algorithms grows exponentially with the number of non-zero values in the rows of the coding matrix, so the coding values vectors are chosen carefully.

Overall, the set of experiments performed for small sensor networks illustrates that the proposed BPL algorithm provides accurate reconstruction given a small number of sparsely encoded messages in the networks. Classical algorithms for the same networks require more bits for recovery, since they may recover signals only upon receiving all the sensor messages.

## 4.6 Conclusions

We have addressed in this chapter the problem of efficient data collection for sensor networks. We have designed the gathering method that codes messages at sensors during the data gathering phase. The message collection initiates at sensors on the network borders and the proposed network encoding scheme leads to innovative message collection with high probability. We have developed an analytic expression for the decoding error and analyzed its dependence on network parameters. Increasing the size of the ring  $F_q$  and narrowing the potential solution set by more precise signal priors decreases the decoding error. Next, we have investigated the coding matrix properties that decrease the analytic decoder error and proposed its practical probabilistic design. Finally, we have proposed a practical Belief Propagation decoder algorithm for signal reconstruction that exploits sensor priors and the topology knowledge. We have shown experimentally that, for signals with locally correlated values, the proposed algorithm outperforms or is competitive with state-of-the-art gathering methods in terms of the number of bits transmitted in the network that are necessary for high probability signal recovery.



## Chapter 5

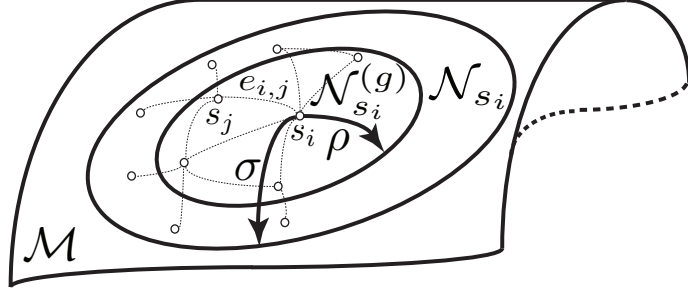
# Distributed interpolation of sensor network signals on manifolds

### 5.1 Introduction

Sensor networks are very often used to monitor some physical phenomena (e.g., a temperature) and interpolation methods are commonly deployed to estimate the signal values at positions where no sensors are located. Commonly, the sensors are distributed in a large geographical area and accurate processing algorithms should take into account the geometry of the manifold where the sensors live. However, in scenarios where the interpolation point set is arbitrary, the manifold geometry is often neglected or approximated by a plane.

In this chapter, we study the problem of distributed data interpolation in sensor networks that takes into account a manifold geometry, where interpolation is performed at an arbitrary set of locations. We consider that irregularly distributed sensors positioned on a smooth geometric manifold capture a signal that lives on this manifold. We assume that the sensor measurements are affected by an additive noise. Our objective is to interpolate the signal at a set of points that are not known prior to signal acquisition. Our proposed method locally estimates the parameters of the unknown function on the manifold from the available set of sensor measurements. Once the parameters of the function are computed, the interpolation boils down to computing geodesic distances between the interpolation points and the sensors that perform interpolation.

Distributed processing algorithms in sensor networks should provide efficient and flexible solutions. Numerous distributed algorithms are recently proposed, such as distributed regression [94], [113] or distribution of the operator computations on graphs [6], to name a few. However, studies of distributed interpolation in sensor networks mainly focus on the signals acquired by dynamic sensors [114]. Recent distributed works that study interpolation problems for high dimensional signals are closely related to discrete regularization methods. The family of regularization algorithms introduced by [47] removes the noise and preserves the intrinsic signal features. To solve discrete regularization problems, the set of differential operators commonly deployed in regularization approaches are redefined to hold in a discrete domain in [115], [116]. Graph signal models simplify the high dimensional signal representation and they are commonly used to design fast graph-based regularization algorithms [117], [118]. However, fast and distributed graph-based ap-



**Figure 5.1:** The graph representation of the irregular sensor network on a geometric manifold  $\mathcal{M}$  is defined by a tuple  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ , where sensors  $s_i$ ,  $1 \leq i \leq S$  build the set of vertices, their connections  $e_{i,j} = (s_i, s_j)$ , define the edges  $\mathcal{E}$  whose weights are  $w(s_i, s_j)$ . The set  $\mathcal{N}_{s_i}^{(g)}$  defines a set of sensors that are connected to  $s_i$  ( $w(s_i, s_j) \leq \rho$ ), while the local sensor neighborhood  $\mathcal{N}_{s_i}$  determines the set of nodes whose geodesic distance to  $s_i$  is smaller than  $\sigma$ .

proaches require the knowledge of the interpolation point set in order to build the graph structure and perform interpolation and we consider that the interpolation set is not available prior to data acquisition. Therefore, we propose to estimate the values of a local function around each sensor, so that interpolation can be performed easily at any point on the manifold.

We propose in this chapter a novel distributed interpolation algorithm for sensor networks positioned on geometric manifolds. We assume that the network consists of  $S$  sensors that are irregularly distributed on a known and smooth  $d$ -dimensional geometric manifold  $\mathcal{M}$ . We represent the sensor network by a weighted undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ , where the set of vertices  $\mathcal{V}$  is a set of sensors  $s_i$ ,  $i \in \{1, \dots, S\}$  and the set of edges  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  defines the connections between sensors. We assume that the graph  $\mathcal{G}$  is connected. Then, the edge weights  $w : \mathcal{E} \rightarrow \mathbb{R}^+$  are defined by geodesic distances, where  $w(s_i, s_j) \geq 0$  and  $w(s_i, s_j) = w(s_j, s_i)$ ,  $\forall e_{i,j} = (s_i, s_j) \in \mathcal{E}$ .

The graph representation of the sensor network on a geometric manifold is illustrated in Fig. 5.1. The vector  $\mathbf{m}_i = (m_1(i), \dots, m_d(i))$  for  $i \in \{1, \dots, S\}$  defines the coordinates of the sensor  $s_i$  on a smooth  $d$ -dimensional geometric manifold. A set of sensors in the graph that are connected to the sensor  $s_i$  build its graph neighborhood  $\mathcal{N}_{s_i}^{(g)}$ , where  $\forall s_j \in \mathcal{N}_{s_i}^{(g)}$  holds that  $w(s_i, s_j) \leq \rho$  for a distance value  $\rho$ . Every sensor can communicate its measurement to its graph neighborhood set. In addition, we define a local neighborhood  $\mathcal{N}_{s_i}$  of the sensor  $s_i$  as the set of sensors  $\{s_j\}$  whose geodesic distance to  $s_i$  is  $dist(s_i, s_j) \leq \sigma$ , where  $dist(\cdot)$  defines the geodesic distance function.

The sensors measure a physical phenomenon (e.g., temperature, pressure, etc.) and their measurements are corrupted by an additive noise. The function defined on the vertex (sensor) set of the graph is denoted by  $\tilde{f} \in \mathcal{H}(\mathcal{V})$ , where  $\tilde{f} : \mathcal{V} \rightarrow \mathbb{R}^+$  assigns a real number to every sensor. The noisy sensor measurements  $y_i$ ,  $i \in \{1, \dots, S\}$ , are the samples of the signal  $\tilde{f}$  at sensor locations  $\mathbf{m}_i = (m_1(i), \dots, m_d(i))$ , which reads

$$y_i = \tilde{f}(\mathbf{m}_i) + \epsilon_i, \quad (5.1)$$

where  $\epsilon_i$  represents the additive noise. The sensors exchange measurements with neighboring sensors in adhoc way.



The objective is to locally estimate the unknown signal function at every sensor, in order to be able to interpolate values anywhere on the manifold. Formally, we want to compute the values of an arbitrarily chosen set of interpolation points using the algorithm that minimizes the estimation error and incorporates the inherent manifold characteristics. We pose the interpolation problem as the convex optimization problem that exploits the geometry of the manifold and we solve it analytically. The computations of the analytic solution are performed at sensors using a set of available, locally gathered neighborhood measurements. Now, each sensor  $s_i$  can compute the unknown function parameters given the measurements and locations of neighbor set of sensors  $\mathcal{N}_{s_i}$ . We perform the experiments on spherical signals and show that the proposed interpolation method outperforms the baseline Nearest Neighbor method in terms of the signal quality. Then, we analyze the performance of the iterative interpolation algorithm. In particular, we provide the analysis of data transmission phases in the network. Finally, we analyze the evolution of the bit consumption and the interpolation signal quality over transmission rounds of the iterative interpolation algorithm and show that it converges fast for the small world network connectivity model.

We summarize below the main contributions of this chapter:

- We propose a novel interpolation algorithm that locally computes interpolated values of a function living on a manifold with a kernel regression method, given noisy measurements captured by a set of irregularly distributed sensors on the manifold.
- We propose an iterative algorithm for the interpolation problem, where sensors transmit measurements in an adhoc way to their neighbors with a gossip based protocol and each sensor estimates a local version of the unknown function.
- We apply our framework to spherical manifolds and perform experiments for spherical images, due to difficulties in obtaining the real data. We here assume that each pixel of the spherical image represents a sensor measurement. Our method is shown to outperform the baseline Nearest Neighbor interpolation method in terms of reconstruction quality.
- We provide a probabilistic model for the data dissemination process and show that the distributed interpolation algorithm uses a smaller number of bits to transmit data compared to a broadcasting scenario, where the sensors transmit measurements to all their neighbors.

The rest of this chapter is organized as follows. In Section 5.2, we propose a generic kernel based regression method for sensor networks. We then propose a distributed version of the kernel-based interpolation algorithm in Section 5.4. In addition, we provide a probabilistic model for data dissemination phases. Finally, we present an application to spherical signals in Section 5.3.

## 5.2 Nonparametric weighted kernel regression on geometric manifolds

We propose an interpolation algorithm that incorporates the inherent manifold geometry into the signal interpolation problems for arbitrary points on the manifold. We assume that the unknown manifold signal is smooth and  $n$  times differentiable. Such signals can be locally represented by a Taylor expansion. The interpolation problem becomes the estimation of the Taylor expansion

coefficients at the sensor locations. The parameter estimation of the function from data samples is a problem similar to nonparametric kernel regression methods [86], which however do not directly incorporate the inherent geometry of high-dimensional manifolds.

We first introduce the differential operators for non-Euclidean manifolds. Then, we use them to define the Taylor expansion of a function at the points of the geometric manifold. We then pose the nonparametric kernel regression problem that incorporates the manifold geometry in order to obtain the Taylor expansion coefficients from noisy sensor measurements. This convex optimization problem minimizes the estimation error between the Taylor expansion estimate and the sensor measurements in the sensor neighborhood. Finally, we solve the problem analytically.

First, the differential operators are defined for a smooth manifold  $\mathcal{M}$  by a Riemannian metric  $\mathbf{g}$ . The vectors in the tangent plane of  $\mathcal{M}$  at a point  $\mathbf{m}_i$  represent the directions of computation for derivatives. For the point  $\mathbf{m}_i$  on  $\mathcal{M}$ , the components of the gradient in its neighborhood are given by  $\vec{\nabla}(\cdot) = \mathbf{g}(\frac{\partial(\cdot)}{\partial \mathbf{m}_i})$ . Similarly, the Laplacian is a scalar operator defined as  $\nabla^2(\cdot) = \vec{\nabla} \cdot (\vec{\nabla}(\cdot))$ .

Without loss of generality, we focus now on the second-order Taylor expansion of  $\tilde{f}$  on a manifold. This function at point  $\mathbf{m}_0$  can be represented by its Taylor series expansion around the point  $\mathbf{m}_i$  by

$$\tilde{f}(\mathbf{m}_0) = \tilde{f}(\mathbf{m}_i) + \nabla \tilde{f}(\mathbf{m}_i)^T (\mathbf{m}_0 - \mathbf{m}_i) + \frac{1}{2!} (\mathbf{m}_0 - \mathbf{m}_i)^T (\nabla^2 \tilde{f}(\mathbf{m}_i)) (\mathbf{m}_0 - \mathbf{m}_i) + R_{\tilde{f}} \quad (5.2)$$

where  $\nabla \tilde{f}(\mathbf{m}_i)$  is the gradient of the function  $\tilde{f}$  evaluated at  $\mathbf{m}_i$ ,  $\nabla^2 \tilde{f}(\mathbf{m}_i)$  is the Hessian matrix evaluated at  $\mathbf{m}_i$  and  $R_{\tilde{f}}$  is the residue. In the interpolation context, the point  $\mathbf{m}_0$  represents the interpolation location and  $\mathbf{m}_i$  is the position of a sensor close to  $\mathbf{m}_0$ . We assume that the second order Taylor expansion of the function well approximates the function value in the point, so we consider that it holds  $\tilde{f} \approx \tilde{f} - R_{\tilde{f}}$ . We denote by  $\beta_{(2)}$  the derivatives of up to the second order ( $n = 2$ ) of the function evaluated at  $\mathbf{m}_i$ , so  $\beta_{(2)} = \{\beta_0, \beta_1, \beta_2\} = \{\tilde{f}(\mathbf{m}_i), \nabla \tilde{f}(\mathbf{m}_i), \nabla^2 \tilde{f}(\mathbf{m}_i)\}$ . More detail on derivation of differential operations and Taylor expansion in a Riemannian tensor metric are available in [119]. Throughout this chapter we denote the optimal set of derivatives of the function given by  $\beta_{(2)}$  at  $\mathbf{m}_0$  by  $\beta^*$  for the notation compactness.

Our objective is to compute the best estimate of the Taylor expansion at position  $\mathbf{m}_0$  that well approximates the function in all neighbor sensors. The simplest approach for estimating the values  $\beta^*$  that provide the best fit to the input data represents a Mean Square Error (MSE) minimization problem

$$\beta^* = \arg \min_{\{\beta_{(2)}\}} \sum_{i=1}^q \|y_i - \tilde{f}\|^2, \quad (5.3)$$

where  $q$  is the number of neighboring sensor measurements,  $y_i$  are the noisy sensor measurements defined by Eq. (5.1). The main drawback of such an approach is that it treats all the samples in the same way, without considering a relative position of the interpolation point and measurements.

To overcome the problem of equal impact of sensor measurements, the authors in [99] have proposed to weight signal samples according to their ‘‘importance’’ (e.g., distance to the interpolation point) by a data-driven weight function. The work in [120] has extended this framework to solving interpolation and denoising problems in image processing. Both works introduce a consistent way to weight the relative importance of the sensor measurements by using a

Kernel function. Kernel functions build the class of real valued continuous functions that fulfill conditions of Mercer's theorem ([121], Sec. 4.3). In other words, for all the functions  $f$  that are square-integrable ( $\int |f(\mathbf{m})|^2 d\mathbf{m} < \infty$ ), candidate Kernel functions  $K(\cdot, \cdot)$  should satisfy  $\int \int K(\mathbf{m}_1, \mathbf{m}_2) f(\mathbf{m}_1) f(\mathbf{m}_2) d\mathbf{m}_1 d\mathbf{m}_2 \geq 0$ . Such properties of the weight function in the optimization problem assure that it preserves the core properties (a convex optimization problem remains convex). This type of functions are studied extensively in problems related to statistical machine learning [122].

We propose to use a kernel-based approach to solve our local interpolation problem for sensor manifold data. In particular, each sensor in the network computes the set of parameters that well describes the function in its neighborhood. These parameters are the solutions of the Kernel-based nonparametric regression problem on the geometric manifold.

Without a loss of generality, we assume that coordinates  $\mathbf{m}_0$  of the manifold point where the Taylor expansion is computed is represented by two dimensional manifold coordinates  $(\theta_0, \phi_0)$ . We choose this example for a notation simplicity. The computations given below are easily extended for signals of an arbitrary dimension, as we show later in this section.

We formulate the weighted MSE cost function as  $J(\beta) = \sum_{i=1}^q \|y_i - \tilde{f}(\theta_0, \phi_0)\|^2 K(\theta_i - \theta_0, \phi_i - \phi_0)$  where  $q$  is the number of neighboring sensor measurements of interest and  $K(\cdot, \cdot)$  is the kernel function that depends on the geodesic distances between the sensors  $s_i$  and its neighbor sensors  $\mathcal{N}_{s_i}$ . The first multiplicative term of  $J(\beta)$  enforces data fidelity, while the kernel function weights the sample points according to their geodesic distance to the interpolation point. For compactness, we denote the kernel function by  $K(\theta_i, \phi_i)$ . We assume that the second order Taylor expansion ( $n = 2$ ) provides generally a good approximation of the function and we neglect the residue  $R_{\tilde{f}}$  in our computations, i.e., we use  $\tilde{f}(\theta_0, \phi_0) - R_{\tilde{f}} = \tilde{f}(\theta_0, \phi_0)$ . We then solve the optimization problem

$$\beta^* = \arg \min_{\{\beta_n\}} J(\beta) \quad (5.4)$$

to determine the parameters of the Taylor expansion. Remark that the cost function is a linear function of the unknown values. Then, the second order Taylor expansion of  $\tilde{f}$  at point  $(\theta_0, \phi_0)$  reads

$$\begin{aligned} \tilde{f}(\theta_0, \phi_0) &= \beta_0 + g_1(\theta_i)(\theta_0 - \theta_i)\beta_{11} + g_2(\phi_i)(\phi_0 - \phi_i)\beta_{12} \\ &+ \frac{1}{2}[g_3(\theta_i^2)(\theta_0 - \theta_i)^2\beta_{21} + 2g_4(\theta_i, \phi_i)(\theta_0 - \theta_i)(\phi_0 - \phi_i)\beta_{22} + g_5(\phi_i^2)(\phi_0 - \phi_i)^2\beta_{23}] \end{aligned} \quad (5.5)$$

Here, the set of unknowns  $\beta^* = (\beta_0, \beta_{11}, \beta_{12}, \beta_{21}, \beta_{22}, \beta_{23})$  represents the values from the set

$$\beta^* = \left( \tilde{f}(\theta_i, \phi_i), \frac{\partial \tilde{f}(\theta_i, \phi_i)}{\partial \theta_0}, \frac{\partial \tilde{f}(\theta_i, \phi_i)}{\partial \phi_0}, \frac{\partial^2 \tilde{f}(\theta_i, \phi_i)}{\partial \theta_0^2}, \frac{\partial^2 \tilde{f}(\theta_i, \phi_i)}{\partial \theta_0 \partial \phi_0}, \frac{\partial^2 \tilde{f}(\theta_i, \phi_i)}{\partial \phi_0^2} \right), \quad (5.6)$$

while the set of values  $\{1, g_1(\theta_i), g_2(\phi_i), g_3(\theta_i^2), g_4(\theta_i, \phi_i), g_5(\phi_i^2)\}$  that multiply the elements of the set  $\beta^*$  in Eq. (5.5) is the consequence of coordinate system transformation from the Cartesian to the manifold coordinate system.

We compute the minimum of Eq. (5.4) by analytic approach. The interpolation value at the point  $(\theta_0, \phi_0)$  computed as in Eq. (5.5) is a linear function of the set of unknowns  $\beta^*$ . Therefore the

first order derivatives with respect to the elements in  $\beta^*$  represent the extremum of the function. We obtain the following set of equations

$$\begin{aligned}
\frac{\partial J(\beta)}{\partial \beta_0} &= -2 \left( \sum_i (y_i - \tilde{f}(\theta_0, \phi_0)) \cdot 1 \cdot K(\theta_i, \phi_i) \right) = 0, \\
\frac{\partial J(\beta)}{\partial \beta_{11}} &= -2 \left( \sum_i (y_i - \tilde{f}(\theta_0, \phi_0)) \cdot g_1(\theta_i)(\theta_0 - \theta_i) \cdot K(\theta_i, \phi_i) \right) = 0, \\
\frac{\partial J(\beta)}{\partial \beta_{12}} &= -2 \left( \sum_i (y_i - \tilde{f}(\theta_0, \phi_0)) \cdot g_2(\phi_i)(\phi_0 - \phi_i) \cdot K(\theta_i, \phi_i) \right) = 0, \\
\frac{\partial J(\beta)}{\partial \beta_{21}} &= -2 \left( \sum_i (y_i - \tilde{f}(\theta_0, \phi_0)) \cdot \frac{1}{2} g_3(\theta_i^2)(\theta_0 - \theta_i)^2 \cdot K(\theta_i, \phi_i) \right) = 0, \\
\frac{\partial J(\beta)}{\partial \beta_{22}} &= -2 \left( \sum_i (y_i - \tilde{f}(\theta_0, \phi_0)) \cdot g_4(\theta_i, \phi_i)(\theta_0 - \theta_i)(\phi_0 - \phi_i) \cdot K(\theta_i, \phi_i) \right) = 0, \\
\frac{\partial J(\beta)}{\partial \beta_{23}} &= -2 \left( \sum_i (y_i - \tilde{f}(\theta_0, \phi_0)) \cdot \frac{1}{2} g_5(\phi_i^2)(\phi_0 - \phi_i)^2 \cdot K(\theta_i, \phi_i) \right) = 0.
\end{aligned} \tag{5.7}$$

We rewrite these expressions in the compact matrix form, where the unknowns  $\beta^*$  form the vector  $\hat{\mathbf{b}}$ . We use the notation  $k_i = K(\theta_i, \phi_i)$  and  $\{c_0, c_1, c_2, c_3, c_4, c_5\} = \{1, g_1(\theta_i)(\theta_0 - \theta_i), g_2(\phi_i)(\phi_0 - \phi_i), g_3(\theta_i^2)(\theta_0 - \theta_i)^2, g_4(\theta_i, \phi_i)(\theta_0 - \theta_i)(\phi_0 - \phi_i), g_5(\phi_i^2)(\phi_0 - \phi_i)^2\}$  to obtain

$$\underbrace{\begin{pmatrix} \sum_i y_i c_0 \\ \sum_i y_i k_i c_1 \\ \sum_i y_i k_i c_2 \\ \sum_i y_i k_i c_3 \\ \sum_i y_i k_i c_4 \\ \sum_i y_i k_i c_5 \end{pmatrix}}_{\mathbf{z}} = \underbrace{\begin{pmatrix} \sum_i k_i c_0^2 & \sum_i k_i c_0 c_1 & \sum_i k_i c_0 c_2 & \sum_i k_i c_0 c_3 & \sum_i k_i c_0 c_4 & \sum_i k_i c_0 c_5 \\ \sum_i k_i c_0 c_1 & \sum_i k_i c_1^2 & \sum_i k_i c_1 c_2 & \sum_i k_i c_1 c_3 & \sum_i k_i c_1 c_4 & \sum_i k_i c_1 c_5 \\ \sum_i k_i c_0 c_2 & \sum_i k_i c_1 c_2 & \sum_i k_i c_2^2 & \sum_i k_i c_2 c_3 & \sum_i k_i c_2 c_4 & \sum_i k_i c_2 c_5 \\ \sum_i k_i c_0 c_3 & \sum_i k_i c_1 c_3 & \sum_i k_i c_2 c_3 & \sum_i k_i c_3^2 & \sum_i k_i c_3 c_4 & \sum_i k_i c_3 c_5 \\ \sum_i k_i c_0 c_4 & \sum_i k_i c_1 c_4 & \sum_i k_i c_2 c_4 & \sum_i k_i c_3 c_4 & \sum_i k_i c_4^2 & \sum_i k_i c_4 c_5 \\ \sum_i k_i c_0 c_5 & \sum_i k_i c_1 c_5 & \sum_i k_i c_2 c_5 & \sum_i k_i c_3 c_5 & \sum_i k_i c_4 c_5 & \sum_i k_i c_5^2 \end{pmatrix}}_{\mathbf{P}} \cdot \underbrace{\begin{pmatrix} \beta_0 \\ \beta_{11} \\ \beta_{12} \\ \beta_{21} \\ \beta_{22} \\ \beta_{23} \end{pmatrix}}_{\hat{\mathbf{b}}}. \tag{5.8}$$

The previous equation rewritten in a matrix form reads

$$\mathbf{z} = \mathbf{P} \cdot \hat{\mathbf{b}}, \tag{5.9}$$

where the elements of the vector  $\mathbf{z}$  and the matrix  $\mathbf{P}$  are the functions of sensor measurements and weights, respectively. Finally, the sensor computes the elements of the vector  $\hat{\mathbf{b}}$  by

$$\hat{\mathbf{b}} = \mathbf{P}^{-1} \mathbf{z}, \tag{5.10}$$

where  $\mathbf{P}^{-1}$  is the pseudoinverse of the matrix  $\mathbf{P}$ . In general,  $\mathbf{P}$  is a full rank matrix. We however note that in the certain cases the elements of the matrix  $\mathbf{P}$  are small and they are approximated with zero values in numerical computations.

It is easy to show that  $\hat{\mathbf{b}}$  is the global minimum of the optimization problem given by Eq. (5.4). The complete interpolation algorithm is described in Algorithm 3. The geometric manifold

**Algorithm 3** Local signal interpolation algorithm**Input:**

- The coordinates of the set of neighboring sensors  $\mathbf{m}_i$  and their measurements  $y_j$ .
- Parameters of the kernel function (standard deviation, function choice).
- Position of the interpolation point  $\mathbf{m}_0$ .

**Preprocessing:**

Representation of the coordinates of position  $\mathbf{m}_i$  in terms of coordinates on the geometric manifold.

Computation of first and second order derivatives of the objective function in Eq. (5.4) with respect to the covariant basis vectors.

**Compute:**

Computation of weights for the set of sensors that are neighbors of the interpolation sensor. The weights represent the kernel function values at positions that correspond to geodesic distances between the interpolation sensor and its neighbor sensors.

Computation of the matrix  $\mathbf{P}$  and the vector  $\mathbf{z}$  whose elements, defined in Eq. (5.8), are obtained by optimizing Eq. (5.4).

Computation of vector  $\hat{\mathbf{b}}$  as in Eq. (5.10).

**Output:**

The interpolated value  $\tilde{f}(\mathbf{m}_0)$  at point  $\mathbf{m}_0$ , defined in Eq. (5.5).

is known a priori. In a preprocessing step, we express the coordinates of each sensor in terms of the geometric manifold coordinates and compute the derivatives with respect to its covariant basis vectors. The elements of the matrix  $\mathbf{P}$  and the vector  $\mathbf{z}$  are built as defined by Eq. (5.8). They are dependent on the kernel function values and the sensor measurement values, respectively. Then, the unknown set of parameters  $\beta^*$  that define the Taylor expansion at location  $\mathbf{m}_0$  are computed by matrix inversion, as shown in Eq. (5.10). Each sensor that solves the matrix inversion problem can finally compute the interpolated value of the unknown function at point  $\mathbf{m}_0$ , according to Eq. (5.5).

Finally, we briefly discuss the extension of our problem to higher dimensions. The positions of sensors (graph vertices) can be defined in many problems by three-dimensional vectors. Also, geometric manifold models are usually parametrized by two or more parameters. Interestingly, the above proposed interpolation algorithm holds for data of arbitrary dimension. The dependency between the elements in the set of parameters  $\beta^*$  and the interpolated values indeed remains linear, independently of the signal dimension. By definition, the Taylor expansion in a two-dimensional case is a linear equation of values  $\beta^*$  (see Eq. (5.5)) and the proposition holds. The functions of  $d$  parameters can be represented by a Taylor (multivariate) expansion (def. in Appendix A.5). Such an expansion is however again a linear combination of the derivatives of the function with respect to covariant basis vector of the geometric manifold evaluated at different positions, irrespective of its dimension. Therefore, the optimization problem for high dimensional functions has a solution of the same form as the one given by Eq. (5.10), where the length of the vector  $\hat{\mathbf{b}}$  corresponds to the number of unknown parameters in the estimation.

Finally, we note that the proposed algorithm is naturally distributed, since each sensor can perform interpolation based on local estimation.

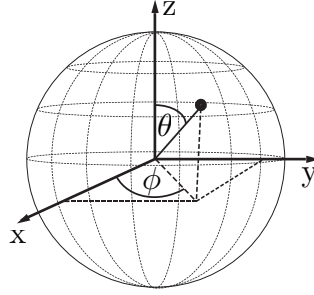


Figure 5.2: Geometry of a 2-d sphere: spherical polar coordinates  $(\theta, \phi)$ .

## 5.3 Application to spherical signals

### 5.3.1 Interpolation on the sphere

In this section, we consider a particular case of data that correspond signals that live on the sphere. The sphere is a pretty common manifold structure in many problems in signal processing and imaging. We assume that the 2-d spherical images correspond to the measurements of the sensor network distributed on the spherical manifold, where each sensor measurement corresponds to the pixel value in the image. We first develop the analytical framework specifically for 2-d spherical signals. Then, we analyze the accuracy of the proposed interpolation algorithm in terms of Mean Squared Error or Peak-Signal-to-Noise (PSNR) for natural images<sup>1</sup>, as well as for different classes of synthetic data. In addition, we provide performance results in terms of Structural Similarity Measure (SSIM) [123] that evaluates the similarity of patches of the signal, rather than point-wise similarity as in PSNR.

Then, under the assumption that sensors in a local neighborhood form a small world network, we analyze the evolution of the data dissemination process as well as its influence on the signal quality. We assume that the positions of neighbor sensors are known and that the number of hops between two neighboring sensors in the graph is at most  $h$  hops. Finally, we provide an analysis of the number of bits transmitted until the convergence of the interpolation algorithm.

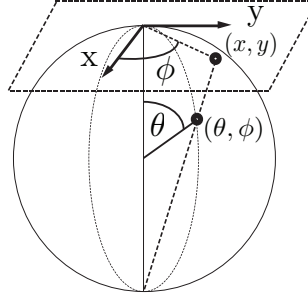
We first formulate the particular expressions used for distributed interpolation on spherical manifolds, that we use later to perform the experiments on natural and synthetic spherical images. A general 2-d sphere represents a unit radius ball. The Cartesian coordinates  $(x, y, z)$  relate to the spherical coordinates by

$$x = r \cos \phi \sin \theta, \quad y = r \sin \phi \sin \theta, \quad z = r \cos \theta,$$

where  $\phi \in [0, 2\pi)$  is the azimuth angle,  $\theta \in [0, \pi]$  represents the zenith angle and  $r > 0$  is the radius.

The 2-d sphere is a smooth manifold with well defined differential operators. The del operator on the sphere reads  $\vec{\nabla} = \hat{r} \frac{\partial}{\partial r} + \frac{\hat{\theta}}{r} \frac{\partial}{\partial \theta} + \frac{\hat{\phi}}{r \sin \theta} \frac{\partial}{\partial \phi}$ . The Laplacian operator is a scalar operator defined by  $\nabla^2(\cdot) = \vec{\nabla} \cdot (\vec{\nabla}(\cdot))$ . We perform simple calculations and rewrite the Laplacian operator in the

<sup>1</sup>Thanks to Ivana Tošić for providing the image.



**Figure 5.3:** The inverse stereographic projection [1]: a point with the Cartesian coordinates  $(x, y)$  in the plane tangent to the North pole is mapped onto the sphere and its coordinates become  $(\theta, \phi)$ .

spherical case as

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{2}{r} \frac{\partial}{\partial r} + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2}{\partial \phi^2} + \frac{\cos \theta}{r^2 \sin \theta} \frac{\partial}{\partial \theta} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}. \quad (5.11)$$

For the unit sphere ( $r = 1$ ), the Laplacian operator reads

$$\nabla^2 = \frac{1}{\sin^2 \theta} \frac{\partial^2}{\partial \phi^2} + \frac{\cos \theta}{\sin \theta} \frac{\partial}{\partial \theta} + \frac{\partial^2}{\partial \theta^2}. \quad (5.12)$$

Then, as the Kernel function we use the symmetric 2-d Gaussian function, since it is a smooth function that has properties of a good Kernel function. The Gaussian function is defined in the Cartesian coordinate system by

$$K(x, y) = \exp \left( - \left( \frac{(x - x_o)^2}{2\sigma_x^2} + \frac{(y - y_o)^2}{2\sigma_y^2} \right) \right),$$

where  $(x_o, y_o)$  are the mean values and  $(\sigma_x, \sigma_y)$  represent the standard deviation values. For spherical manifolds, the kernel function is constructed by projecting this 2-d kernel function to the unit sphere. We use the inverse stereographic projection [1] to map points in the plane that is tangential to the North pole onto a 2-d sphere. Each point  $(x, y)$  in the plane is uniquely mapped to the point  $(\theta, \phi)$  on the sphere where  $(x, y) = (2 \tan \frac{\theta}{2} \cos \phi, 2 \tan \frac{\theta}{2} \sin \phi)$ , as shown in Fig. 5.3. Positioning the Kernel at a particular point on the sphere is easily performed with a single rotation in the SO(3) group [124]. Finally, the set of neighboring points is selected as the points whose geodesic distance to the interpolation point is smaller than the standard deviation values of the kernel. In particular, the geodesic distance between the interpolating sensor  $s_0$  at coordinates  $(\theta_0, \phi_0)$  and the sensor  $s_i$  at  $(\theta_i, \phi_i)$  reads

$$\sigma_g = \operatorname{atan} \left( \frac{\sqrt{[\cos(\pi/2 - \theta_i) \sin(-\phi_i)]^2 + [\cos(\pi/2 - \theta_i) \cos(-\phi_i)]^2}}{\sin(\pi/2 - \theta_i)} \right) \quad (5.13)$$

We note that the above expression holds for angles  $\theta \in \{0, \pi\}$  and  $\phi \in \{-\pi, \pi\}$ , which is different from the standard notation for geodesic distances.

Equipped with the above elements, we now compute the Taylor expansion of signals on the

sphere. We plug the appropriate expressions of the first and second order derivatives of a spherical signal  $\tilde{f}$  in vicinity of  $(\theta_0, \phi_0)$ , where  $r = 1$  into Eq. (5.5) and obtain

$$\begin{aligned} \tilde{f}(\theta_0, \phi_0) &= \tilde{f}(\theta_i, \phi_i) + \frac{\partial \tilde{f}(\theta_i, \phi_i)}{\partial \theta_0}(\theta_0 - \theta_i) + \frac{1}{\sin \theta_i} \frac{\partial \tilde{f}(\theta_i, \phi_i)}{\partial \phi_0}(\phi_0 - \phi_i) \\ &+ \frac{1}{2} \left[ \frac{1}{\sin^2 \theta_i} \frac{\partial^2 \tilde{f}(\theta_i, \phi_i)}{\partial \phi_0^2} (\phi_0 - \phi_i)^2 + \frac{\cos \theta_i}{\sin \theta_i} \frac{\partial \tilde{f}(\theta_i, \phi_i)}{\partial \theta_0} (\theta_0 - \theta_i) + \frac{\partial^2 \tilde{f}(\theta_i, \phi_i)}{\partial \theta_0^2} (\theta_0 - \theta_i)^2 \right]. \end{aligned}$$

By observing the previous equation and Eq. (5.5) given for the general Taylor expansion, we identify the elements of sets  $\beta^*$  and  $\{g_1(\cdot), \dots, g_5(\cdot)\}$ , that are specific for the spherical signals. These sets are given by

$$(\beta_0, \beta_{11}, \beta_{12}, \beta_{21}, \beta_{22}, \beta_{23}) = (\tilde{f}(\theta_i, \phi_i), \frac{\partial \tilde{f}(\theta_i, \phi_i)}{\partial \theta_0}, \frac{\partial \tilde{f}(\theta_i, \phi_i)}{\partial \phi_0}, \frac{\partial^2 \tilde{f}(\theta_i, \phi_i)}{\partial \theta_0^2}, 0, \frac{\partial^2 \tilde{f}(\theta_i, \phi_i)}{\partial \phi_0^2}), \quad (5.14)$$

$$(g_1(\theta_i), g_2(\phi_i), g_3(\theta_i^2), g_4(\theta_i \phi_i), g_5(\phi_i^2)) = (1 + \frac{\cos \theta_i}{\sin \theta_i}, \frac{1}{\sin \theta_i}, 1, 0, \frac{1}{\sin^2 \theta_i}). \quad (5.15)$$

Once these sets are identified the computation of the Taylor expansion coefficients reads

$$\begin{pmatrix} \beta_0 \\ \beta_{11} \\ \beta_{12} \\ \beta_{21} \\ \beta_{23} \end{pmatrix} = \begin{pmatrix} \sum_i k_i & \sum_i k_i c_1 & \sum_i k_i c_2 & \sum_i k_i c_3 & \sum_i k_i c_5 \\ \sum_i k_i c_1 & \sum_i k_i c_1^2 & \sum_i k_i c_1 c_2 & \sum_i k_i c_1 c_3 & \sum_i k_i c_1 c_5 \\ \sum_i k_i c_2 & \sum_i k_i c_1 c_2 & \sum_i k_i c_2^2 & \sum_i k_i c_2 c_3 & \sum_i k_i c_2 c_5 \\ \sum_i k_i c_3 & \sum_i k_i c_1 c_3 & \sum_i k_i c_2 c_3 & \sum_i k_i c_3^2 & \sum_i k_i c_3 c_5 \\ \sum_i k_i c_5 & \sum_i k_i c_1 c_5 & \sum_i k_i c_2 c_5 & \sum_i k_i c_3 c_5 & \sum_i k_i c_5^2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \sum_i y_i \\ \sum_i y_i k_i a \\ \sum_i y_i k_i b \\ \sum_i y_i k_i c \\ \sum_i y_i k_i d \end{pmatrix}. \quad (5.16)$$

Here,  $k_i$  stands for the kernel function value at the position that corresponds to the sensor  $s_i$  and the set of coefficients  $\{c_1, \dots, c_5\}$  read

$$c_1 = (\theta_0 - \theta_i)[1 + \frac{1}{2} \text{atan} \theta_i], c_2 = \frac{\phi_0 - \phi_i}{\sin \theta_i}, c_3 = \frac{1}{2} (\frac{\phi_0 - \phi_i}{\sin \theta_i})^2, c_4 = 0, c_5 = \frac{1}{2} (\theta_0 - \theta_i)^2. \quad (5.17)$$

We note that  $\beta_{22} = 0$  because  $\partial^2 / \partial \theta \partial \phi$  component is not present in the spherical Laplacian operator defined in Eq. (5.12).

### 5.3.2 Experimental interpolation results

We provide interpolation results for several spherical image data sets and discuss them. To recall, we consider that each pixel value represents a scalar sensor measurement, where sensors are distributed on a 2-d sphere. We first show results for an omnidirectional image obtained by a catadioptric camera in noisy and noiseless cases and then present results for other spherical signals with different data density. The two parameters that play important role in the proposed algorithm are the selection of neighboring sensors and the choice of the kernel function. In particular, the standard deviation parameter  $\sigma$  of the Gaussian function represents the geodesic distance on the sphere in Eq. (5.13) and determines the neighborhood of the interpolation point.

We first examine the performance of the kernel regression method for zero, first and second





**Figure 5.4:** Omnidirectional test image.

order Taylor expansions for omnidirectional images. The pixel positions (that correspond to sensor locations) form the equiangular grid. The resolution of the omnidirectional image is  $1024 \times 1024$ , out of which  $1024 \times 326$  pixels have non-zero values (recall that the omnidirectional image has blocks of zero values close to the poles). The unfolded non-zero test image is illustrated in Fig. 5.4. We randomly choose a set of interpolation points from the total set of image pixels. We interpolate the values of the spherical function at the positions of the chosen pixel set using the proposed algorithm. The input data represents the remaining image pixels and the ground truth are the image pixels (sensor readings) at the interpolation positions. We consider the following two datasets with  $16 \cdot 10^3$  (5% of the image pixels) and  $33 \cdot 10^3$  (10% of the image pixels) interpolation points. We average the interpolation results over ten randomly chosen interpolation sets per one dataset.

We compare the performance of the proposed method with a nearest neighbor interpolation method which computes an average of the neighbor sensor values. In the discussions below, we denote the proposed interpolation method by IM and a Nearest Neighbor method by NN.

**(a) Noiseless case** We first consider a noiseless scenario, where the signal values are not corrupted by additive noise. To achieve the best possible performance with NN method, we heuristically optimize the number of neighbor values used in computations. This value is empirically set to value five in our simulations. We provide results for the proposed IM algorithm for zero (IM,  $n = 0$ ) and second order (IM,  $n = 2$ ) Taylor expansion. Note that IM in the case of zero order Taylor expansion corresponds to the manifold-adapted Nearest Neighbor method, where the number of neighbors is determined by the geodesic distance on the manifold.

We study the performance of the interpolation methods in terms of the mean squared error (MSE) values between the ground truth and the interpolated values. The results given in Table 5.1 show that the proposed interpolation method IM achieves better performance than the NN method when the kernel parameter is optimized ( $\sigma \leq 0.7 \cdot 10^{-2}$ ). In the proposed IM method, the selection of the standard deviation of the kernel function is important since it controls the number of the neighboring samples that are considered in the interpolation. Note that the function represented by the zero order Taylor expansion corresponds to the manifold-adapted Nearest Neighbor method. Without optimizing the parameters, it performs similarly to the NN method for a certain range of standard deviation values of the kernel ( $\sigma \leq 0.8 \cdot 10^{-2}$ ).

**(b) Noisy case** We then perform interpolation in the presence of additive Gaussian noise and

MSE ( $\cdot 10^{-4}$ )						
	Standard deviation value $\sigma$ of the kernel for IM					
method	0.02	0.01	0.009	0.008	0.007	0.0069
IM, $n=2$	12	3.584	3.228	2.515	<b>2.362</b>	<b>2.358</b>
IM, $n=0$	13	3.859	3.491	2.662	2.516	2.516
NN	<b>2.435</b>	<b>2.435</b>	<b>2.435</b>	<b>2.435</b>	2.435	2.435

**Table 5.1:** MSE for different values of the standard deviation parameter in the noiseless case. The number of interpolation points is  $16k$ . The parameters of the NN method are optimized to provide minimal MSE (five neighbors).

MSE ( $\cdot 10^{-4}$ )						
	SNR(dB)					
method	100	90	80	70	60	50
IM, $n = 2$	<b>2.358</b>	<b>2.358</b>	<b>2.358</b>	<b>2.358</b>	<b>2.360</b>	<b>2.372</b>
IM, $n = 0$	2.516	2.516	2.515	2.516	2.517	2.530
NN	2.435	2.435	2.435	2.435	2.437	2.457

**Table 5.2:** MSE in the noisy cases with different SNRs. The number of sample points is  $16 \cdot 10^3$  and the standard deviation is fixed to  $\sigma = 0.0069$ . The parameters of the classical NN method are optimized to provide minimal MSE (five neighbors).

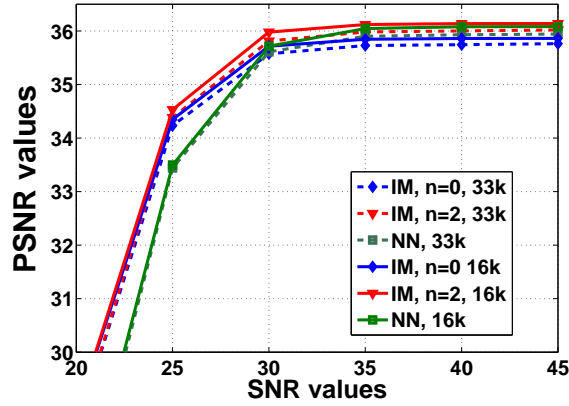
observe the influence of noise on the MSE values for high Signal-to-Noise Ratio (SNR). Interpolation points represent locations of a randomly chosen set of  $16k$  image pixels and the results are shown in Table 5.2. We observe that the proposed methods are pretty robust against noise when their parameters are optimized. Also, the IM method with a second order Taylor expansion achieves a lower MSE than the NN method and IM with a zero order Taylor expansion.

In addition, Fig. 5.5 shows the PSNR values of the reconstructed signals versus the SNR values of the noisy input measurements. We repeat all the experiments ten times and report the average performance results. As we can see, the proposed IM method performs the interpolation task more accurately than the NN method for all the values of SNR for input data. The advantage of IM algorithms is particularly visible in cases where the input signal has low SNR values, where the geometry knowledge compensates for the noisy input data. The IM case with  $n = 0$  corresponds to the manifold-adapted NN method, which can provide an improvement of approximately 2 dB in PSNR in noisy scenarios compared to the NN method.

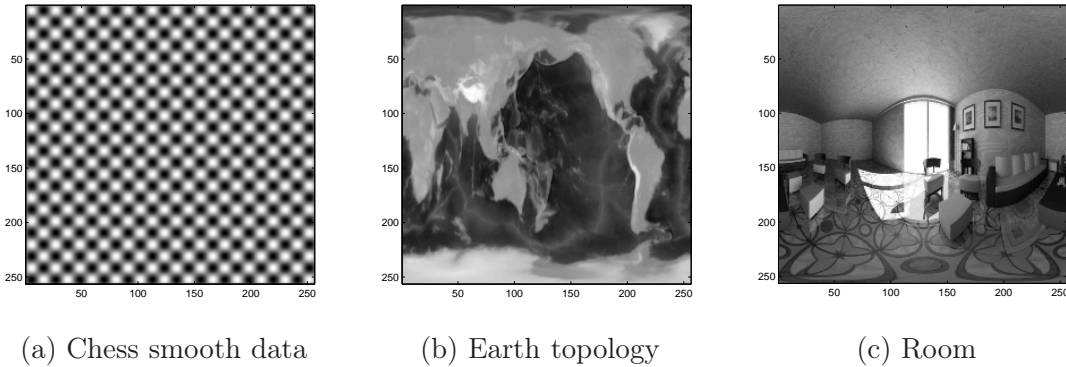
We now perform additional interpolation of noisy spherical signals. Sensors lie on an irregular grid (*Chess smooth* data<sup>2</sup>) or grids that follow a certain structure. Based on the resolution of the original image compared to the set of points in the interpolation set, we distinguish two types of input data: (i) interpolation points that are selected randomly from the equiangular grid (*Earth*<sup>3</sup> dataset) and (ii) irregular interpolation points that are obtained from the grid of a large resolution

<sup>2</sup>Thanks to Luigi Bagnato for providing the original image.

<sup>3</sup>Available from the National Geophysical Data Center, NOAA US Department of Commerce under data announcement 88-MGG-02.



**Figure 5.5:** PSNR results for sets of  $16 \cdot 10^3$  and  $33 \cdot 10^3$  interpolation points in presence of additive Gaussian noise,  $\sigma = 0.07$ .



**Figure 5.6:** Original spherical images (unwrapped) are down-sampled to dimension  $256 \times 256$  and their intensity values are rescaled to the range  $[0,1]$ .

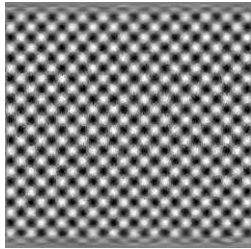
image. Pixel locations in the high resolution signal do not match the positions of the interpolation point set (two sets *Room regular* and *Room irregular*<sup>4</sup>). For the Earth dataset, the value of standard deviation parameter has to be big enough to ensure that all interpolation points have at least one neighbor.

The last set (*Room irregular*) contains interpolation points whose distribution guarantees good spatial coverage of the 2-d sphere. In more details, an equal number of interpolation points within two spherical patches of the same geodesic radius around poles and the equator ensures that the signal densities are close to equal for the two patches. This is achieved by sampling randomly the number of points proportional to the latitude angle  $\theta$  (see Fig. 5.2). The ground truth spherical images are illustrated in Fig. 5.6.

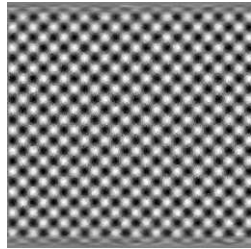
<sup>4</sup>Thanks to Zafer Arican for providing the original image.

**Table 5.3:** The structural similarity measure (SSIM) values for interpolation of noisy spherical signals. Kernel and NN parameters are optimized for each of the datasets.

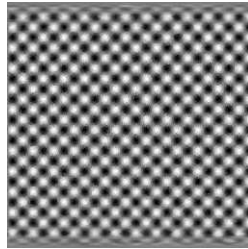
SSIM ( $\times 10^{-2}$ )								
SNR (dB)	10	15	20	25	10	15	20	25
	<i>Earth dataset</i>				<i>Chess smooth dataset</i>			
IM	<b>99.88</b>	<b>99.96</b>	<b>99.99</b>	<b>100</b>	<b>99.57</b>	<b>99.97</b>	<b>99.97</b>	<b>99.98</b>
NN	99.79	99.93	99.97	99.99	95.67	95.68	95.67	95.68
	<i>Room regular dataset</i>				<i>Room irregular dataset</i>			
IM	<b>99.98</b>	<b>99.99</b>	<b>99.99</b>	<b>99.99</b>	<b>99.97</b>	<b>99.99</b>	<b>99.99</b>	<b>99.99</b>
NN	99.93	99.97	99.98	99.99	99.50	99.75	99.91	99.96



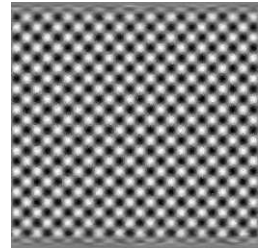
(a) PSNR=16.77 dB



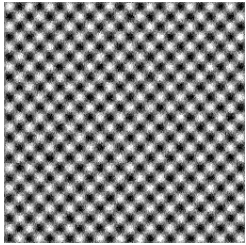
(b) PSNR=18.65 dB



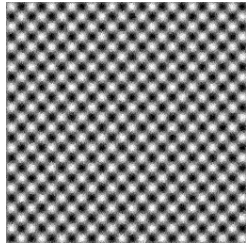
(c) PSNR=20.34 dB



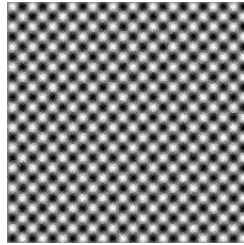
(d) PSNR=21.05 dB



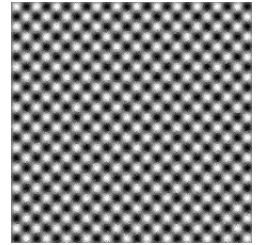
(a) PSNR=20.79 dB



(b) PSNR=23.91 dB



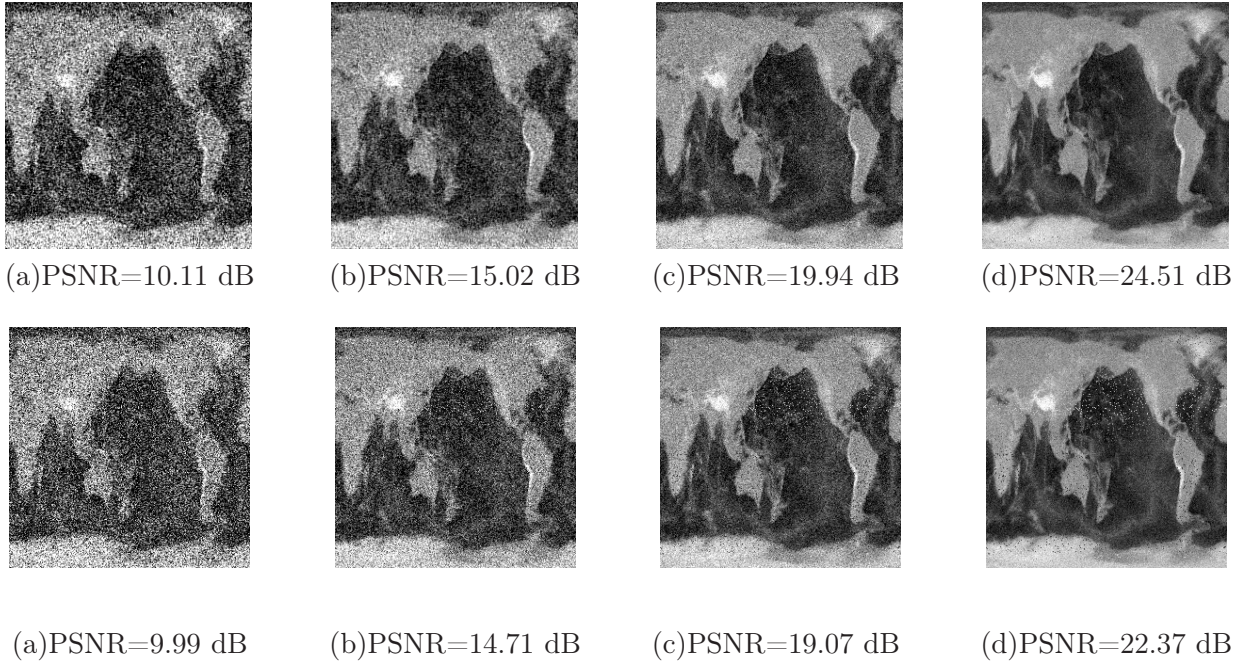
(c) PSNR=25.62 dB



(d) PSNR=26.34 dB

**Figure 5.7:** Results for the Chessboard smooth image data. Top row: The proposed IM method, parameter  $\sigma = 0.03$ . Bottom row: The *Nearest Neighbor* method: the parameter used in simulation is in the range  $\{1, 2, 3\}$ , optimized for each interpolation point. (a) SNR=10 dB, (b) SNR=15 dB, (c) SNR=20 dB, (d) SNR=25 dB.

We perform simulations and measure the interpolation performance in terms of Peak-Signal-To-Noise Ratio (PSNR). However, it is a well known fact that in certain cases higher PSNR values do not guarantee higher reconstruction qualities of the corresponding signals [123]. Therefore, we also consider in this work a quality measure that examines the similarity of two signals based on the signal patches (groups of sample pixels) rather than examining pixel-wise similarities, as in PSNR.



**Figure 5.8:** Results for the Earth Topo image data with 5% of missing data. Top row: Proposed method (IM,  $n = 2$ ). The value  $\sigma$  used in (a) and (b) is  $\sigma = 0.03$  and for (c) and (d)  $\sigma = 0.02$ . Bottom row: The *Nearest Neighbor* method uses 10 neighbor values for interpolation. (a) SNR=10 dB, (b) SNR=15 dB, (c) SNR=20 dB, (d) SNR=25 dB.

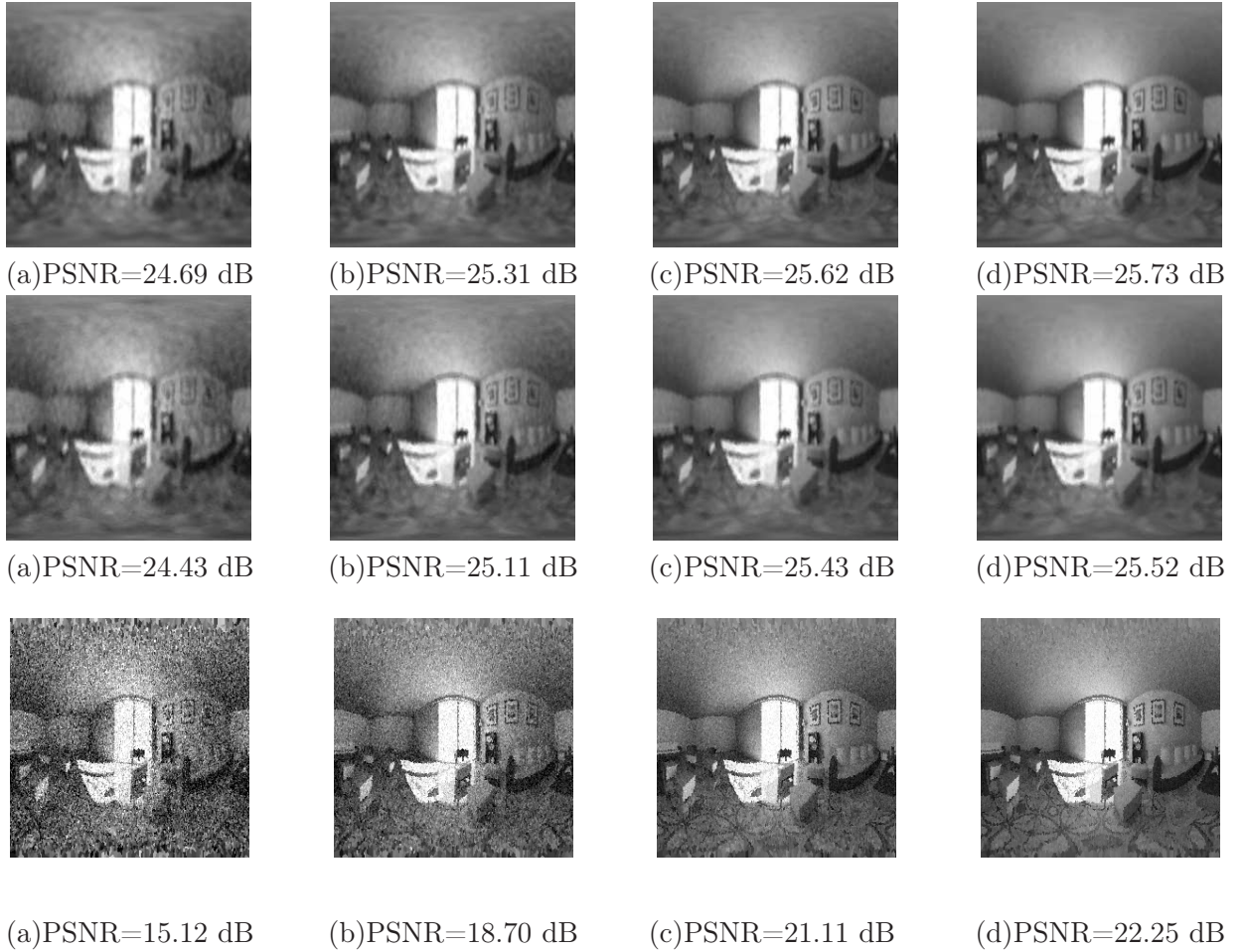
The Structural-SIMilarity-Measure (SSIM) is defined as the average similarity of the total set of pairs of signal patches. For a pair of patches  $M_1$  and  $M_2$ , SSIM reads

$$s(M_1, M_2) = \left( \frac{2\mu_{M_1}\mu_{M_2} + c_1}{\mu_{M_1}^2 + \mu_{M_2}^2 + c_1} \right) \cdot \left( \frac{2\sigma_{M_1M_2} + c_2}{\sigma_{M_1}^2 + \sigma_{M_2}^2 + c_2} \right) \quad (5.18)$$

where  $(\mu_{M_1}, \mu_{M_2})$  are the means of two patches  $M_1$  and  $M_2$ ,  $(\sigma_{M_1}, \sigma_{M_2})$  are their standard deviations, and  $\sigma_{M_1M_2}$  is the cross-correlation. Its values lie in the range  $-1 \leq s(M_1, M_2) \leq 1$ , where the value  $s(M_1, M_2) = 1$  means that  $M_1$  and  $M_2$  are identical sets.

We compute PSNR values of the interpolated spherical signals for the two competing schemes (IM and NN). Interpolated spherical signals are illustrated in Figs. 5.7-5.10 for different levels of Signal-to-Noise ratio values,  $SNR \in \{10, 15, 20, 25\} dB$ . Also, we provide in Tab. 5.3 the SSIM values for all the datasets. We now discuss these results for each of the test sets.

**(a) Chess-smooth data set** The *Chess smooth* set does not have the sharp transitions between the white and black patches, but it consists of regular smooth variations of intensities that build the chess pattern (see Fig. 5.6 (a)). The results for the *Chess smooth* irregular dataset are given in Fig. 5.7. As we can see, the NN method presents higher PSNR values. However, visual quality in the transition regions between white and black chess fields is not satisfactory, which is demonstrated by its lower SSIM values (Tab. 5.3) compared to the proposed method. In

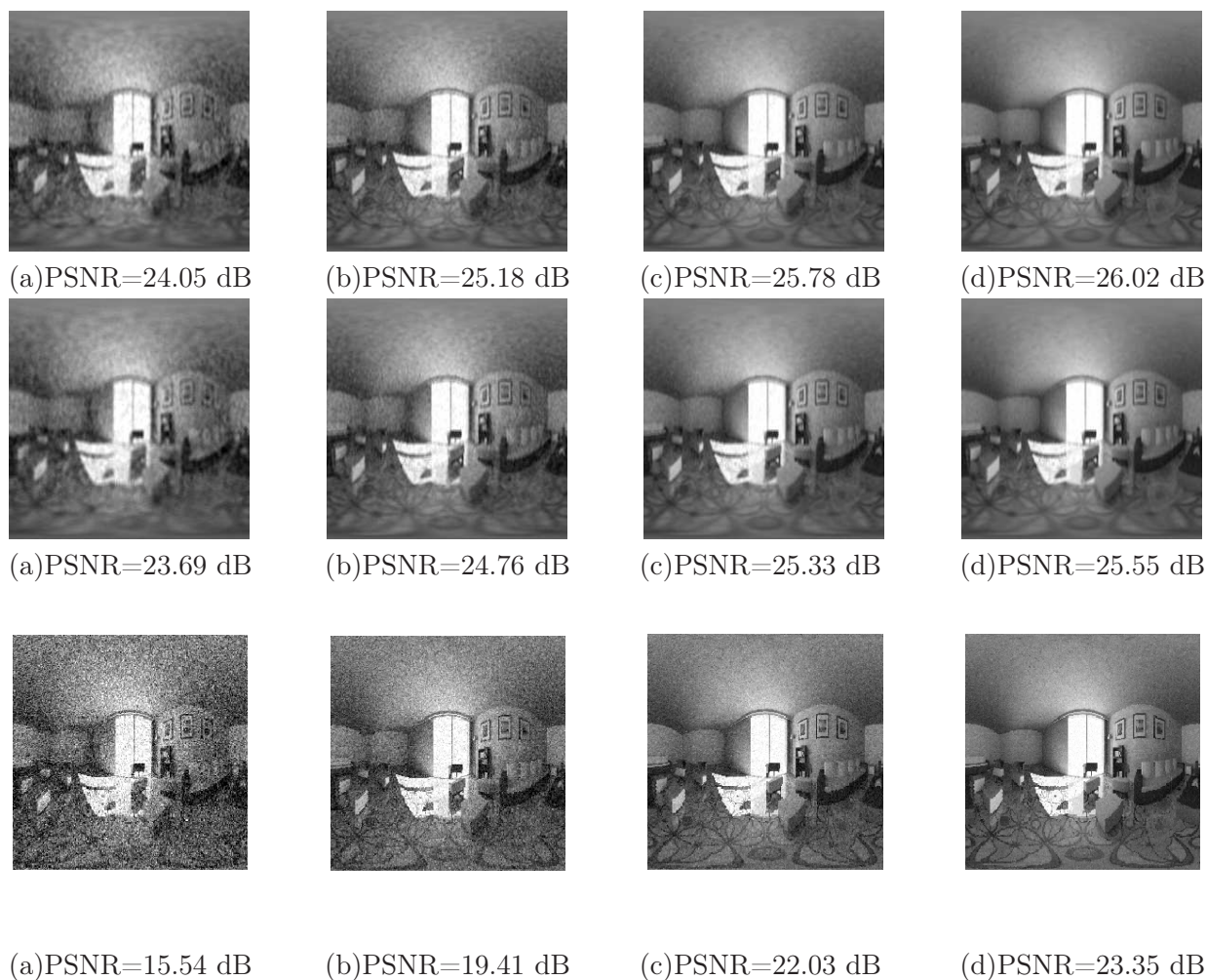


**Figure 5.9:** Results for the irregular Room image data set. Top row: *Proposed method (IM,  $n = 2$ ),  $\sigma = 0.07$* . Middle row: *Proposed method (IM,  $n = 0$ ),  $\sigma = 0.01$* . Bottom row: The *Nearest Neighbor* method uses optimized parameter value. (a) SNR=10 dB, (b) SNR=15 dB, (c) SNR=20 dB, (d) SNR=25 dB.

comparison, the IM method gives results that are visually more pleasant.

**(b) Earth data set** For the *Earth* data set, the union of interpolation points and input samples forms an equiangular spherical grid. In such a scenario, interpolation points form “patches” of missing data and the minimal standard deviation of a kernel of the IM method is fixed by the diameter of such a patch. Consequently, the quality of the local information gathered by the IM and NN methods for the interpolation changes over regions in the image. The proposed IM method outperforms the baseline NN method in such settings, as demonstrated in Fig. 5.8 for PSNR and Tab. 5.3 for SSIM. Also, the results are visually more pleasant.

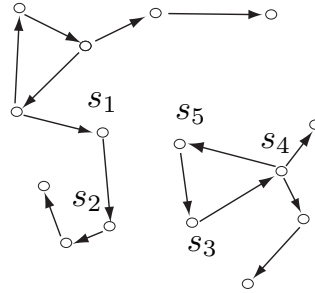
**(c) Room irregular data set** For the *Room irregular* data set data samples are taken irregularly but provide a good spatial coverage. Therefore, this set does not impose strict constraints to the choice of standard deviation of the kernel for the IM method. As we can see in Fig. 5.9,



**Figure 5.10:** Results for the regular Room image data set. Top row: *Proposed method* ( $IM, n = 2$ ). Middle row: *Proposed method* ( $IM, n = 0$ ). Bottom: The *Nearest Neighbor* method uses optimized parameters. (a) SNR=10 dB, (b) SNR=15 dB, (c) SNR=20 dB, (d) SNR=25 dB.

the proposed IM method outperforms the baseline NN method. In addition to the results for the proposed method with second order Taylor expansion, we also present results for the case when lower order Taylor expansion is considered ( $IM, n = 0$ ). As explained before, IM in the latter setup corresponds to the Nearest Neighbor method adapted to the manifold. Therefore, results shown in Fig. 5.9 demonstrate the benefits of incorporating manifold knowledge into the interpolation algorithms, in particular in the noisy setups. Overall, IM with ( $n = 2$ ) achieves the best visual quality of the reconstructed image, as well as the best PSNR and SSIM (Tab. 5.3) results.

**(d) Room regular data set** The *Room Regular* sample set is formed by down-sampling a higher resolution image. Due to the geometry of the sphere, this leads to dense sampling around poles and a less dense sampling in equatorial areas. In this case, the proposed method achieves



**Figure 5.11:** Schematic illustration of the measurement dissemination process that is based on a gossip algorithm with pull protocol. Every sensor requests one measurement from one of its neighbor, chosen uniformly at random.

better performance in terms of SSIM (Tab. 5.3). The visual quality and the PSNR results are illustrated in Fig. 5.10. The results for the IM with  $n = 0$  show again that exploiting manifold knowledge in the interpolation algorithms is helpful, in particular in noisy environments. More precise estimation and weighting of the relative importance of sensor measurements by the kernel function further improve the performance of the interpolation algorithm, as shown in Fig. 5.10 for the proposed method with IM with  $n = 2$ .

## 5.4 Iterative nonparametric weighted kernel regression on geometric manifolds

In the previous section we assumed that the set of  $q$  neighbor measurements is available at the interpolation sensor. However, this is not usually the case in realistic scenarios. Therefore, we propose in this section an iterative interpolation algorithm and provide a fully distributed interpolation algorithm, where sensors exchange measurements with their neighbors using a gossip dissemination algorithm and perform in-network computations that iteratively refines the interpolated values.

### 5.4.1 Interpolation algorithm

Our goal is to build an interpolation algorithm at any sensor  $s_i$  in the network, where the sensors estimate the parameters of the function required for interpolation from the set of neighboring measurements  $\mathcal{N}_{s_i}$  that are currently available. Initially, sensors do not have knowledge about the measurements of neighboring sensors and the measurements are distributed in the network by a gossip dissemination algorithm. When sensors receive measurements from neighbors, they update the interpolated value.

Communication between sensors is performed at each system round, indexed by  $t$ . The measurement dissemination initiates at the transmission round  $t = 1$ . At each round, the sensors randomly choose one of their neighbors and request its measurement, as illustrated in Figure 5.4. This type of mechanism for message transmission is called the gossip pull algorithm [17]. As rounds proceed, the sensors gather more measurements and obtain a better estimate of the interpolated values.



We explain in more details the distribution of the operations at sensors. Let  $q(t)$  be the number of measurements available at a sensor at round  $t$  and  $y_t$  be the measurement collected in the round  $t$ . Then, we rewrite the optimization problem in Eq. (5.4) in terms of the data available at rounds  $(t-1)$  and  $t$ , which respectively represent the previous and the current transmission round

$$\beta^* = \min_{\{\beta_n\}} \sum_{i=1}^{q(t-1)} \|y_i - \tilde{f}(\theta_0, \phi_0)\|^2 K(\theta_i, \phi_i) + \|y_t - \tilde{f}(\theta_0, \phi_0)\|^2 K(\theta_t, \phi_t), \quad (5.19)$$

where  $\tilde{f}(\theta_0, \phi_0)$  is defined in Eq. (5.5) by the Taylor expansion and  $K(\theta_i, \phi_i)$  are the weights computed by the kernel function. As before, the solution is obtained analytically, according to Eqs. (5.7 - 5.8). At round  $(t-1)$ , where  $(t > 1)$ , the sensors store the values of  $\mathbf{z}(t-1)$  and  $\mathbf{P}(t-1)$  from the previous round. The computed matrices are equivalent to those computed by Eq. (5.8), using the current available set of  $q(t-1)$  measurements at sensors. We use the notation  $k_i = K(\theta_i, \phi_i)$  and

$$\begin{aligned} & \{c_0, c_1, c_2, c_3, c_4, c_5\} \\ & = \{1, g_1(\theta_i)(\theta_0 - \theta_i), g_2(\phi_i)(\phi_0 - \phi_i), g_3(\theta_i^2)(\theta_0 - \theta_i)^2, g_4(\theta_i, \phi_i)(\theta_0 - \theta_i)(\phi_0 - \phi_i), g_5(\phi_i^2)(\phi_0 - \phi_i)^2\}. \end{aligned} \quad (5.20)$$

We observe the inherent additive structure of matrices and rewrite it by

$$\mathbf{z}(t-1) + \underbrace{\begin{pmatrix} y_j c_0 \\ y_j k_j c_1 \\ y_j k_j c_2 \\ y_j k_j c_3 \\ y_j k_j c_4 \\ y_j k_j c_5 \end{pmatrix}}_{\mathbf{z}_j} = \left( \mathbf{P}(t-1) + \underbrace{\begin{pmatrix} k_j c_0^2 & k_j c_0 c_1 & k_j c_0 c_2 & k_j c_0 c_3 & k_j c_0 c_4 & k_j c_0 c_5 \\ k_j c_0 c_1 & k_j c_1^2 & k_j c_1 c_2 & k_j c_1 c_3 & k_j c_1 c_4 & k_j c_1 c_5 \\ k_j c_0 c_2 & k_j c_1 c_2 & k_j c_2^2 & k_j c_2 c_3 & k_j c_2 c_4 & k_j c_2 c_5 \\ k_j c_0 c_3 & k_j c_1 c_3 & k_j c_2 c_3 & k_j c_3^2 & k_j c_3 c_4 & k_j c_3 c_5 \\ k_j c_0 c_4 & k_j c_1 c_4 & k_j c_2 c_4 & k_j c_3 c_4 & k_j c_4^2 & k_j c_4 c_5 \\ k_j c_0 c_5 & k_j c_1 c_5 & k_j c_2 c_5 & k_j c_3 c_5 & k_j c_4 c_5 & k_j c_5^2 \end{pmatrix}}_{\mathbf{P}_j} \right) \cdot \underbrace{\begin{pmatrix} \beta_0(t) \\ \beta_{11}(t) \\ \beta_{12}(t) \\ \beta_{21}(t) \\ \beta_{22}(t) \\ \beta_{23}(t) \end{pmatrix}}_{\hat{\mathbf{b}}(t)}. \quad (5.21)$$

The elements of matrices  $\mathbf{P}(t-1)$  and  $\mathbf{z}(t-1)$  are respectively functions of the values of the kernel function and measurements, for the sensors whose messages have been received up to the communication round  $(t-1)$ . We denote the matrices that represent the contributions of the measurement  $y_j$  received at round  $t$ , by  $\mathbf{P}_j$  and  $\mathbf{z}_j$ . Then the values of Taylor coefficients  $\hat{\mathbf{b}}(t)$  at round  $t$  are computed by

$$\hat{\mathbf{b}}(t) = (\mathbf{P}(t-1) + \mathbf{P}_j)^{-1} \cdot (\mathbf{z}(t-1) + \mathbf{z}_j). \quad (5.22)$$

We assume that every sensor receives all the data from its neighbors in  $T$  rounds. It is then straightforward to show that as the number of rounds approaches  $T$ , the iterative algorithm converges to the solution of Algorithm 3, that is given for the case when all the data from neighboring sensors are available at once.

The iterative distributed interpolation algorithm is summarized in Algorithm 4. It proceeds similarly to Algorithm 3. The elements of the matrices  $\mathbf{P}(t)$  and  $\mathbf{Z}(t)$  are computed using the expressions obtained for the solution of the optimization problem in Eq. (5.22). At each round  $t$ , every sensor requests a message (denoted as a caller sensor) from a randomly chosen neighboring

**Algorithm 4** Iterative local signal interpolation algorithm**Input:**

- The coordinates of the set of neighboring sensors  $\mathbf{m}_i$ .
- Parameters of the kernel function (standard deviation, function choice).
- Position of the interpolation point  $m_0$ .

**Preprocessing:**

- Representation of the coordinates of position  $\mathbf{m}_i$  in terms of coordinates on the geometric manifold.
- Computation and storage of the sets  $\{k_i\}$  and  $\{c_1, \dots, c_5\}$  which are given by  $k_i = K(\mathbf{m}_i)$  and Eq. (5.20), respectively.
- Computation of the first and second order derivatives of the objective function in Eq. (5.4) with respect to covariant basis vectors.

**Initialization:**

- Set the counter  $t = 0$ .

**while**  $t \leq T$  **do**

- Message transmission from a randomly chosen neighbor.
- Computation of  $\mathbf{P}_j$  and the vector  $\mathbf{z}_j$ .
- Update of the sensor estimate  $\hat{\mathbf{b}}$  as in Eq. (5.22).
- Update of  $\mathbf{P}(t)$  and  $\mathbf{z}(t)$  that represent the weight and measurement contributions of the received measurement.
- $t \leftarrow t + 1$ .

**end while****Output:**

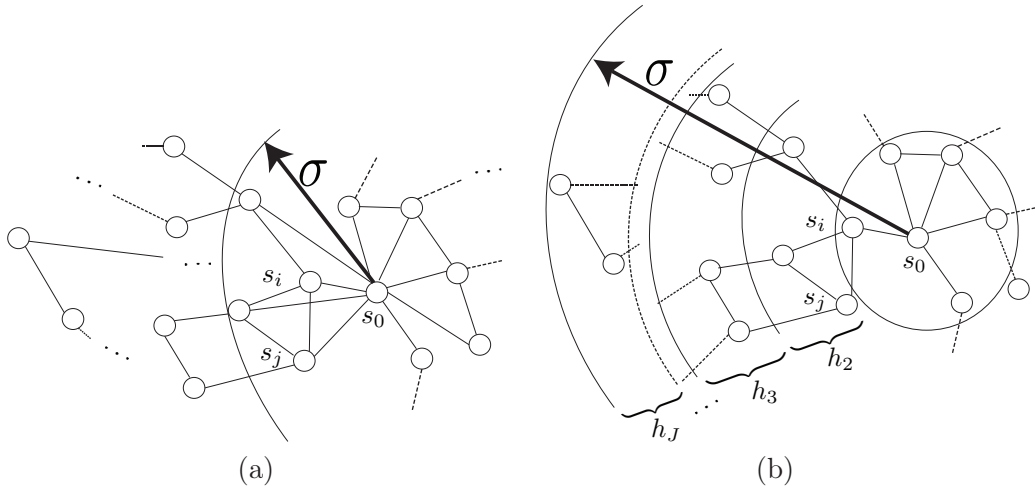
- The interpolated value  $\tilde{f}(\mathbf{m}_0)$  at point  $\mathbf{m}_0$ , defined in Eq. (5.5).

sensor (a sender sensor). The sender forwards the novel message to the caller in case it collected such a message in previous rounds, as explained later on in details. After receiving the measurements at the first round, the sensors compute  $\mathbf{P}(1)$  and  $\mathbf{Z}(1)$ . At each subsequent round, the sensors update the values of the stored matrices in Eq. (5.21) with the values  $\mathbf{P}_j$  and  $\mathbf{z}_j$ . Depending on application settings and computational constraints at sensors, the parameters  $\beta^*$  might be computed periodically and not at every communication round. After  $T$  transmission rounds, we assume that all the sensors received all messages from their neighbors with high probability. Finally, the interpolation values at arbitrary positions within a network coverage range are computed by plugging the coordinates of the locations of interpolated values into the Taylor expansion term in Eq. (5.5), whose parameters are the elements of  $\hat{\mathbf{b}}(T)$  in the final round.

### 5.4.2 Analysis of data transmission phases

The convergence of the proposed interpolation algorithm is reached when the total set of measurements are available to the sensor which performs the interpolation. We analyze now the data transmission in sensor networks and we observe that it can be separated into several phases. We use the general probabilistic framework to model the data dissemination phases.

Without loss of generality, we study the setup where a single interpolation sensor  $s_0$  collects



**Figure 5.12:** Schematic illustration of the two main dissemination cases. The standard deviation of a kernel defines the interpolation data set as the set of sensors  $\mathcal{N}_{s_0}$  that are positioned at a distance smaller than  $\sigma$  from the interpolation sensor  $s_0$ . (a) Sensors within a distance  $\sigma$  can directly communicate with the interpolation sensor. (b) The range  $\sigma$  is larger than the maximal sensor communication range, so that adhoc transmission of measurements is adopted. To simplify the measurement dissemination analysis, the sensors are divided in sets  $\mathcal{H}_j$  of cardinality  $h_j$ ,  $j \in \{1, \dots, J\}$ , based on the number of the transmission rounds  $j$  required for the sensor measurement to reach the interpolation sensor, using the shortest transmission path.

neighbor sensor measurements. We denote the sensor that requests the measurements as the *caller* sensor, while the sensor chosen to transmit its measurement is denoted as the *sender*. The sensors follow a gossip dissemination algorithm with pull protocol (see Fig. 5.4), and data dissemination is performed over transmission rounds, indexed by  $t$ . At each round, gossip is achieved by all sensors. A caller  $s_i$  requests from a randomly chosen sensor in neighborhood to send a novel measurement to the caller, when available.

Recall that  $\mathcal{N}_{s_0}$  defines the sensors whose values are used for interpolation, while  $\mathcal{N}_{s_0}^{(g)}$  defines a set of sensors linked by edges with  $s_0$ . The sensor  $s_0$  can receive and send data only to the direct link sensors in  $\mathcal{N}_{s_0}^{(g)}$ . We assume that the interpolation sensor  $s_0$  knows the identifiers of all the sensors whose measurements are used to perform the proposed interpolation algorithm. In total, there are  $q$  such sensors. Then, based on the values of the standard deviation of the kernel function  $\sigma$ , and the communication range  $\rho$ , dependent on the geodesic distance between sensors linked by an edge  $w$ , we distinguish two data dissemination cases:

The first case is illustrated in Fig. 5.12 (a). In this case, all the sensors used in the interpolation algorithm,  $\mathcal{N}_{s_i}$  are positioned within the communication range  $\rho$  of  $s_0$ . Therefore, within  $t = d_{s_0}$  rounds, where  $d_{s_0}$  is the degree of  $s_0$ , the full set of neighboring measurements reaches the interpolation sensor with probability one.

The second case is illustrated in Fig. 5.12 (b). In this case, the number of data dissemination rounds necessary for the full set of neighbor measurements to reach  $s_0$  with high probability is a function of several parameters. In particular, the number of rounds is the function of the connec-

tivity, the standard deviation of a kernel and the number of sensors in  $\mathcal{H}_j$ ,  $j \in \{1, \dots, J\}$ . The set  $\mathcal{H}_j$  is the collection of sensors whose measurements reach  $s_0$  in  $j$  transmission rounds, when the transmission is performed over the shortest transmission path between the sensors  $\mathcal{H}_j$  and  $s_0$ . The number of sensors in  $\mathcal{H}_j$  is equal to  $h_j = |\mathcal{H}_j|$ . Note that  $\mathcal{N}_{s_0}^{(g)} = \mathcal{H}_1$ . We assume that the connectivity matrix and the values  $J$  and  $h_j$ ,  $j \in \{1, \dots, J\}$  are known. We denote the total degree of all the sensors in  $\mathcal{H}_j$  by  $d_{h_j}$ . Amongst the connections  $d_{h_j}$ , we distinguish the outgoing connections  $d_{o,h_j}$ , the incoming connections  $d_{i,h_j}$  and the connections  $d_{b,h_j}$  between sensors from the set  $\mathcal{H}_j$ , where  $d_{h_j} = d_{o,h_j} + d_{i,h_j} + d_{b,h_j}$ . The degree  $d_{s_0}$  of the interpolation sensor is equal to the total number of its communication links with the sensors in  $\mathcal{H}_1$ . The total set of neighboring sensors  $\mathcal{N}_{s_0}$  is equal to  $q = \sum_{j=1}^J h_j$ . In the rest of this section, we focus on the dissemination analysis for this setup.

We now model dissemination phases for the setup illustrated in Fig. 5.12 (b). Let  $r$  denote the number of different sensor measurements collected by the sensor  $s_0$ . We introduce the random variable  $T_r$  whose value is equal to the number of transmission rounds which ensures that a novel message reaches the sensor  $s_0$  with high probability, where  $s_0$  previously collected  $(r-1)$  messages. We analyze now the evolution of  $T_r$  when the data dissemination phase corresponds to a worst case scenario. We distinguish the following phases in the data dissemination process:

- (a) The first dissemination phase corresponds to the phase when novel measurements reach the interpolation sensor  $s_0$  at every round with high probability. The sensor  $s_0$  gathers measurements only from  $\mathcal{H}_1$ . In the first round, all the measurements in  $\mathcal{H}_1$  are novel for  $s_0$ . This phase ends when the number of sensors with novel messages is less than half of the number of sensors in  $\mathcal{H}_1$ . A more detailed description is given below in this section.

In this phase, the random variable  $T_r$  is bounded by the geometric random variable  $Y_r$  in a stochastic sense, which we denote by  $T_r \prec_{st} Y_r$ . Here,  $T_r \prec_{st} Y_r$  denotes that for any  $r \in \mathbb{Z}$ , the probability of  $T_r$  exceeding  $r$  is smaller than the probability that  $Y_r$  exceeds  $r$ .

- (b) The second phase represents the phase of the accumulation of novel measurements at nodes in the set  $\mathcal{H}_1$  that arrive from more distant sensors  $\mathcal{H}_j$ ,  $j > 1$ . At the beginning of this phase, the novel messages reach the sensor that performs interpolation with low probability. As the rounds evolve, measurements of more distant sensors reach sensors in  $\mathcal{H}_1$ , so the probability that novel message arrives to  $s_0$  increases. Therefore, the random variable  $T_r$  is the function of the current number of measurements known to  $s_0$ , as well as of the current round. Such random variables follow a discrete phase type distribution. This phase finishes when the number of novel accumulated messages is bigger than half of the number of sensors in  $\mathcal{H}_1$ .

Then, the random variable  $T_r \prec_{st} Y_r$ , where  $Y_r$  has a discrete phase type distribution, as shown in Fig. 5.4.2. We note that in setups with a dense local connectivity and small value  $J$  (see Fig. 5.12), the first two phases (a) and (b) are in general sufficient to transmit all the interpolation data when the sensors are well connected.

- (c) The third dissemination phase itself consists potentially from three phases: (i) novel measurements at first reach the interpolation sensor  $s_0$  with high probability; in (ii) this number decreases, but still, novel messages arrive after several rounds; finally in (iii) a set of a remaining few measurements arrives only after a larger number of rounds. Such random variables

are respectively modeled by (i) the geometric probability whose parameter is a probability  $p_g > 0.5$  (almost at every round, novel measurements arrives to  $s_0$ ), (ii) binomial probability (every few rounds, novel measurements arrive) and (iii) geometric probability whose success probability is low,  $p_g < 0.5$ .

We assume that  $J$  is not large in our case (it is dependent on the  $\sigma$ ) and we therefore approximate this phase with the phase (i).

Then, in the third dissemination phase, the random variable  $T_r \prec_{st} Y_r$ , where  $Y_r$  follows a geometric distribution. We can not analyze this phase in details, since it is highly dependent on the connectivity and previous phases.

The probability of the random variable  $T_r$  is a function of the number of elements in the set of *non-helper sensors*, which we denote by  $\mathcal{U}_0(t)$ . Non-helper sensors are direct neighbors of  $s_0$  that can not transmit a novel measurement to  $s_0$  at the round  $t$ . Let  $\mathcal{S}_{s_0}(t)$  denote the set formed by the measurements available at  $s_0$  at the beginning of the transmission round  $t$ . The dimension of the set  $r = \dim(\mathcal{S}_{s_0}(t))$  is equal to the number of distinct measurements collected by  $s_0$ . Then, the set of non-helper sensors is defined by  $\mathcal{U}_0(t) = \{s_i \in \mathcal{N}_{s_0} | \mathcal{S}_{s_i}(t) \subseteq \mathcal{S}_{s_0}(t)\}$  and their number is equal to  $|\mathcal{U}_0(t)|$ . The non-helper sensors form the subset of the sensors in  $\mathcal{H}_1$ ,  $\mathcal{U}_0(t) \subset \mathcal{H}_1$ . We define helper sensors as all the sensors in  $\mathcal{H}_1$  that are not in  $\mathcal{U}_0(t)$ . The probability that  $s_0$  contacts a non-helper sensor at the round  $t$  is defined by

$$p(|\mathcal{U}_0(t)|) = |\mathcal{U}_0(t)|/h_1, \quad (5.23)$$

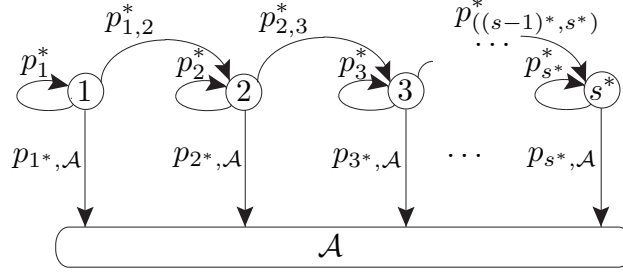
where  $h_1$  is the number of sensors with direct links with  $s_0$ .

#### (a) The first dissemination phase

In the first dissemination round, the sensors acquire the signal. All the measurements in  $\mathcal{H}_1$  are novel for the interpolation sensor, therefore,  $s_0$  for sure collects a novel measurement, so  $T_1 = 1$ . In general, the probability of the event that  $s_0$  receives a novel measurement in a round  $t$ , denoted by  $\mathcal{S}_{s_0}(t) > \mathcal{S}_{s_0}(t-1)$ , is equal to the probability of calling a helper sensor

$$p(\mathcal{S}_{s_0}(t) > \mathcal{S}_{s_0}(t-1)) \geq \left(1 - \frac{|\mathcal{U}_0(t)|}{h_1}\right), \quad (5.24)$$

The  $s_0$  has more than 50% chance to receive the novel measurement at each round (since the number of non-helpers is lower than  $1/2|\mathcal{H}_1|$ , see Eq. 5.23), due to the probabilistic data transmission. This happens when  $|\mathcal{U}_0(t)|/h_1$  is smaller than  $1/2$ . In our scenario, the number of non-helper sensors  $|\mathcal{U}_0(t)|$  in the round  $t$  depends on the number of measurements in  $\mathcal{H}_j$  for  $j > 1$  that arrived to sensors  $s_i \in \mathcal{H}_1$  during previous rounds. The value of  $|\mathcal{U}_0(t)|$  can increase, stagnate or decrease over rounds. For instance, the number of non-helpers increases over two consequent rounds if the sensor  $s_i \in \mathcal{H}_1$  called by  $s_0$  transmits a novel message, but in the same round,  $s_i$  calls the sensor  $s_j$  with  $s_j \notin \mathcal{H}_2$  that can not increase the number of novel measurements at  $s_0$  in the following rounds. In the opposite case, the number of non-helper nodes decreases. The number of non-helpers remains the same in two consequent rounds if the sensor  $s_i \in \mathcal{H}_1$  that has one novel message sends it to  $s_0$ , while at the same time it also receives a message novel to  $s_0$ .



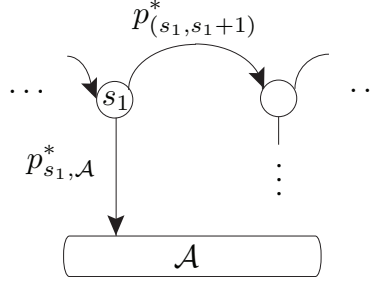
**Figure 5.13:** Schematic illustration of a general discrete phase type distribution, defined by the absorbing Markov chain with one absorbing state  $\mathcal{A}$ . Each of the states of the Markov chain  $\{1, 2, \dots, s^*\}$  represents one of the phases. We use it to model the random variable  $T_r$  that describes the number of rounds that pass until the interpolation sensor that collected  $r$  sensor measurements receives a novel measurement that is different to the previously collected values.

In this work, we assume the worst case scenario, where the number of non-helpers increases at every round. Then, the duration of the first phase is equal to the number of rounds for which  $p(|\mathcal{U}_0(t)|) < 1/2$  holds. Given  $p(|\mathcal{U}_0(t)|) = |\mathcal{U}_0(t)|/h_1 < 1/2$ , we conclude that  $t < h_1/2$ , or,  $t \in [1, \lfloor h_1/2 \rfloor]$ . Here, the operator  $\lfloor \cdot \rfloor$  is the floor operator. The random variable  $T_r$  is bounded in a stochastic sense by the random variable  $Y_r$ ,  $T_r \prec_{st} Y_r$ , that follows the geometric probability  $Y_r \sim \text{Geom}(p_g)$ ,  $p_g \geq 1/2$ .

### (b) The second dissemination phase

At the beginning of this phase, the probability of the non-helper sensors is  $p(|\mathcal{U}_{s_0}(t)|) > 1/2$ . Since  $p(|\mathcal{U}_{s_0}(t)|) > 1/2$ , the interpolation sensor has a low probability to receive a novel message in the first several rounds of this phase. However, over rounds, the novel measurements of sensors in  $\mathcal{H}_j$ ,  $j > 1$  reach sensors in  $\mathcal{H}_1$ . Then, the probability  $p(|\mathcal{U}_{s_0}(t)|)$  monotonically drops and  $s_0$  may receive a novel measurement after a number of dissemination rounds  $\mathcal{R} = \{1, 2, \dots, s^*\}$ . From these facts we observe that a random variable  $T_r$  in the second dissemination phase is related to the class of the discrete-phase distributions. The absorbing Markov chain that models such sequential geometric distributions is given in Fig. 5.4.2. The states of the discrete-phase distribution are the elements of  $\mathcal{R}$ , while its absorption state  $\mathcal{A}$  models the event that  $s_0$  received a novel message. The probabilities  $\{p_{1,\mathcal{A}}^*, \dots, p_{s^*,\mathcal{A}}^*\}$ ,  $\{p_{1,2}^*, \dots, p_{(s-1)^*,s^*}^*\}$  and  $\{p_1^*, \dots, p_{s^*}^*\}$  define the absorption probability (i.e., the sensor  $s_0$  receives a novel message), the state transition probabilities (i.e., the sensor  $s_0$  does not receive a novel message in the round  $t$  and it shifts to the next state) and the recurrent probability (probability to stay in the same state for one more round), respectively. In our setup, the recurrent probabilities are zero, because the states in  $\mathcal{R}$  represents the value of the round, which continuously increases. We define below these probabilities.

We first study the state transition probabilities. The sensor  $s_0$  does not receive a novel message if it calls a non-helper sensor. This happens with probability defined in Eq. (5.23). Then,  $s_0$  does not receive a novel message during consecutive rounds as long as the probability of calling a non-helper sensor remains  $\frac{|\mathcal{U}_{s_0}(t)|}{h_1} > 1/2$  in a novel round  $t$ . For the current state  $s_1$ , this probability is



**Figure 5.14:** Schematic illustration of probabilities of the state  $s_1$  in Markov chain with one absorbing state.

equal to

$$p_{(s_1-1), s_1}^* = p\left(\dim(\mathcal{S}_{s_0}(t)) = r \mid \dim(\mathcal{S}_{s_0}(t-1)) = r\right) \leq 1 - \left(1 - \frac{|\mathcal{U}_{s_0}(t)|}{h_1}\right) = \frac{|\mathcal{U}_{s_0}(t)|}{h_1}. \quad (5.25)$$

We now study the probability that the sensor obtains a novel measurement within one dissemination round, where the current state in the Markov chain is  $s_1$ . This probability is called the absorption probability and it is denoted by  $p_{s_1, \mathcal{A}}^*$  for the state  $s_1$ , as previously mentioned. To analyze the absorption probability values, we compute the state transition probability. Then, the absorption probability is equal to  $p_{s_1, \mathcal{A}}^* = 1 - p_{s_1, s_1+1}^*$  for the state  $s_1$ . Note that the transition probabilities  $p_{s_1-1, s_1}^*$  (defined in Eq. (5.25)) and  $p_{s_1, s_1+1}^*$  given in Fig. (5.14) are different. The latter one is related to probability that  $T_r$  increases. In particular, it is the probability of the event that a non-helper sensor remains the non-helper sensor in the following round, conditional to the event that  $s_0$  did not receive novel messages in both rounds.

Let  $F(t)$  denote the event that the sensor  $s_0$  fails to obtain a novel sensor measurement at the round  $t$ . Then, the probability that the non-helper sensor  $s_j \in \mathcal{U}_{s_0}(t)$  remains a non-helper sensor in the following round  $(t+1)$ , conditional to  $F(t)$  reads

$$p(s_j \in \mathcal{U}_{s_0}(t+1) \mid s_j \in \mathcal{U}_{s_0}(t), F(t)) \leq \left(\frac{1}{h_1} \frac{d_{i, h_1}}{d_{h_1}}\right) \frac{|\mathcal{U}_{\mathcal{H}_1}(t)|}{h_2}. \quad (5.26)$$

The ratio on the right hand side is equal to the probability of choosing a particular sensor  $s_j \in \mathcal{H}_1$  that does not receive a novel measurement from the set  $\mathcal{H}_2$  of size  $h_2$ . Here,  $\mathcal{U}_{\mathcal{H}_1}$  is the set of non-helper sensors in  $\mathcal{H}_2$ .

Now, we observe the event illustrated in Fig. (5.14), that a non-helper sensor remains a non-helper in two consequent rounds, where in these rounds  $s_0$  does not receive a novel message. We introduce the random variable  $Z_r$  to model this event. Then, this event is equal to choosing one out of  $|\mathcal{U}_{s_0}(t+1)|$  non-helper sensors that remain non-helpers, which occurs with probability in Eq.

(5.26), so  $Z_r$  is bounded in the stochastic sense by a binomial distribution

$$Z_r \prec_{st} \text{Bin}\left(\underbrace{|\mathcal{U}_{s_0}(t)|}_{n_b}, \underbrace{\frac{1}{h_1} \frac{d_{i,h_1}}{d_{h_1}} \cdot \frac{|\mathcal{U}_{\mathcal{H}_1}(t)|}{h_2}}_{p_b}\right), \quad (5.27)$$

where one non-helper was chosen from  $n_b$  with probability  $p_b$ , where  $n_b$  and  $p_b$  are the parameters of the binomial distribution. Also,  $(t+1)$  is denoted as  $t$ , for convenience. The Chernoff bound [125] for the binomial random variable  $\text{Bin}(n_b, p_b)$  is given by

$$\begin{aligned} p\left(\text{Bin}\left(|\mathcal{U}_{s_0}(t)|, \frac{1}{h_1} \frac{d_{i,h_1}}{d_{h_1}} \cdot \frac{|\mathcal{U}_{\mathcal{H}_1}(t)|}{h_2}\right) \geq |\mathcal{U}_{s_0}(t)| \frac{1}{h_1} \frac{d_{i,h_1}}{d_{h_1}} \cdot \frac{|\mathcal{U}_{\mathcal{H}_1}(t)|}{h_2} (1+\epsilon)\right) \\ \leq \exp\left(-\frac{\epsilon^2 d_{i,h_1} |\mathcal{U}_{s_0}(t)| |\mathcal{U}_{\mathcal{H}_1}(t)|}{4d_{h_1} h_1 h_2}\right). \end{aligned} \quad (5.28)$$

We know that for this dissemination phase  $|\mathcal{U}_{s_0}(t)| \leq h_1/2$  holds and we want to examine when the exponential function has a small values. We here make an assumption that dissemination at sensors in  $\mathcal{H}_1$  is also in the second dissemination phase. Then it similarly holds  $|\mathcal{U}_{\mathcal{H}_1}(t)| \leq h_2/2$ , so the exponent value boils down to  $\exp(-\frac{\epsilon^2 d_{i,h_1}}{16d_{h_1}})$ . We choose the value of  $\epsilon$  to be  $\epsilon^* = 4\sqrt{3} \sqrt{\frac{\ln(d_{i,h_1}/d_{h_1})}{(d_{i,h_1}/d_{h_1})}}$ , to enforce the exponent bound to scale inverse to the third degree of the probability  $d_{i,h_1}/d_{h_1}$ . The probability  $d_{i,h_1}/d_{h_1}$  is equal to the probability of choosing the transmission links that help in receiving novel messages by  $\mathcal{H}_1$  and  $s_0$ , which is possible only over the links  $d_{i,h_1}$ .

Using  $\epsilon^*$ , we obtain  $\exp(-\frac{\epsilon^{*2} d_{i,h_1}}{16d_{h_1}}) = 1/(d_{i,h_1}/d_{h_1})^3$ . From Eq. (5.28)

$$p\left(\frac{|\mathcal{U}_{s_0}(t+1)|}{h_1} \leq \frac{1}{(h_1)^2} \frac{d_{i,h_1}}{d_{h_1}} \cdot \frac{|\mathcal{U}_{\mathcal{H}_1}(t)|}{h_2} (1+\epsilon)\right) \geq 1 - 1/(d_{i,h_1}/d_{h_1})^3. \quad (5.29)$$

Finally, we compute the absorption probabilities. The value of the  $p_{1,\mathcal{A}}^*$  is equal to the value  $p_{1,\mathcal{A}}^* = \frac{|\mathcal{U}_{s_0}(t+1)|}{h_1}$ , bounded by Eq. (5.29),  $|\mathcal{U}_{s_0}(t+1)| \leq \frac{1}{h_1} \frac{d_{i,h_1}}{d_{h_1}} \frac{|\mathcal{U}_{\mathcal{H}_1}(t)|}{h_2} (1+\epsilon^*)$ . When the connectivity and the sets  $\mathcal{H}_J$  are given, the remaining absorption probabilities can be similarly computed.

This phase lasts as long as the value  $|\mathcal{U}_{s_0}(t)|/h_1 < 1$ . Then, the phase termination state  $s^*$  can be computed as the maximal  $|\mathcal{U}_{s_0}(t)|$  that satisfies  $|\mathcal{U}_{s_0}(t)|/h_1 < 1$ , if the connectivity and the sets  $\mathcal{H}_J$  are given.

Finally, the random variable  $T_r$  takes the value in  $\mathcal{R}$  by the absorption probability  $p_{s_1,\mathcal{A}}^*$ .

### 5.4.3 Data dissemination analysis

The performance of the iterative interpolation algorithm depends on the available data at the interpolation sensor when the measurement transmission is performed in dissemination rounds. We here study data dissemination in the network. We first introduce the network connectivity model, which we then use in estimations of the average number of sensors reachable from the interpolation sensor in the given number of hops. This value is used to estimate the minimal number of dissemination rounds sufficient for gathering of the full set of sensor values used in the



interpolation algorithm for ideal cases, when sensors use the minimal hop path to deliver their measurement to the interpolation sensor.

We here provide detail on the used connectivity model. The deterministic connectivity models [126] are very often unsuitable for practical network scenarios, because connections between sensors in the network may change in time. Therefore, we assume here that the sensor connectivity follows a probabilistic model. Particularly, the number of sensors' connections follows a certain probability distribution. This connectivity distribution function is modeled by a generating function. For more details on generating functions refer to [127].

In order to illustrate the performance of our distributed interpolation algorithm, we assume that the connectivity of the local neighborhood in the network follows a *small world model* [128]. The main characteristic of such networks is that the shortest path between two sensors increases logarithmically with the increase of the graph size. We use the particular connectivity model of random graphs with specific degree distributions as in [129]. We below compute the hop distance between two sensors for this model. This value is used later on to compute the minimal number of dissemination rounds for gathering of the set of sensor values used in the interpolation algorithm, in the ideal case (sensors send their values over the shortest transmission paths to the interpolation sensor). We now describe properties of the probabilistic model.

Let  $d$  denote the degree of a sensor, which is defined as the total number of connections with other sensors. Then, the probability that a sensor has exactly  $d$  connections is given by the generating function  $D(x)$  that reads

$$D(x) = \sum_{d=0}^{\infty} p_d x^d, \quad (5.30)$$

where  $p_d$  is the probability that a sensor in the network has the degree  $d$  and  $|x| \leq 1$  is the parameter of the generating function. The distribution  $p_d$  is properly normalized, such that  $D(1) = \sum_d p_d x^d|_{x=1} = 1$ . The function  $D(x)$  has several important properties [129]. The function  $D(x)$  contains all the information about the probability  $p_d$  because  $p_d$  is computed as the  $d$ -th derivative of the function  $D(x)$ ,  $p_d = \frac{1}{d!} \frac{\partial^d D}{\partial x^d} |_{x=0}$ . When the generating function  $D(x)$  is given, the mean and the higher order moments of  $D(x)$  are computed using the derivatives  $\bar{d} = \sum_d d p_d = D'(1)$  and  $\bar{d}^n = \sum_d d^n p_d = [(x \frac{\partial}{\partial x})^n D(x)] |_{x=1}$ . The sum of the degrees of the randomly chosen set of sensors in the network is equal to the power of the generating function proportional to the set size. For example, the distribution of the sum of degrees of two randomly chosen sensors is equal to  $D(x)^2$ , since  $D(x)^2 = \left( \sum_d p_d x^d \right)^2 = \sum_i \sum_j p_i p_j x^{i+j}$ . The crucial property used for determining the distribution of the sensor degree comes from the observation that the measurement arrives to a sensor with a probability proportional to the degree of that sensor [129]. Let  $j \in \{1, \dots, J\}$  denote the shortest number of hops between the interpolation sensor and the neighbor sensors used for interpolation. The degree probability that a neighbor of the observed sensor has a degree  $d$  is proportional to  $d p_d$  (not only  $p_d$ ), it holds  $\frac{\sum_d d p_d x^d}{\sum_d d p_d} = x \frac{D'(x)}{D'(1)}$ .

Next, we provide details about how to model the degree distribution of  $d$  graph neighbors of the interpolation sensor given  $D(x)$ . For a randomly chosen interpolation sensor, the degree probability distribution of its  $d$  neighbors is proportional to the remaining number of connections, without counting the edges between the neighbors and the interpolation sensor. Now, the distribution of

the connections of the first-hop neighbors is given by  $D_1(x) = \frac{D'(x)}{D'(1)} = \frac{1}{\bar{d}}D'(x)$ , where  $\bar{d}$  is the average degree of the sensors. We observe the model where the number of connections between the interpolation sensor and its first-hop neighbors is small compared to the number of connections among its first-hop neighbors. Then, the generating function for the probability distribution of the number of connections for the second-hop neighbors of the interpolation sensor is  $\sum_d p_d (D_1(x))^d = D(D_1(x))$ . The number of the second-hop neighbors is  $\bar{d}_2 = \left(\frac{\partial}{\partial x} D(D_1(x))\right) |_{x=1} = D'(1)D_1'(1)$ . For the  $j$ -hop neighbors, the distribution is equal to  $D_j(x) = D(D_1(\dots D_j(x) \dots))$ , because  $D_1(1) = 1$ , while the number of neighbors is  $\bar{d}_j = D'(1)D_1'(1) \dots D_{j-1}'(1)$ . The number of neighbors that are  $j$ -hop distant from the interpolation sensor is now equal to

$$\bar{d}_j = \frac{\partial D_j(x)}{\partial x} |_{x=1} = D_j'(1)\bar{d}_{j=1}.$$

Since  $\bar{d}_1 = D'(1)$ , we rewrite  $\bar{d}_j = \left(\frac{\bar{d}_2}{\bar{d}_1}\right)^{j-1}\bar{d}_1$ . Finally, we can compute the shortest number of hops  $j$  between two randomly chosen sensors in the network. As the total number of neighbors is equal to  $q$ , the following holds:  $q = 1 + \sum_{j=1}^J \bar{d}_j$ , where  $J$  is the maximal, but unknown value of  $j$ . Finally, from this equation and the definition of  $\bar{d}_j$ , we obtain that amongst  $q$  sensors used in the interpolation algorithm, the sensors located the farthest from the interpolation sensor can reach the interpolation sensor in at least  $J$  transmission rounds, where  $J$  is equal to

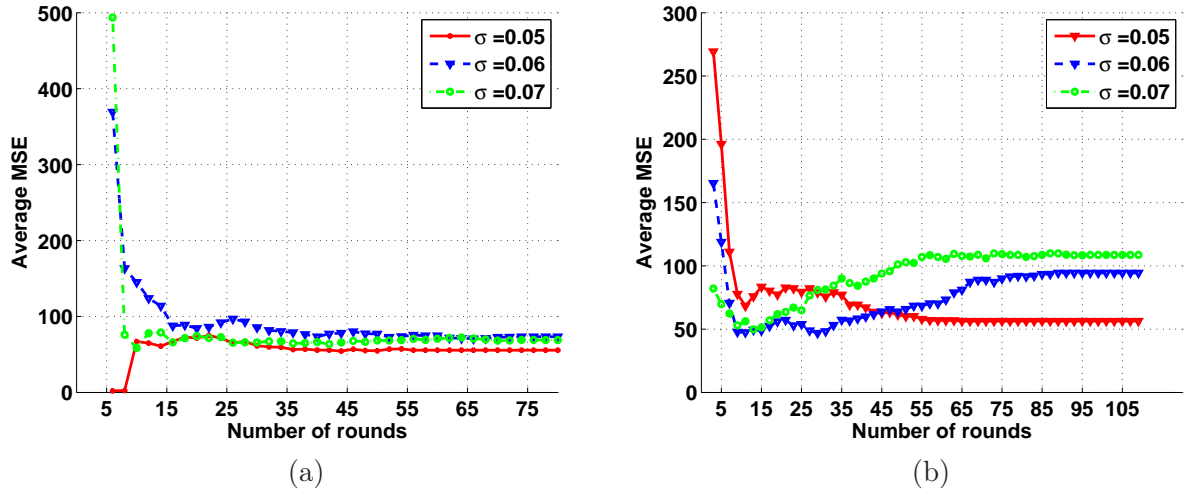
$$J = \frac{\log\left((q-1)(\bar{d}_2 - \bar{d}_1) + \bar{d}_1^2\right) - 2\log(\bar{d}_1)}{\log(\bar{d}_2/\bar{d}_1)}. \quad (5.31)$$

This value can be used for estimation of the minimal number of dissemination rounds sufficient for gathering of the full set of sensor values used in the interpolation algorithm for particular networks with known parameters  $d$ ,  $p_d$  and  $q$ .

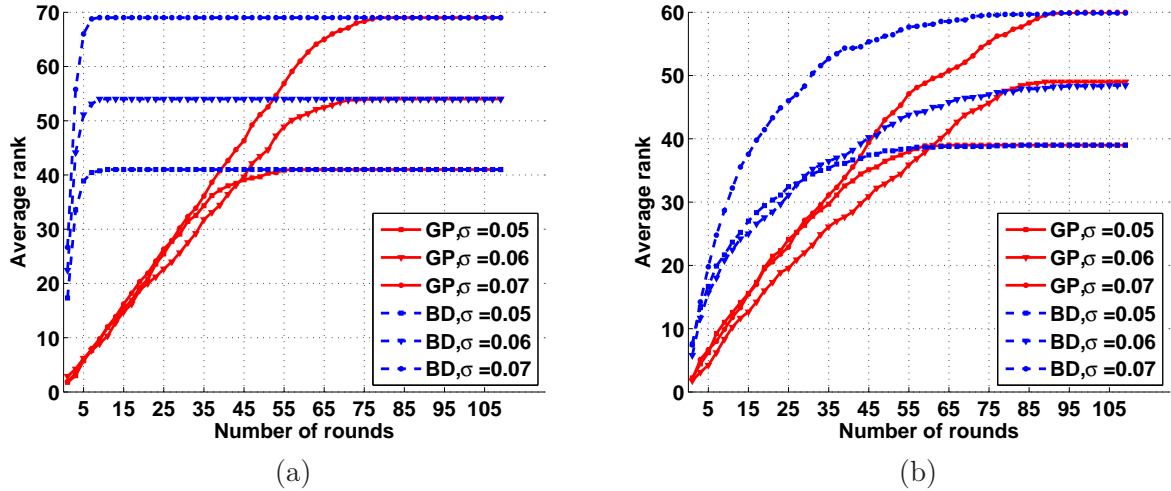
#### 5.4.4 Experimental measurement transmission analysis

We provide interpolation and data transmission analysis for a spherical *Room irregular* dataset (illustrated in Fig. 5.6) and discuss them. In particular, we perform iterative interpolation for a subset of seventy sensors that are randomly distributed on the sphere. Provided values are the mean values of ten random network realizations. The key parameters for the iterative interpolation are the selection of parameters  $\sigma$  and  $\rho$  and the selection of the network protocol. The parameters  $\sigma$  and  $\rho$  determine the data used for interpolation (defines the set  $\mathcal{N}_{s_i}$ ) and the communication range parameter (defines the set  $\mathcal{N}_{s_i}^{(g)}$ ), respectively.

We now provide the iterative interpolation results for the two main dissemination cases described in section 5.4.2 (see Fig. 5.12). We show in Fig. 5.15 (a) the iterative interpolation results for the case when sensors used for interpolation are positioned within the communication range  $\rho$  from the interpolation sensor. The interpolation results for the second case, when a subset of sensors is positioned within the communication range  $\rho$  is given in Fig. 5.15 (b). The slight differences in the interpolated results for cases (a) and (b) are the consequence of random choices of network realizations. The results demonstrate that the iterative interpolation algorithm converges to the solution



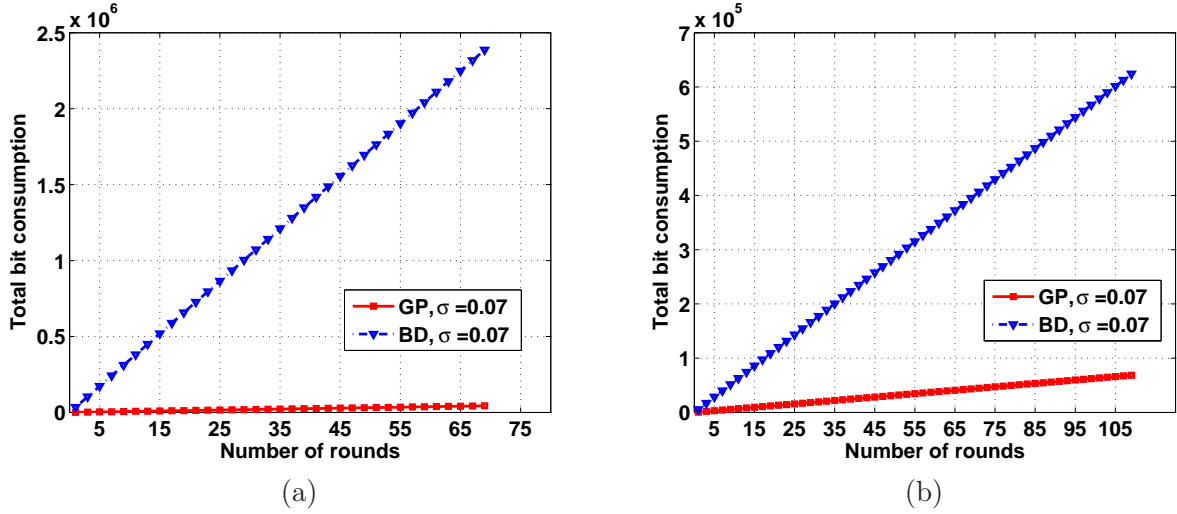
**Figure 5.15:** Mean square error of interpolation decreases over the transmission rounds,  $(|\mathcal{N}_{s_i}|, \sigma) = (\{42, 54, 69\}, \{0.05, 0.06, 0.07\})$ : (a) most of the sensors are positioned within the communication range  $\rho$ ; (b) a small subset of sensors is positioned within the communication range  $\rho$ .



**Figure 5.16:** Average rank values of the sensors vs. the number of transmission rounds: (a) most of the sensors are positioned within the communication range  $\rho$ ; (b) a small subset of sensors is positioned within the communication range  $\rho$ .

in a reasonable number of rounds. Used parameters are  $(|\mathcal{N}_{s_i}|, \sigma) = (\{42, 54, 69\}, \{0.05, 0.06, 0.07\})$ .

We proceed with the analysis of the rank of data available at sensors. We use two data dissemination protocols in simulations. The first one is the gossip algorithm (pull mechanism) denoted by GD, where each sensor randomly calls one of the sensors from a direct link neighborhood to forward a measurement. The second algorithm follows a broadcasting rule, where all sensors forward the



**Figure 5.17:** Total bit consumption for different number of transmission rounds for cases: (a) most of the sensors are positioned within the communication range  $\rho$ ; (b) a small subset of sensors is positioned within the communication range  $\rho$ .

randomly chosen measurement from their set to all their direct link neighbors  $\mathcal{N}_{s_i}^{(g)}$ . In graphs, we denote this broadcasting algorithm as BD. The average rank of messages available at sensors is illustrated in Fig. 5.16. We notice that the rank analysis for GD matches the model given in Section 5.4.2. In the first phase, the rank increases in every round. The second phase models the flat regions and the iterative interpolation result converges to the estimated value before the second transmission phase terminates. The broadcasting protocol disseminates the messages in the network in couple of rounds. However, in terms of bit consumption the algorithm denoted as BD is inefficient, as illustrated in Fig. (5.17).

Finally, we estimate the generating function  $D(x)$  (defined in Eq.(5.30)) from the data and we compute that messages reach the interpolation in average in  $J = 2$  hops in the ideal case. This is valid for the case when most of the sensors are positioned within the communication range  $\rho$ . In the case (b) the computed number of hops over the shortest path between the neighboring sensors and the interpolation sensor is equal to  $J = 4$ .

## 5.5 Conclusions

We have investigated the interpolation problem for sensor network data distributed on a geometric manifold. We have proposed a new distributed interpolation method of an arbitrary set of values given irregular and noisy sensor measurements that incorporates the geometry of the manifold. We have performed extensive experimental analysis for spherical datasets in noiseless and noisy environments and shown that the proposed method achieves better performance in terms of data interpolation distortion than a baseline Nearest Neighbor method, particularly in the presence of noise. Finally, we have shown that when the local neighborhood in the sensor network can be represented by a small world model with probabilistic connections, the proposed distributed

---

algorithm converges fast to good interpolation performance, and that our solution represents an interesting alternative for processing sensor data on geometric manifolds. However, we also note that the performance of the proposed method is quite sensitive to the choice of the standard deviation of the kernel, which is certainly a topic for further investigations.



## Chapter 6

# Conclusions and Perspectives

This thesis investigates multiple instances of distributed signal processing problems under the perspective of modern challenges in sensor networks. We have first studied the problem of distributed detection of the defective sensors in networks. We have proposed a probabilistic message design of binary detection signals propagated distributedly, which allows for detection with an inexpensive distance decoder at each sensor in the network. Next, we have studied efficient data gathering methods for signal reconstruction with small number of messages in sensor networks. In particular, we have proposed a practical reconstruction algorithm. We have then studied the interpolation problem in distributed sensor networks deployed on geometric manifolds and we have designed a distributed interpolation algorithm that incorporates manifold geometry for signal estimation, for arbitrary set of interpolation positions.

We first addressed the problem of failure sensor detection in sensor networks. We proposed a novel distributed algorithm that detects a small number of defective sensors in networks. We propose a probabilistic design of signal messages based on local sensor observations. These messages represent observations of the binary failure signal and they are communicated using a gossip algorithm. Sensors combine the collected messages into a new message that is further propagated in the network by a gossip algorithm. After a number of collected messages at sensors, defectives in the network are detected using a simple and efficient distance decoder. We derived the lower bound on the required number of linearly independent messages per cluster that sensors need to collect to ensure detection of one defective sensor with high probability. This result is given for the worst case scenario. We next show experimentally that this derived value is quite smaller in practice, even for the small size networks, which confirms the validity of the theoretical bound. The simulation results show that the proposed method outperforms other detection schemes in terms of successful detection probability.

Next, we considered the problem of efficient data collection in sensor networks and studied the performance of the reconstruction methods for inverse problems, given partial network data. We propose a distributed gathering algorithm that updates messages at sensors during the message transmission. The observations represent the finite field combinations of quantized sensor network measurements. The proposed encoding guarantees innovative message collection with high probability. For the proposed gathering framework, we developed the analytic decoding error expression for signals with locally correlated values and analyzed its dependence on network parameters. In-

creasing the size of the finite field and narrowing the potential solution set decrease the decoding error. We also investigated coding matrix properties that decrease the analytic decoder error and proposed its practical probabilistic design. Finally, we have proposed the practical Belief Propagation based decoder algorithm for signal reconstruction that exploits sensor priors and the topology structure knowledge. We have shown experimentally that for signals with locally correlated values, the proposed algorithm outperforms or is competitive with comparison gathering methods in terms of the number of bits transmitted in the network for high probability signal recovery.

We studied the interpolation problem in sensor networks for the set of arbitrary interpolation point positions. The goal is to compute values of the underlying function given the noisy values of sensors that are positioned on a geometric manifold. The sensor distribute their observations to other sensors in ad-hoc way. We posed the interpolation problem as the convex optimization problem that incorporates the information of the geometric manifold. We solved the problem analytically for a general manifold. The proposed method is based on the kernel regression and it exploits properties of high dimensional signals and efficiently manages noisy irregular samples. We proposed a novel distributed algorithm, where sensors perform computations as they collect novel neighbor sensor measurements. We provided the analysis of data dissemination in the neighborhood. We performed the experiments for typical sample data structures on 2-d spherical signals. Proposed methods provide higher quality reconstruction results then the comparison method in terms of the signal quality measures.

## 6.1 Limitations

Though our work on detection of failure signals have offered many insights into building of distributed algorithms by using the flexible designs, it has some limitations. The first limitation comes from the noise model, which considers that bit flips may occur only for the nonzero values. However, due to noise, it may occur that the zero valued bits are flipped as well. Another limitation arises from the fact that sensors generate the failure signal messages based on local observations. In our work, we assume that sensors use a simple thresholding criteria to build the messages. However, we do not consider that sensors themselves may introduce the error (false alarms or missed detections) in messages. Our algorithm is not robust in such cases. As discussed in the future work, an extension of this work would be to consider solutions more robust to noise. The second limitation is that the detection algorithm is designed to detect a small portion of defective sensors that are geographically distributed. However, it is well known that in many applications, a failure of one sensor may cause a neighbor sensor failure. Therefore, distributed detection of the failure for network regions is the important research topic with many potential applications, for instance, in power networks. The experimental analysis provided in this thesis is limited to synthetic data. We considered to apply the algorithm for a rare event analysis; however, it was not pursued due to lack of a ground truth signals.

The study of inverse problems in finite fields faces several significant limitations. The main source of limitations arises from the decoder complexity. Due to the nature of the problem, decoders in general search for the solution amongst a full set of potential solutions, known a priori by the decoder. However, the size of this solution set is known to grow exponentially with the number of unknowns. Therefore, the experimental analysis of large scope networks may be computationally



challenging, so we limit the analysis provided in this thesis to the analysis of small and medium sensor networks. The main limitation of our algorithm is that it is not prone to noise, due to the nonlinearity of the problem. Therefore, the denoising problem in finite fields is a very interesting problem.

The proposed iterative interpolation method for manifold network data faces several challenges. The most important issue is related to the choice of the two parameters: the communication range parameter, which determines the network connectivity, and the parameter that determines the useful data for interpolation. The main limitation of our algorithm is that it does not provide the way to set these values. Also, it does not optimize these two parameters jointly. The interesting direction is to investigate their joint influence on the convergence in further works.

## 6.2 Future Work

We now briefly discuss some of the most interesting directions in sensor network signal processing that are related to the problems discussed in this thesis. First, we discuss the extension of the distributed detection and the recovery problems in sensor networks to noisy scenarios. It is not trivial to model the influence of noise on data values, due to the nonlinearity of the problem. The additional messages that encode the noise information should be designed and transmitted together with the signal. However, these additional messages should not burden the network. In addition, developing novel approaches that speed up signal decoding in ring algebraic structures are very important, particularly for the future generations of sensors, that will operate with more than two states per bit.



# Appendix A

## Appendix

### A.1 Model for probability $P(q|m)$

We model  $P(q|m)$  that represents the probability of the event that multiple defective sensors exist in the same cluster but only a subset of defective sensors participates in the test. This event introduces errors while detection of defective sensors. Recall that sensors participate in the test with the probability  $q$ . For  $m$  defective sensors, all the possible realizations of tests are given with elements of the polynomial  $(q+(1-q))^m$ . For example,  $q^m$  represents that all the  $m$  defective sensors take part in the test,  $q^{m-1}(1-q)$  means that only one defective element does not participate in the test, and so on. This polynomial represents the binomial expansion of the form  $(x+y)^m$ , with  $x=q$  and  $y=(1-q)$ . Polynomial expansion is equal to  $(x+y)^m = c_0x^m + c_1xy^{m-1} + \dots + c_my^m$  and the coefficients  $c_i = \binom{m}{i}$  represent the numbers of  $i$ -th row of Pascal's triangle. Messages that do not cause decoding error are the messages of all zeros and of all ones. These messages occur with probabilities  $q^m$  and  $(1-q)^m$ , respectively and they have coefficients equal to 1. Note that  $(q+(1-q))^m = 1^m = 1$  and that probability of error event is therefore equal to:

$$P(q|m) = \frac{1 - q^m - (1 - q)^m}{(q + (1 - q))^m} = 1 - q^m - (1 - q)^m. \quad (\text{A.1})$$

### A.2 Ring properties

A ring  $(R, \oplus, \odot)$  is an algebraic structure with the two binary operators  $\oplus : R \rightarrow R$  and  $\odot : R \rightarrow R$  which satisfies the following properties:

- $(R, \oplus)$  is an Abelian group under addition (closure, associativity under addition, existence of additive identity, existence of additive inverse, commutativity of addition).
- $(R, \odot)$  is a monoid under multiplication (closure, associativity under multiplication, existence of multiplicative identity)
- The distributive laws  $\forall a, b, c \in R$  hold:
  - (a)  $a \odot (b \oplus c) = (a \odot b) \oplus (a \odot c)$

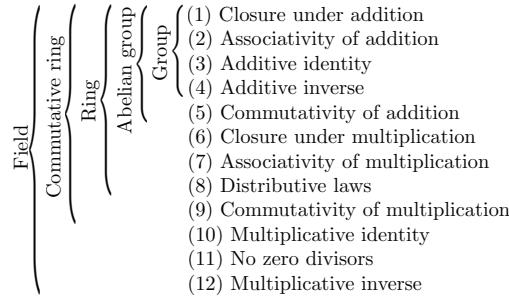


Figure A.1: Properties of structures in algebra.

$$(b) (a \oplus b) \odot c = (a \odot c) \oplus (b \odot c)$$

Therefore, a structure  $(F_q, \oplus, \odot)$  whose elements are  $\{0, 1, \dots, q - 1\}$  and the addition and multiplication are modulo  $q$  operations, where  $q$  has an arbitrary value, forms the ring.

### A.3 The distribution of the product of two discrete random variables

The probability mass function (pmf) of the product of two discrete random variables  $X$  and  $Y$ , namely  $h(XY)$ , is defined below. We assume that the discrete random variable  $X$  with known pmf  $f(X)$  is positive and defined on the interval  $(i_1, i_2)$ , and that the discrete random variable  $Y$  with known pmf  $g(Y)$  is positive and defined on the interval  $(i_3, i_4)$ . These pmf-s can be defined in a discrete piecewise fashion. We assume that the intervals of these variables are defined in the  $I$ -st quadrant so  $0 < i_1 < i_2 < \infty$ ,  $0 < i_3 < i_4 < \infty$ ,  $i_1 i_3 < i_2 i_4$ . Using the transformation technique, the transformation  $Z = X$ ,  $V = XY$  constitutes the “1 – 1” mapping from  $\mathcal{A} \rightarrow \mathcal{B}$ ,  $\mathcal{A} = \{(x, y) | i_1 < x < i_2, i_3 < y < i_4\}$ ,  $\mathcal{B} = \{(z, v) | i_1 < z < i_2, i_3 z < v < i_4 z\}$ , where  $Z = X$  and  $V = XY$ . Now we can write:

$$h(V) = f_{Z,V}(z, v) = \sum_{z=i_1}^{i_2} f(z)g\left(\frac{v}{z}\right) = \sum_{x=i_1}^{i_2} f(x)g\left(\frac{v}{x}\right) \tag{A.2}$$

### A.4 Disjunct and separable matrix definitions

**Theorem 1 (Definition 1).** *Disjunct matrix:* A boolean matrix  $\mathbf{M}$  with  $S$  columns  $M_1, M_2, \dots, M_S$  is called  $(K, \epsilon)$ -disjunct if, for every subset of  $T$  columns with  $|T| \leq K$  and every  $i \in \{1, \dots, S\}$  holds:

$$| \text{supp}(M_i) \setminus \left( \bigcup_{j \in T \setminus \{i\}} \text{supp}(M_j) \right) | > \epsilon, \tag{A.3}$$

where  $\text{supp}(M_i)$  denotes the nonzero elements (support) of the column  $M_i$  and  $\setminus$  is the set difference operator.

**Theorem 2 (Definition 2).** *Separable matrix: A boolean  $B \times S$  matrix  $\mathbf{M}$  is  $K$ -separable if and only if  $\forall \mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V} = \{s_1, \dots, s_S\}$  and  $|\mathcal{S}_1|, |\mathcal{S}_2| \leq K$  holds  $\cup_{j \in \mathcal{S}_1} M_j \neq \cup_{i \in \mathcal{S}_2} M_i$ .*

In words, for a matrix to be  $K$ -disjunct, the Boolean sum of every  $K$  columns does not contain any other column in the matrix, while  $K$ -separable property is fulfilled if the Boolean sum of every  $K$  columns is unique. Note that  $K$ -disjunct matrices are also  $K$ -separable, while the reverse does not hold.

## A.5 Taylor multivariate expansion

For the sake of completeness, we provide the definition of the Taylor multivariate expansion for functions  $\tilde{f} : \mathbb{R}^d \rightarrow \mathbb{R}$  that are  $n$ -times differentiable. Let  $\bar{\alpha} \in \mathbb{N}^d$  be the index where  $\bar{\alpha} = \bar{\alpha}_1 + \dots + \bar{\alpha}_d$ ,  $\bar{\alpha}! = \bar{\alpha}_1! + \dots + \bar{\alpha}_d!$  and  $\mathbf{m}_i^{\bar{\alpha}} = m_i(1)^{\bar{\alpha}_1} \dots m_i(d)^{\bar{\alpha}_d}$ . Let the notation  $D^{\bar{\alpha}} \tilde{f} = \frac{\partial^{|\bar{\alpha}|} \tilde{f}}{\partial m_i^{\alpha_1}(1) \dots \partial m_i^{\alpha_d}(d)}$ ,  $|\bar{\alpha}| < n$  represents the higher order partial derivatives of  $\tilde{f}$ . Then, given the neighborhood of  $s_0 \in \mathcal{N}_{s_i}$ , the value of the function in the sensor  $s_i$  can be written as

$$\tilde{f}(\mathbf{m}_i) = \sum_{|\bar{\alpha}| \leq n} \frac{D^{\bar{\alpha}} \tilde{f}(\mathbf{m}_0)}{\bar{\alpha}!} (\mathbf{m}_i - \mathbf{m}_0)^{\bar{\alpha}} + \sum_{|\bar{\alpha}| = n} h_{\bar{\alpha}}(\mathbf{m}_i) (\mathbf{m}_i - \mathbf{m}_0)^{\bar{\alpha}}, \quad (\text{A.4})$$

where  $h_{\bar{\alpha}} : \mathbb{R}^d \rightarrow \mathbb{R}$  and  $\lim_{\mathbf{m}_0 \rightarrow \mathbf{m}_i} h_{\bar{\alpha}}(\mathbf{m}_i) = 0$ .



# Bibliography

- [1] H. S. M. Coxeter, *Introduction to Geometry*. New York: Wiley; 2nd ed., 1969.
- [2] F. L. Lewis, *Wireless Sensor Networks, in Smart Environments: Technologies, Protocols, and Applications*. John Wiley and Sons, Inc., Hoboken, NJ, USA, 2005.
- [3] C. M. Allen, “Minimization technique for multiple-valued logic systems,” *IEEE Trans. on Computers*, vol. C-17, pp. 182–184, Feb. 1968.
- [4] S. L. Hurst, “Multiple valued logic - its status and its future,” *IEEE Trans. on Computers*, vol. C-33, pp. 1160–1179, Dec. 1984.
- [5] G. Mateos, J. A. Bazerque, and G. B. Giannakis, “Distributed lasso for in-network linear regression,” *Proc. of Int. Conf. Acoustic, Speech and Sig. Proc. (ICASSP)*, 2010.
- [6] D. I. Shuman, P. Vandergheynst, and P. Frossard, “Distributed signal processing via Chebyshev polynomial approximation,” <http://arxiv.org/abs/1111.5239>, 2011.
- [7] A. Boulis, S. Ganeriwal, and M. Srivastava, “Aggregation in sensor networks: An energy-accuracy trade-off,” *Proc. of IEEE Int. Workshop on Sensor Network Protocols and Applications*, 2003.
- [8] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next century challenges: Scalable coordination in sensor networks,” *Proc. of ACM/IEEE Int. Conf. on Mobile computing and networking*, pp. 263–270, 1999.
- [9] M. Rabbat and R. Nowak, “Distributed optimization in sensor networks,” *Proc. of the Int. Conf. on Inf. Proc. in Sensor Networks (IPSN)*, 2004.
- [10] K. P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky, “Bimodal multicast,” *ACM Tran. on Computer Systems*, vol. 17, no. 2, pp. 4–88, May 1999.
- [11] R. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking, “Randomized rumor spreading,” *IEEE Symp. on Found. of Computer Science*, 2000.
- [12] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.
- [13] S. Deb and M. Médard, “Algebraic Gossip: A network coding approach to optimal multiple rumor mongering,” *Proc. of Allerton Conf. on Comm., Control and Computing*, 2004.

- [14] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," *Proc. of the 6th annual ACM Symp. on Princip. of distr. computing*, pp. 1–12, 1987.
- [15] S. Deb, M. Medard, and C. Choute, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2486–2507, 2006.
- [16] D. Mosk-Aoyama and D. Shah, "Information dissemination via network coding," *IEEE Intern. Symposium on Inform. Theory*, pp. 1748–1752, 2006.
- [17] A. Dimakis, S. Kar, J. Moura, M. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE Trans. Inform. Theory*, vol. 98, pp. 1847–1864, Nov. 2010.
- [18] N. Fountoulakis and K. Panagiotou, "Rumor spreading on random regular graphs and expanders," *Proc. of the 13th Int. Conf. on Approximation, and Int. Conf. on Randomiz. and Combin. Optimiz., APPROX-RANDOM*, 2010.
- [19] S. Pradhan and K. Ramchandran, "Distributed Source Coding Using Syndromes (DISCUS): Design and Construction," *IEEE Trans. of Inf. Theory*, vol. 49, no. 3, pp. 626–642, March 2003.
- [20] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed Source Coding for Sensor Networks," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–94, Sept. 2004.
- [21] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Tran. on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [22] P. A. Chou, Y. Wu, and K. Jain, "Practical Network Coding," *Proc. of Allerton Conf. on Comm., Control and Computing*, vol. 41, no. 1, pp. 40–49, 2003.
- [23] C. Fragouli, J.-Y. L. Boudec, and J. Widmer, "Network coding: An instant primer," *ACM SIGCOMM Computer Communication Review*, vol. 36, pp. 63–68, 2006.
- [24] S. B. Cruz, G. Maierbacher, and J. Barros, "Joint source-network coding for large-scale sensor networks," *IEEE Int. Symp. on Inf. Theory Proceedings (ISIT)*, 2011.
- [25] P. K. Varshney, *Distributed Detection and Data Fusion*, 1st ed. Springer-Verlag New York, Inc., 1996.
- [26] J. N. Tsitsiklis, "Decentralized detection," *Proc. of Advanced Statistical Signal Processing*, vol. 2-Signal Detection, pp. 297–344, 1993.
- [27] Q. Tian and E. J. Coyle, "Optimal distributed detection in clustered wireless sensor networks," *IEEE Trans. on Signal Proc.*, vol. 55, no. 7, pp. 3892–3904, 2007.
- [28] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors: Part I-Fundamentals," *Proc. IEEE*, vol. 85, no. 1, pp. 54–63, Jan. 1997.
- [29] R. S. Blum, S. A. Kassam, and H. V. Poor, "Distributed detection with multiple sensors: Part II-Advanced topics," *Proc. IEEE*, vol. 85, no. 1, pp. 64–79, Jan. 1997.



- [30] R. Dorfman, "The detection of defective members of large populations," *Annals of Mathematical Statistics*, vol. 14, pp. 436–440, 1943.
- [31] M. Young and R. Boutaba, "Overcoming Adversaries in Sensor Networks: A Survey of Theoretical Models and Algorithmic Approaches for Tolerating Malicious Interference," *IEEE Communications Surveys and Tutorials*, vol. 13, pp. 617–641, April 2011.
- [32] D.-Z. Du and F. K. Hwang, *Combinatorial Group Testing and its Applications*, 2nd ed. World Scientific, 2000.
- [33] H.-B. Chen and F. K. Hwang, "A survey on nonadaptive group testing algorithms through the angle of decoding," *J. Comb. Optim.*, vol. 15, pp. 49–59, 2008.
- [34] M. Cheraghchi, A. Karbasi, S. Mohajer, and V. Saligrama, "Graph-constrained group testing," *Proc. of Int. Symp. on Inform. Theory (ISIT)*, pp. 1913–1917, 2010.
- [35] M. Mézard and C. Toninelli, "Group testing with random pools: Optimal two-stage algorithms," *IEEE Trans. Inform. Theory*, vol. 57, no. 3, pp. 1736–1745, March 2011.
- [36] Y.-W. Hong and A. Scaglione, "Group testing for sensor networks: The value of asking the right question," *38th Asilomar Conf. on Signals, Systems and Computers*, vol. 2, pp. 1297–1301, 2004.
- [37] W. Dai and O. Milenkovic, "Weighted superimposed codes and constrained integer compressed sensing," *IEEE Trans. Inform. Theory*, vol. 55, pp. 2215–2229, May 2009.
- [38] A. D. Bonis and U. Vaccaro, "Constructions of generalized superimposed codes with applications to group testing and conflict resolution in multiple access channels," *Theor. Comput. Sci.*, vol. 306, no. 1-3, pp. 223–243, 2003.
- [39] M. T. Thai, Y. Xuan, I. Shin, and T. Znati, "On detection of malicious users using group testing techniques," *Proc. of Int. Conf. on Distributed Computing Systems*, pp. 206–213, 2008.
- [40] P. Indyk, H. Q. Ngo, and A. Rudra, "Efficiently decodable non-adaptive group testing," *Proc. of the 21st Annual ACM - SIAM Symp. on Disc. Algorithms*, pp. 1126–1142, 2010.
- [41] R. Rajagopalan and P. K. Varshney, "Data aggregation techniques in sensor networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 4, pp. 48–63, Oct. 2006.
- [42] J. Aćimović, B. Beferull-Lozano, and R. Cristescu, "Efficient distributed decorrelating processing for data gathering in sensor networks," *Proc. of Int. Conf. Acoustic, Speech and Sig. Proc. (ICASSP)*, vol. 4, pp. 837–840, 2005.
- [43] S. K. Narang, G. Shen, and A. Ortega, "Unidirectional graph-based wavelet transforms for efficient gathering in sensor networks," *Proc. of Int. Conf. Acoustic, Speech and Sig. Proc. (ICASSP)*, 2010.
- [44] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," *IEEE INFOCOM*, pp. 2571–2582, 2004.

- [45] Y. Chen, S. Levine, and M. Rao, "Solution of incorrectly formulated problems and the regularization method," *Doklady Akademii Nauk SSSR, Translated in Soviet Mathematics 4*, vol. 4, pp. 1035–1038, 1963.
- [46] —, "Variable exponent, linear growth functionals in image restoration," *SIAM Journal on Applied Mathematics*, vol. 66, pp. 1383–1406, 2006.
- [47] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1-4, pp. 259–268, 1992.
- [48] T. F. Chan, G. H. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," *SIAM Journal on Scientific Computing*, vol. 20, no. 6, pp. 1964–1977, Feb 1999.
- [49] P. Blomgren, T. Chan, P. Mulet, and C. Wong, "Total variation image restoration: numerical methods and extensions," *Proc. of Int. Conf. on Image Processing*, vol. 3, 1997.
- [50] G. Peyré, S. Bougleux, and L. Cohen, "Non-local regularization of inverse problems," *Computer Vision & ECCV 2008, Lecture notes in Computer Science*, vol. 5304, pp. 57–68, Aug 2008.
- [51] D. Zhou and B. Schölkopf, "Regularization on discrete spaces," *Lecture notes in Computer Science*, vol. 3663, p. 361, 2005.
- [52] —, *Discrete regularization*. MIT Press, Cambridge, MA, 2006.
- [53] A. Chambolle, "An algorithm for Total Variation minimization and applications," *J. Math. Imaging Vis.*, vol. 20, no. 1-2, pp. 89–97, Jan 2004.
- [54] G. Peyré, S. Bougleux, and L. Cohen, "Non-local regularization of inverse problems," *Proc. of the Euro. Conf. on Computer Vision*, pp. 57–68, 2008.
- [55] M. Zhu, S. J. Wright, and T. F. Chan, "Duality-based algorithms for total-variation-regularized image restoration," *Comput. Optimization and Applications*, vol. 47, no. 3, pp. 377–400, 2010.
- [56] D. L. Donoho, "Compressed sensing," *IEEE Tran. Tran. on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [57] E. J. Candes and T. Tao, "Decoding by linear programming," *CoRR*, 2005.
- [58] R. G. Baraniuk, "Compressive sensing," *Lecture Notes in IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–120, Jul. 2007.
- [59] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," *Proc. of Int. Conf. of Mobile Comp. and Networking*, pp. 145–156, 2009.
- [60] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed Sensing for Nnetworked Data," *IEEE Signal Processing Magazine*, vol. 25, pp. 92–101, 2008.

- [61] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk, "Distributed compressive sensing," *CoRR*, vol. abs/0901.3403, 2009.
- [62] S. C. Drapper and S. Malekpour, "Compressed Sensing over Finite Fields," *ISIT*, June 2009.
- [63] V. Tan, L. Balzano, and S. C. Draper, "Rank minimization over finite fields: Fundamental limits and coding-theoretic interpretations," *IEEE Trans. on Inf. Theory*, vol. 58, no. 4, pp. 2018–2039, 2012.
- [64] T. Tomic, N. Thomos, and P. Frossard, "Distributed sensor failure detection in sensor networks," *Signal Processing*, vol. 93, no. 2, pp. 399–410, 2013.
- [65] R. Gallager, "Low-density parity-check codes," *Monograph, M.I.T. Press*, 1963.
- [66] I. Csiszár, "Linear Codes for Sources and Source Networks: Error Exponents, Universal Coding," *IEEE Trans. of Inf. Theory*, vol. 28, no. 4, pp. 585–592, July 1982.
- [67] M. Davey and D. MacKey, "Low-Density Parity Check Codes over  $GF(q)$ ," *IEEE Communication Letters*, vol. 2, no. 6, June 1998.
- [68] M. Rudelson and R. Vershynin, "Geometric Approach to Error-Correcting Codes and Reconstruction of Signals," *Int. Mathematics Research Notes*, no. 64, 2005.
- [69] Y. Lin, B. Liang, and B. Li, "Geometric Random Linear Codes in Sensor Networks," *IEEE Int. Conf. on Communications*, Aug 2008.
- [70] D. MacKay, "Good Error-Correcting Codes Based on Very Sparse Matrices," *IEEE Trans. of Inf. Theory*, vol. 45, no. 2, pp. 399–431, March 1999.
- [71] R. C. Alamino and D. Saad, "Properties of Sparse Random Matrices over Finite Fields," *Journal of statistical Mechanics*, 2009.
- [72] R. Russell and J. Christiansen, "Adaptive mesh selection strategies for solving boundary value problems," *SIAM Journal on Numerical Analysis*, vol. 15, no. 1, pp. 59–80, 1978.
- [73] D. Levin, "Mesh-independent surface interpolation," *Geometric Modeling for Scientific Visualization*, vol. 3, 2003.
- [74] A. Aldroubi and K. Gröchenig, "Nonuniform sampling and reconstruction in shift-invariant spaces," *Society for industrial and Applied Mathematics*, vol. 43, pp. 585–620, 2001.
- [75] S. Ramani, D. V. D. Ville, T. Blu, and M. Unser, "Nonideal sampling and regularization theory," *IEEE Trans. on Signal Processing*, vol. 56, pp. 1055–1070, March 2008.
- [76] L. Condat, "Reconstruction from non-uniform samples: A direct, variational approach in shift-invariant spaces," *Digital Signal Processing*, <http://dx.doi.org/10.1016/j.dsp.2013.01.015>.
- [77] H. G. Feichtinger, K. Gröchenig, and T. Strohmer, "Efficient numerical methods in non-uniform sampling theory," *Numerische Mathematik*, vol. 69, pp. 423–440, 1995.

- [78] E. Vuçini, T. Möller, and M. E. Gröller, “Efficient reconstruction from non-uniform point sets,” *The Visual Computer*, vol. 24, pp. 555–563, 7 2008.
- [79] O. Davydov and L. Schumaker, “Interpolation and scattered data fitting on manifolds using projected Powell Sabin splines,” *IMA Journal of Numerical Analysis*, Jan 2007. [Online]. Available: <http://imajna.oxfordjournals.org/cgi/content/abstract/drm033v1>
- [80] S. Lee, G. Wolberg, and S. Shin, “Scattered data interpolation with multilevel B-splines,” *IEEE Tran. on Visualization and Computer Graphics*, vol. 3, no. 3, pp. 228–244, 1997.
- [81] I. Sloan and R. Womersley, “Constructive polynomial approximation on the sphere,” *Journal of Approximation Theory*, vol. 103, no. 1, pp. 91–118, 2000.
- [82] D. Levin, “The approximation power of moving least-squares,” *Mathematics of Computation*, vol. 67, no. 224, pp. 1517–1532, 1998.
- [83] L. Machado, F. Leite, and K. Hüper, “Generalized least squares problems on riemannian manifolds,” *ROBOMAT07*, 2007.
- [84] D. Lazzaro and L. Montefusco, “Radial basis functions for the multivariate interpolation of large scattered data sets,” *Journal of Computational and Applied Mathematics*, vol. 140, no. 1-2, pp. 521–536, 2002.
- [85] B. Baxter, “The interpolation theory of radial basis functions,” *PhD Thesis Dissertion, Cambridge University*, Dec 1992.
- [86] C. Loader, *Local Regression and Likelihood (Statistics and Computing)*. Springer, 1999.
- [87] H. Zhao, S. Osher, B. Merriman, and M. Kang, “Implicit, nonparametric shape reconstruction from unorganized points using a variational level set method,” *Computer Vision and Image Understanding*, vol. 80, no. 3, pp. 295–314, Jan 2000.
- [88] M. Bertalmio, F. Memoli, L. Cheng, G. Sapiro, and S. Osher, “Variational problems and partial differential equations on implicit surfaces,” *Geometric Level Set Methods in Imaging, Vision and Graphics*, pp. 381–397, 2003.
- [89] G. Peyré, “Manifold models for signals and images,” *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 249–260, 2009.
- [90] P. J. Bickel and B. Li, “Local polynomial regression on unknown manifolds,” *IMS Lecture Notes - Monograph Series; Complex Datasets and Inverse Problems: Tomography, Networks and Beyond*, vol. 54, pp. 177–186, 2007.
- [91] M.-Y. Cheng and H.-T. Wu, “Local linear regression of Manifolds and its geometric interpretation,” *arXiv:1201.0327*, 2012.
- [92] F. Stainke, M. Hein, and B. Schölkopf, “Nonparametric Regression between general Riemannian Manifolds,” *SIAM Journ. of Imaging Sciences*, vol. 3, no. 3, pp. 527–563, 2010.

- 
- [93] J.-M. Loubes and B. Pelletier, "A kernel-based classifier on a riemannian manifold," *Statistics and Decisions*, vol. 26, pp. 35–51, 2008.
- [94] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," *Proc. of the Int. Conf. on Inf. Proc. in Sensor Networks (IPSN)*, pp. 1–10, 2004.
- [95] J. Chen, C. Richard, P. Honeine, and J. C. M. Bermudez, "Non-negative distributed regression for data inference in wireless sensor networks," *Proc. of Asilomar Conf. on Sign., Sys. and Comp.*, pp. 451–455, 2010.
- [96] D. Varagnolo, G. Pillonetto, and L. Schenato, "Distributed function and time delay estimation using nonparametric techniques," *IEEE Conf. on Dec. and Control*, 2009.
- [97] P. Honeine, M. Essoloh, C. Richard, and H. Snoussi, "Distributed regression in sensor networks with a reduced-order kernel model," *Proc. of IEEE "GLOBECOM"*, 2008.
- [98] L. Györfi and M. Wegkamp, "Quantization for nonparametric regression," *IEEE Trans. on Inf. Th.*, vol. 54, no. 2, pp. 867–874, 2008.
- [99] A. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis*. Clarendon Press - Oxford Science Publication, 1997.
- [100] V. N. Vapnik, "The nature of the statistical learning theory," *New York, Springer-Verlag*, 1995.
- [101] B. Schölkopf and A. Smola, "Learning from kernels," *M.I.T. Press*, 2002.
- [102] M. Cheraghchi, A. Hormati, A. Karbasi, and M. Vetterli, "Group testing with probabilistic tests: Theory, design and application," *IEEE Trans. of Inf. Theory*, vol. 57, no. 10, pp. 7057–7067, Oct. 2011.
- [103] C. Fragouli and E. Soljanin, "Decentralized network coding," *Proc. of IEEE Information Theory Workshop*, 2004.
- [104] C. Fragouli and A. Markopoulou, "A network coding approach to network monitoring," *Proc. of Allerton Conf. on Comm., Control and Computing*, 2005.
- [105] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Tran. on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.
- [106] J. H. van Lint and R. M. Wilson, *A course in combinatorics*. Cambridge University Press, 2001.
- [107] H. B. Chen and F. K. Hwang, "Exploring the missing link among d-separable, d-separable and d-disjunct matrices," *Discrete Applied Mathematics*, vol. 155, no. 15, pp. 662–664, 2007.
- [108] C. Cooper, "On the distribution of rank of a random matrix over finite field," *Proc. of Int. Conf. on Rand. Struc. and Algor.*, pp. 197–212, 2000.

- [109] J. Hammersley and P. Clifford, "Markov field on finite graphs and lattices," 1971.
- [110] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistics Society, Series B*, no. 36, pp. 192–236, 1974.
- [111] S. Inusah and T. Kozubowski, "A discrete analogue of the laplace distribution," *Journal of statistical planning and inference*, no. 136, pp. 1090–1102, 2004.
- [112] A. Wang and A. Chandraksan, "Energy-efficient DSPs for wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 4, pp. 68–78, 2002.
- [113] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. on Signal Proc.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.
- [114] S. Martinez, "Distributed interpolation schemes for field estimation by mobile sensor networks," *IEEE Tran. on Control Syst. Techn.*, vol. 18, no. 2, pp. 491–500, 2010.
- [115] D. Zhou and B. Schölkopf, "Regularization on discrete spaces," *Lecture notes in Computer Science*, vol. 3663, p. 361, 2005.
- [116] O. L. A. Elmoataz and S. Boughleux, "Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing," *IEEE Tran. on Image Processing*, vol. 17, no. 7, pp. 1047–1060, 2008.
- [117] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "Signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *accepted for publication in IEEE Signal Processing Magazine*, May 2013.
- [118] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis, Elsevier*, vol. 30, pp. 129–150, 2011.
- [119] D. T. Guarrera, N. G. Johnson, and H. F. Wolfe, "The Taylor expansion of a Riemannian Metric," *Preprint*, 2002.
- [120] H. Takeda, S. Farsiu, and P. Milanfar, "Robust kernel regression for restoration and reconstruction of images from sparse noisy data," *Proc. of the Int. Conf. on Image Processing*, 2006.
- [121] F. Cucker and D. X. Zhou, *Learning Theory: An approximation Theory Viewpoint*. Cambridge University Press, 2007.
- [122] R. Duda, P. Hart, and D. Stork, *Pattern classification*. A Willey-Interscience Publication, 2001.
- [123] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it?-A new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.
- [124] M. Hamermesh, *Group Theory and its Application to Physical Problems*. New York: Dover, 1962.

- 
- [125] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *Annals of Mathematical Statistics*, vol. 4, no. 23, pp. 493–507, 1952.
- [126] J. Hromkovic, R. Klasing, B. Monien, and R. Peine, *Dissemination Of Information In Interconnection Networks: Broadcasting and Gossiping*. Kluwer Academic Publishers, 1996.
- [127] H. S. Wilf, "Generatingfunctionology," *Academic Press, London*, 1994.
- [128] D. Watts and H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.
- [129] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Phys. Review E*, vol. 64, no. 2, 2001.





## Tamara Tošić

Adress: Av. de Tivoli 32  
1007 Lausanne  
Switzerland

Birth place: Subotica, Serbia  
Birth date: 12.08.1981.  
Marital status: Single  
Citizenship: Serbian

Tel: +41 (0) 21 693 27 08  
Email: tamtos@gmail.com  
Web: <http://lts4.epfl.ch/ttosic>

## Research Interests

---

Distributed algorithms, Sensor Network Signal Processing, Network Coding, Detection, Inverse problems, Nonlinear processing, Distributed Source Coding, Statistical Analysis, Nonuniform Sampling

## Education

---

- May 2008 – PhD Thesis  
*Distributed detection, reconstruction and interpolation for sensor network data*  
in collaboration with Prof. Pascal Frossard,  
Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland  
<http://www.epfl.ch>
- Oct. 2000 – Jun 2006 Bachelor (GPA 9.57 out of 10) and Master degree (GPA 9.14 out of 10),  
Electrical Engineering at Faculty of Technical Sciences (FTN),  
University of Novi Sad, Novi Sad, Serbia  
<http://www.ftn.uns.ac.rs/english>

## Professional Experience

---

- May 2008 – June 2013 Research Assistant at EPFL (Lausanne, Switzerland)  
Development of the distributed and efficient signal processing techniques for large scale sensor networks.
- Feb. 2007 – March 2008 Research Assistant at IDIAP Research Institute (Martigny, Switzerland)  
Development of the algorithm for detection of out-of-vocabulary words for speech recognition systems.
- June 2006 – Oct. 2006 Internship project at EPFL, (Lausanne, Switzerland)  
*Weighted FST-based reconstruction method for 3D objects*  
Development of the 3D object reconstruction algorithm based on the object geometry.
- April 2006 – May 2006 Internship in the Institute of Oncology (Sremska Kamenica, Serbia)  
Implementation and evaluation of Image Processing techniques for real data gathered in the Center for imaging diagnostics.
- Aug 1999 – Sep. 1999 Research project in Petnica Science Center (Valjevo, Serbia)  
*CCD as a Contactless Temperature Sensor*: theoretical and experimental temperature sensor analysis.

## Teaching Experience

---

Spring 2009	Teaching Assistant for the undergraduate course <i>Digital Signal Processing</i> given by Prof. Pascal Frossard
Spring 2010	Supervision of the Master Thesis <i>Quantization for sparse signal reconstruction from sensor network observations</i> , Yu-Chen Chan, EPFL
Spring 2011	Supervision of the Master Thesis <i>Sensor Network Inference from partial and correlated observations</i> , Contijoch Culler Xavier, UPC
Spring 2011	Supervision of the Semester Project <i>Compressed sensing on correlated signals using thresholding</i> , Alhussein Fawzi, EPFL

## Language Skills

---

Serbian (native), English (fluent), French (basic), Italian (basic)

## Technical skills

---

Programming	Matlab, Python, Pascal, C, C++
OS	Linux, Mac

## Other interests

---

Astronomy, roller-skating, jogging, literature

## Publications

---

### Publications in Journals

- Tamara Tošić, Nikolaos Thomos and Pascal Frossard, Distributed sensor failure detection in sensor networks, *Signal Processing* 93(2013), pg. 399 – 410, Feb., 2013.
- Tamara Tošić and Pascal Frossard, Efficient Distributed Data Gathering and Reconstruction in Sensor Networks *In preparation*

### Publications in International Conferences

- Tamara Tošić and Pascal Frossard, Performance bounds for sensor network gathering by coding in finite fields, *Accepted to 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.
- Alhussein Fawzi, Tamara Tošić and Pascal Frossard, Joint signal reconstruction using thresholding, *Proc. of 37th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, March 2012.
- Tamara Tošić and Pascal Frossard, Detection in Sensor Networks, *Proc. of 37th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, March 2012.
- Tamara Tošić and Pascal Frossard, Graph-based Regularization for Spherical Signal Interpolation, *Proc. of 35th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, Texas, March 2010.
- Tamara Tošić, Ivana Tošić and Pascal Frossard, Nonparametric Least Squares Regression for Image Reconstruction on the Sphere, *Proc. of 27th Picture Coding Symposium (PCS)*, Chicago, Illinois, May 2009.
- Tamara Tošić, Mathew Magimai.-Doss and Hynek Hermansky, Using Comparison of Parallel Phoneme Probability streams for OOV Word Detection, *Proc. of 16th European Signal Processing Conference (EUSIPCO)*, Lausanne, Switzerland, August 2008.